Protein structure prediction using bidirectional neural networks

Chen, Jin Miao

2007

Chen, J. M. (2007). Protein structure prediction using bidirectional neural networks.
Doctoral thesis, Nanyang Technological University, Singapore.

<https://hdl.handle.net/10356/35725>

<https://doi.org/10.32657/10356/35725>

6530242

Protein Structure Prediction Using Bidirectional Neural Networks

Chen Jinmiao

School of Computer Engineering

A thesis submitted to the Nanyang Technological University
in fulfillment of the requirement for the degree of
Doctor of Philosophy

2007

Abstract

This thesis is focused on protein secondary structure (PSS) prediction which is one of the most important problems in bioinformatics. Most of existing prediction methods either use a sliding fixed-sized window centered on the residue of interest or train on single positions of the sequence to classify mutually independent secondary structures. They are unable to take into account long-range interactions between amino acids and strong correlations between secondary structure (SS) elements. In this thesis, we propose and develop three novel prediction models, namely bidirectional long short-term memory (BLSTM), bidirectional segmented-memory recurrent neural network (BSMRNN) and cascaded bidirectional recurrent neural network (Cascaded BRNN), which are all variants of bidirectional recurrent neural network (BRNN). Our models make use of the whole protein sequence instead of local segments, and consider long-range interactions and SS-SS correlations for secondary structure prediction. For the purpose of comparison, we also implement Pollastri's BRNN that is the first recurrent neural network (RNN) architecture reported for PSS prediction, and the original BRNN that was proposed for speech recognition. All of these five prediction models are evaluated on the PSIPRED set. The experimental results show that our cascaded BRNN and BSMRNN are the best two amongst the five prediction models. Our cascaded BRNN achieves the highest accuracies and BSMRNN achieves the second highest accuracies. Their prediction performance is comparable to that of the best existing methods for PSS prediction.

Firstly, we conduct a thorough analysis of long-range interactions in proteins. Our study reveals that as an amino acid sequence is folded, long-range interactions commonly occur between amino acids far apart along the protein chain. Long-range interactions play an important role in protein folding and have a non-negligible effect on the formation of secondary structure. Furthermore, we provide statistical evidence of the negative influence of long-range interactions on PSS prediction accuracy. The long-range interactions establish relationships that are called long-term dependencies in linguistics. Hence, the predictor's capability of capturing long-term dependencies has a great impact on PSS prediction performance.

Our first two models, BLSTM and BSMRNN take into account long-range interactions for PSS prediction. Our first model BLSTM uses the state-of-the-art RNN architecture, i.e. LSTM RNN to tackle the problem of long-term dependencies. The bidirectional LSTM

achieves an overall three state accuracy Q3 of 71.53% and a segment overlap (SOV) value of 59.6577 on the PSIPRED set. Our second model bidirectional segmented-memory recurrent neural network further improves the predictor's capability of capturing long-term dependencies. Based on the observation of human memorization, we heuristically propose a novel RNN architecture called segmented-memory recurrent neural network (SMRNN) for capturing long-term dependencies. We examine the performance of SMRNN on the information latching problem and the "2-sequence problem". Our experimental results indicate that SMRNN outperforms conventional RNNs on long-term dependency problems. We also theoretically analyze how the segmented-memory of SMRNN helps in learning long-term dependencies, and study the impact of the segment length. Subsequently, we develop a bidirectional segmented-memory recurrent neural network (BSMRNN) for PSS prediction. The BSMRNN uses a forward SMRNN and a backward SMRNN to capture upstream and downstream context respectively. Then a feed-forward network is used to combine upstream context, the residue itself and downstream context to decide the secondary structure type of the residue. We test the BSMRNN predictor on the PSIPRED set and obtain a Q3 accuracy of 74.10% and an SOV value of 63.4083. The BSMRNN improves prediction performance over Pollastri's BRNN and the original BRNN.

Our third model cascaded BRNN incorporates the correlations between secondary structures for prediction. We carry out a comprehensive study of the SS-SS correlations on a large data set of non-homologous proteins. Our study suggests that secondary structure elements are strongly and non-locally correlated along the secondary structure sequence. To exploit such correlations, we propose cascaded BRNN, which consists of one BRNN on top of another BRNN. The lower BRNN predicts the secondary structures based on amino acid sequences. The higher BRNN refines the raw predictions from the lower BRNN. The two BRNNs in cascade are trained jointly. The cascaded BRNN achieves a Q3 of 74.38% and an SOV of 66.0455. It improves the SOV by 4% over the original BRNN. This substantial improvement in the SOV proves that the cascaded BRNN successfully incorporates the correlations between secondary structure patterns in predicting protein secondary structure.

Acknowledgements

First of all, I would like to express my deepest respect and gratitude to my supervisor, **Associate Professor Narendra S. Chaudhari**, who encouraged me to start this research and gave me constructive advice and patient guidance.

I would also like to extend my thanks to the laboratory technicians in the Intelligent Systems Lab for their support in equipments and tools for the project requirements.

Last but not least, I must thank my parents and all my friends for their love, encouragement and support.

Contents

Abstract	ii
Acknowledgements	iii
List of Figures	x
List of Tables	xiv
Glossary	xv
1 Introduction	1
1.1 Background	1
1.2 Problem Statement	7
1.3 Motivation and Objective	9
1.4 Thesis Structure	12
2 Literature Review: Protein Structure Determination	13
2.1 Experimental Approach	13
2.2 Protein Structure Prediction	14
2.2.1 Comparative Approaches	16
2.2.2 Ab Initio Methods	20
2.3 Molecular Dynamics Simulation	21
2.4 Protein Secondary Structure Prediction	22
2.4.1 History	23
2.4.2 Statistical Methods	25
2.4.3 Nearest Neighbor Methods	27
2.4.4 Bayesian Methods	30
2.4.5 Hidden Markov Model Methods	32
2.4.6 Neural Network Methods	33
2.4.7 Kernel Methods	36

2.4.8	Ensemble Methods	38
3	Protein Secondary Structure Prediction System Overview	40
3.1	Protein Secondary Structure Definition	40
3.2	Data Preparation	41
3.2.1	Non-Homologous Data Set	41
3.2.2	Multiple Sequence Alignment	42
3.3	Recurrent Neural Networks and Bi-directionality	44
3.4	Comparison	46
3.4.1	The Original Bidirectional Recurrent Neural Network	46
3.4.2	Pollastri's Bidirectional Recurrent Neural Network	47
3.5	Measures of Prediction Accuracy	48
3.6	Summary of This Chapter	54
4	Statistical Analysis of Long-Range Interactions in Proteins	55
4.1	Data Acquisition	55
4.1.1	EVA set	55
4.1.2	Collection of Secondary Structure Predictions	57
4.2	Inter-Residue Contacts and Long-Range Interactions	57
4.3	Correlations and Mutual Information	58
4.4	Importance of Long-Range Interactions in Protein Folding	59
4.5	Influence of Long-Range Interactions on SS Formation	61
4.6	Negative Effect of Long-Range Interactions on Prediction Accuracy	65
4.6.1	Quantity of Long-Range Interactions	66
4.6.2	α -Helices are Better Predicted than β -Sheets	68
4.6.3	Relationship between RCO and Prediction Accuracy	69
4.6.4	Prediction Accuracy with Respect to RWCO	70
4.7	Potential Approaches to Capture Long-Range Interactions	72
4.7.1	Long-Term Dependencies	72
4.7.2	Stochastic Grammar	73
4.7.3	Recurrent Neural Networks	74
4.8	Discussion	75

5	Bidirectional Long Short-Term Memory for Protein Secondary Structure Prediction	76
5.1	Learning Long-Term Dependencies is Difficult	77
5.1.1	Vanishing Gradients and Long-Term Dependencies	77
5.1.2	Several Remedies	80
5.2	Long Short-Term Memory	82
5.2.1	Memory Cells and Gate Units	83
5.2.2	Long Short-Term Memory Learning Algorithm	86
5.3	Bidirectional Long Short-Term Memory	87
5.4	Experimental Results	89
5.4.1	Results of BLSTM	89
5.4.2	Results of PBRNN	91
5.4.3	Results of the Original BRNN	93
5.4.4	Comparison Among BLSTM, PBRNN and the Original BRNN . . .	93
5.5	Discussion	93
6	Segmented-Memory Recurrent Neural Network for Long-Term Dependencies	97
6.1	Segmented-Memory Recurrent Neural Network	98
6.1.1	Segmented Memory	98
6.1.2	Network Architecture	99
6.1.3	Illustration of Dynamics	100
6.1.4	Formulation of Dynamics	101
6.1.5	SMRNN Learning Strategy	103
6.2	An Analytical Explanation of SMRNN Behavior	106
6.3	Implementation and Evaluation	111
6.3.1	The Information Latching Problem	111
6.3.2	SMRNN vs. Elman's Network on the Information Latching Problem	114
6.3.3	SMRNN vs. NARX on the Information Latching Problem	115
6.3.4	SMRNN vs. LSTM on the "2-Sequence Problem"	116
6.3.5	Impact of the Interval Length	117
6.4	Discussion	120

7	Bidirectional Segmented-Memory Recurrent Neural Network for Protein Secondary Structure Prediction	123
7.1	Bidirectional Segmented-Memory Recurrent Neural Network	124
7.1.1	Architecture	124
7.1.2	Inference and Learning	125
7.2	Experiments and Discussion	126
8	Cascaded Bidirectional Recurrent Neural Networks for Protein Secondary Structure Prediction	131
8.1	Protein Sequence-Structure Correlations	132
8.1.1	Primary Structure is Random	132
8.1.2	Primary-Secondary Structure Correlation is Weak	134
8.1.3	Strong Correlations in Secondary Structure	134
8.2	Cascaded Bidirectional Recurrent Neural Network	136
8.2.1	Architecture and Dynamics	136
8.2.2	Inference Algorithm	139
8.2.3	Learning Algorithm	139
8.3	Experiment	142
8.3.1	Experimental Settings	142
8.3.2	Results of Cascaded BRNN	143
8.3.3	Comparison among Cascaded BRNN, PBRNN and BRNN	145
8.3.4	Comparison among BLSTM, BSMRNN, Cascaded BRNN, PBRNN and BRNN	148
8.3.5	Statistical Significance Test	152
8.4	Discussion	160
9	Summary of This Research	161
9.1	Summary	161
9.2	Achievements	164
9.3	Future Work	167
9.3.1	Protein Grammar	167
9.3.2	Integration of Protein Grammar and Connectionist Approaches	169
9.4	Publications	170

References

List of Figures

1.1	The basic components of protein structure. (Adapted from [126])	2
1.2	The α helix and β sheets of protein secondary structure. The backbone of the chain is shown in red, the C_α atoms and the $C = O$ and NH groups are shown in blue, yellow, and green respectively.	5
1.3	Protein structure prediction.	8
2.1	Computational methods of protein structure prediction.	16
2.2	Structural hierarchy of proteins.	23
3.1	A simple recurrent neural network.	45
3.2	The original bidirectional recurrent neural network. The left sub-network is forward RNN, consisting of layer I_t , F_{t-1} and F_t ; the right sub-network is backward RNN, consisting of layer I_t , B_{t+1} and B_t ; the top sub-network is feed-forward network, consisting of layer I_t , F_t , B_t and O_t	47
3.3	Pollastri's bidirectional recurrent neural network. The left sub-network is forward RNN, consisting of layer I_t , F_{t-1} , FH_t and F_t ; the right sub-network is backward RNN, consisting of layer I_t , B_{t+1} , BH_t and B_t ; the top sub-network is feed-forward network, consisting of layer I_t , H_t , F_t , B_t and O_t	49
3.4	Illustration of a SOV_E calculation.	53
4.1	The correlation between the Q3 prediction accuracy of phd and RWCO.	71
4.2	The correlation between the Q3 prediction accuracy of phdpsi and RWCO.	71
4.3	The correlation between the Q3 prediction accuracy of psipred and RWCO.	71
4.4	The correlation between the Q3 prediction accuracy of apssp and RWCO.	71
4.5	The correlation between the Q3 accuracy of jpred prediction and RWCO.	72
5.1	Information latching and vanishing gradients. [13]	79

5.2	Architecture of memory cell c_j^v (the box) and its gate units in_j, out_j	83
5.3	Long short-term memory. A LSTM network with 2 input units, 2 output units, 1 memory cell block of size 2 and 1 memory cell block of size 1. in_1 marks the input gate of block 1, out_1 marks the output gate of block 1, and cell 1/block 1 marks the first memory cell of block 1. The hidden layer is fully connected. Each gate unit and each memory cell are connected to all units in the hidden layer.	84
5.4	Bidirectional long short-term memory for protein secondary structure prediction. The current input vector $I(t)$ encodes the amino acid at position t . The feed-forward sub-network combines the forward LSTM, the backward LSTM and the current input $I(t)$ to produce an output for position t	88
6.1	Illustration of segmented memory.	98
6.2	Segmented-memory recurrent neural network.	99
6.3	Dynamics of segmented-memory recurrent neural network.	100
6.4	A simple Elman's network.	107
6.5	A simple SMRNN.	107
7.1	Bidirectional segmented-memory recurrent neural network.	126
8.1	AA-AA mutual information $I(R_i; R_{i+j})$ with respect to j	133
8.2	AA-SS mutual information for residues at various separations along the amino acid chain.	135
8.3	SS-SS mutual information for residues at various separations along the amino acid chain.	136
8.4	Cascaded BRNN.	137
8.5	Direct dependencies among the variables involved in a cascaded BRNN.	140

List of Tables

2.1	Steps of the protein folding pathways	22
3.1	Three-class assignment statistics of the PSIPRED set. num_ <i>t</i> = the number of residues in secondary structure class $t \in \{C, H, E\}$; num_R = the total number of residues; num_P = the number of protein chains.	42
4.1	The ratio of long-range interactions to the total number of inter-residue contacts. contact_num refers to the number of residue-contacts, and each column represents the percentage of long-range contacts defined by different threshold values.	61
4.2	Summary of mutual information in residue-residue contacts.	64
4.3	RCO of residues in three secondary structure classes. num_R = the number of residues in three different secondary structure classes.	68
4.4	RWCO of residues in three secondary structure classes. num_R = the number of residues in three different secondary structure classes.	69
4.5	Performance of several prediction methods. num_P = the number of proteins; num_R = the number of residues.	69
4.6	The prediction accuracy for residues with local interactions only and for those with not only local interactions but also long-range interactions. num_R = the number of residues with different ranges of RCO values.	70
4.7	The correlation coefficient of the residue-wise contact order and the prediction accuracy.	72
5.1	The performance of the BLSTMs on protein secondary structure prediction. In the first row, m refers to the number of memory blocks in the LSTM subnetworks, n refers to the number of memory cells in each memory block, and W refers to the total number of weights.	90

5.2	Best results of BLSTM on different testing data sets.	91
5.3	Experiment results of PBRNN on protein secondary structure prediction .	92
5.4	The best results of Pollastri's BRNN (PBRNN) on the PSIPRED data sets. The architectural size of the PBRNN is: Inum=20, Hnum=15, Onum=3, Fnum (Bnum)=10, and FHnum (BHnum)=15.	92
5.5	Experimental results of the original BRNN on protein secondary structure prediction. The original BRNN examined has 20 input units and 3 output units, that is Inum=20 and Onum=3. In this table, Fnum (Bnum) represents the number of units in layer F (B), and Fnum is equal to Bnum; Wnum is the total number of weights.	94
5.6	The best results of the original BRNN on different testing sets. The original BRNN has 20 input units and 3 output units, that is Inum=20 and Onum=3. In this table, Fnum (Bnum) represents the number of units in layer F (B), and Fnum is equal to Bnum. Wnum is the total number of weights.	95
5.7	Comparison among PBRNN, BRNN and BLSTM.	96
6.1	The performance of the Elman's network on the information latching prob- lem by using different error threshold values.	113
6.2	Information latching by SMRNN using different sizes of training data set.	113
6.3	Information latching in SMRNN with interval $d = 15$	114
6.4	Information latching in Elman's network.	115
6.5	Information latching in NARX	116
6.6	Results of LSTM on the "2-sequence problem"([62]). N is the number of information-conveying elements at sequence begin. T is minimal sequence length. The column headed by stop gives the number of sequence presenta- tions required to achieve stopping criterion. The right-most column lists the fraction of misclassified post-training sequences from a test set consisting of 2560 sequences.	117
6.7	Results of SMRNN on the "2-sequence problem". stop=the number of epoches needed to converge. HN=number of hidden nodes. TrainNum=number of training samples. fraction misclassified=the fraction of misclassified post- training sequences from a test set consisting of 2560 sequences.	118

6.8	The results of several SMRNNs with different intervals on the information latching problem.	119
6.9	Information latching in SMRNN with interval $d = 500$	119
6.10	Information latching in SMRNN with interval $d = 1$	120
7.1	Experimental results of BSMRNN on the PSIPRED data sets. The BSMRNN size is: Inum=20, Hnum=20, Onum=3, Fnum (Bnum) =10, FSnum (BSnum) =10.	128
7.2	Performance of BSMRNNs with optimal interval lengths.	129
7.3	Comparison among PBRNN, BRNN and BSMRNN.	130
8.1	Experimental results of cascaded BRNN on protein secondary structure prediction. The cascaded BRNN has 20 input units (Xnum=20) and 3 output units (Znum=3). FYnum (BYnum)=the number of units in layer FY (BY); Ynum=the number of units in layer Y; FXnum (BXnum)=the number of units in layer FX (BX); FYnum=BYnum and FXnum=BXnum.	144
8.2	The best results of cascaded BRNN on different testing sets. The cascaded BRNN has 20 input units (Xnum = 20) and 3 output units (Znum = 3). FYnum (BYnum) = number of units in layer FY (BY); Ynum = number of units in layer Y; FXnum (BXnum) = number of units in layer FX (BX); FYnum = BYnum and FXnum = BXnum.	144
8.3	Comparison among PBRNN, BRNN and Cascaded_BRNN.	146
8.4	Parameters and total weights of cascaded BRNN, BSMRNN, BLSTM, Polastri's BRNN, and the original BRNN. Wnum is the number of parameters. m is the number of memory blocks in the forward/backward LSTM and n is the number of memory cells in each memory block.	150
8.5	Comparison among several BRNN architectures on the problem of protein secondary structure prediction.	151
8.6	Q3 distribution of BRNN, PBRNN, BSMRNN and Cascaded BRNN on the combined test set.	155
9.1	Changed PDB ids of the EVA proteins.	172
9.2	Theoretical model structures in EVA Set.	173
9.3	Redundant Entries in EVA set	174

9.4 Protein secondary structure prediction servers participating. 174

Glossary

BLAST	basic local alignment search tool
BLSTM	bidirectional long short-term memory
BRNN	bidirectional recurrent neural network
BSMRNN	bidirectional segmented-memory recurrent neural network
Causal system	A dynamic system is causal if the output only depends on the current and previous input values and not on any future input values.
Combined test set	Combined test set is defined as the union of the three independent testing sets, i.e. Combined test set=Test set1 $\cup$ Test set 2 $\cup$ Test set 3.
DEFINE	DEFINE-STRUCTURE
DSSP	Definition of Secondary Structure of Proteins
LSTM	long short-term memory
MSAP	multiple sequence alignment profile
NARX	Nonlinear Auto-Regressive models with eXogenous
PSS	protein secondary structure
PSSM	position-specific score matrix
RCO	residue contact order
RNN	recurrent neural network
RWCO	residue-wise contact order
SMRNN	segmented-memory recurrent neural network
SOV	segment overlap
SS	secondary structure
STRIDE	STRuctural IDentification
interval d	Segmented-memory recurrent neural networks break the input sequence into segments, and try to memorize the sequence segment by segment. Interval d denotes the segment length. It can be fixed or variable.

Chapter 1

Introduction

1.1 Background

Proteins are polypeptide chains carrying out most of the basic functions of life at the molecular level. The linear polypeptide chains of proteins fold into complex 3D structures whose shapes are responsible for the functions of proteins. According to the conventional Linderstrom-Lang criteria [82, 83], we specify four general levels of protein structure: primary, secondary, tertiary, and quaternary structure.

Primary Structure

The primary structure of a protein is a linear chain of amino acids joined head to tail by peptide bonds (Figure 1.1). Twenty different amino acids are commonly found in proteins. The basic chemical composition common to all 20 amino acids is shown in Figure 1.1. The central carbon atom, called C_α , forms four covalent bonds, one each with NH_3^+ (amino group), COO^- (carboxyl group), H (hydrogen), and R (side chain). The first three are common to all amino acids; the side-chain R is a chemical group that differs for each of the 20 amino acids. The amino acids are linked by asymmetric peptide bonds, allowing the designation of a beginning (N-terminus) and end (C-terminus) to the chain. The peptide

CHAPTER 1. INTRODUCTION

bond forms as a result of the reaction between the carboxyl and amino groups of neighboring residues. One oxygen from the carboxyl group and two hydrogens from the amino group get separated out as a water molecule, leading to the formation of a covalent bond between the carbonyl carbon and nitrogen. As each amino acid can be denoted by a canonical letter of the alphabet, the chain of amino acids can be represented succinctly by a sequence of letters (Figure 1.1). This sequence represents a linear chain of peptide bonded amino acids; it is called as a *primary* structure of the protein. The primary structure determines all higher levels of the structure.

Figure 1.1: The basic components of protein structure. (Adapted from [126])

Secondary Structure

In chemistry, a hydrogen bond is a type of attractive intermolecular force that exists between two partial electric charges of opposite polarity. One part of the bond involves a hydrogen atom. The hydrogen must be attached to a strongly electronegative heteroatom, such as oxygen, nitrogen or fluorine, which is called the hydrogen-bond donor. This electronegative element attracts the electron cloud from around the hydrogen nucleus and, by decentralizing the cloud, leaves the atom with a positive partial charge. Because of the small size of hydrogen relative to other atoms and molecules, the resulting charge, though only partial, nevertheless represents a large charge density. A hydrogen bond results when this strong positive charge density attracts a lone pair of electrons on another heteroatom, which becomes the hydrogen-bond acceptor. In proteins, hydrogen bonds form between the backbone oxygens and amide hydrogens. The main-chain hydrogen bonding has given rise to formal definitions of secondary structure.

The secondary structure assigns segments of consecutive residues of a protein to one of several conformational classes. These classes reflect the participation of each amino acid residue in local conformations of the chain. These local conformations are generally based on patterns of hydrogen bonds between the main-chain CO and main-chain NH atoms. There are several different definitions of protein secondary structure (abbreviated as PSS). Most commonly used definition is the Definition of Secondary Structure of Proteins (DSSP) given by Kabsch and Sander [69]. They defined eight classes of secondary structures. In what follows, i and j index the amino acid along the polypeptide chain.

CHAPTER 1. INTRODUCTION

- 1) **α -helix (H)**: A helical arrangement of residues, 3.6 residues per turn, formed by hydrogen bonds between the i th main-chain CO and the $(i + 4)$ th main-chain NH atoms (the left part of Figure 1.2).
- 2) **π -helix (I)**: Similar to the α -helix, with 4.4 residues per turn and hydrogen bonds between the atoms in the i th and $(i + 5)$ th residues.
- 3) **3_{10} -helix (G)**: Similar to the α -helix, with hydrogen bonds between the atoms in the i th and $(i + 3)$ rd residues.
- 4) **Isolated β -bridge (B)**: Two hydrogen bonds characteristic of β -structure are formed between two non-overlapping stretches of three residues each, $i - 1, i, i + 1$ and $j - 1, j, j + 1$. Either a parallel or antiparallel bridge is assigned between residues i and j , depending on which of two basic patterns is matched:
 - (a) Parallel bridges have hydrogen bonds connecting the $(i - 1)$ th CO to j th NH and j th CO to $(i + 1)$ th NH, or the $(j - 1)$ th CO to i th NH and i th CO to $(j + 1)$ th NH.
 - (b) Antiparallel bridges have hydrogen bonds connecting the i th CO to j th NH and j th CO to i th NH, or $(i - 1)$ th CO to $(j + 1)$ th NH and $(j - 1)$ th CO to $(i + 1)$ th NH.
- 5) **Extended β -strand or β -sheet (E)**: A β -sheet is formed by multiple, adjacent β -strands arranged in a plane and joined side-by-side by bridges. Strands connected by parallel bridges form parallel β -sheets (the right part of Figure 1.2); strands connected

CHAPTER 1. INTRODUCTION

by antiparallel bridges form antiparallel β -sheets (the right part of Figure 1.2).

- 6) **n -Turn (T):** An n -turn is assigned at residue i if there is a hydrogen bond from $\text{CO}(i)$ to $\text{NH}(i+n)$ where $n \in \{3, 4, 5\}$. Repeating n -turns are 3_{10} , α , and π -helices.
- 7) **Bend (S):** A bend occurs at residue i when the angle between atoms C_{i-2}^α , C_i^α , C_{i+2}^α exceeds 70° .
- 8) **Coil (C):** A region of secondary structure that does not fall into the above seven categories.

Figure 1.2: The α helix and β sheets of protein secondary structure. The backbone of the chain is shown in red, the C_α atoms and the $C=O$ and NH groups are shown in blue, yellow, and green respectively.

CHAPTER 1. INTRODUCTION

Tertiary Structure

The tertiary structure of a protein is the three-dimensional arrangement of the atoms within a single polypeptide chain. For a protein composed of a single polypeptide molecule, tertiary structure is the highest level of structure that is attained. The tertiary structure of a protein determines its biological functions.

The gross tertiary structure of a protein is captured by the specification of the structure of its backbone—the chain of NH, C_α , and CO atoms stretching from the N-terminus to the C-terminus. The structure of the backbone may be specified by the $n(\phi, \psi)$ angle pairs (ϕ denotes the NH- C_α bond angle and ψ denotes the C_α -CO bond angle), or alternatively by three-dimensional spatial coordinates of each atom in the main-chain. The full specification of the tertiary structure of a protein requires the details of side-chain conformations, which can be specified either by bond angles or by atomic coordinates.

Quaternary Structure

Many proteins are actually assemblies of more than one polypeptide chains, which in the context of the larger assemblage are known as protein subunits. In addition to the tertiary structure of the subunits, multiple-subunit proteins possess a quaternary structure, which is the arrangement into which the subunits assemble.

Quaternary structure results from the interactions between independent polypeptide chains. Factors influencing quaternary structure include hydrophobic/hydrophilic interactions, hydrogen bonding and the shape and charge distribution on associating polypeptides. Hydrophobic interaction is the main stabilizing force for protein subunits in quaternary

structure. When a single protein subunit folds into a three-dimensional shape to expose its polar side chains to an aqueous environment and to shield its nonpolar side chains, there are still some hydrophobic sections on the exposed surface. Two or more protein subunits will assemble so that their exposed hydrophobic sections are in contact.

1.2 Problem Statement

Predicting the three-dimensional (3D) structure of a protein based on the linear sequence of amino acids is a fundamental open problem in computational biology. Although the prediction of 3D protein structures is the ultimate goal, the structures still cannot be accurately predicted directly from the sequences. An intermediate but useful step is to predict the protein secondary structure (PSS), which is the linear representation of the 3D structure (Figure 1.3).

Protein secondary structure prediction is to classify a pattern of residues in amino acid sequences as protein secondary structure elements: H, I, G, B, E, T, S and A. The most commonly occurring structures are α -helices, β -strands and coils. Hence, the prediction methods normally classify only 3 standard secondary structure types associated with helices (H), β -strands (E), and coils (C). In order to understand the problem of protein secondary structure prediction, we give some definitions as below:

Definition 1 An amino acid sequence is an n -tuple $\mathbf{r} = (r_1, r_2, \dots, r_n)$ where $r_i \in \Sigma_R$ denotes the i th element in the amino acid sequence and Σ_R denotes the set of 20 amino acids, i.e. $\Sigma_R = \{A, R, N, D, C, Q, E, G, H, I, L, K, M, F, P, S, T, W, Y, V\}$; n is the length of the sequence.

CHAPTER 1. INTRODUCTION

Figure 1.3: Protein structure prediction.

Definition 2 A secondary structure sequence is an n -tuple $\mathbf{t} = (t_1, t_2, \dots, t_n)$ where $t_i \in \Sigma_T$ denotes the i th element in the secondary structure sequence and $\Sigma_T = \{H, E, C\}$; n is the length of the sequence; H , E and C denote the secondary structure elements α -helix, β -strand, and coil respectively.

Definition 3 Protein secondary structure prediction problem is to predict the secondary structure sequence $\mathbf{t}$, from the given amino acid sequence $\mathbf{r}$; that is to find the mapping F such that

$$F : \Sigma_R^n \rightarrow \Sigma_T^n \quad (1.1)$$

where n is the length of the sequence.

One approach to the PSS prediction problem, *protein structure homology modeling*, extrapolates the structure of a new sequence (target) from the known structures of related

family members (templates). It can give excellent results based on the assumption that proteins sharing high sequence similarity should have similar protein fold. However, it does not work for remote homologues ($< 25\%$ pairwise identity). Moreover, this assumption is not always true because protein can be designed with a few mutations/insertions/deletions to completely unfold, misfold and even fold to another secondary structure. In this research, we focus on predicting secondary structures for protein sequences that share less than 25 percent sequence similarity with sequences of known structure.

1.3 Motivation and Objective

During the last few decades, substantial efforts have been made for solving the problem of protein secondary structure prediction, with various approaches including Garnier, Osguthorpe, and Robson (GOR) techniques [52], Profile network from HeiDelberg (PHD) [113], nearest-neighbor methods [148] and support vector machines (SVMs) [63]. Most existing prediction methods use a sliding fixed-width window centered on the residue of interest to predict its secondary structure type. Fixed-width small window has the advantage of architectural simplicity and it allows controlling parameter over-fitting. However, they cannot exploit relevant information contained in distant portions of the sequence. Permitting certain far away inputs to exert an effect on the current prediction is paid in terms of parametric complexity. Hence, substantially increasing the size of local input window does not improve the prediction performance due to the problem of over-fitting. This weakness of local window approaches is often cited as the current limitation to accurate secondary structure prediction [29, 35, 71, 102]. There is an emerging consensus of an accuracy up-

CHAPTER 1. INTRODUCTION

per bound, slightly above 70%, to any prediction method based on *local* information only. By incorporating evolutionary information in the form of multiple sequence alignments, accuracy seems to top at the 73-74% level in spite of several attempts with sophisticated architectures. There is still a margin of 10%-15% left for further improvement to reach the upper limit of the prediction accuracy of 90% [71]. It is suggested that in order to improve prediction performance, one must make use of the whole protein sequence instead of small portions of it, and exploit distant information, in sequences and alignments, which is not contained in local input windows [102].

As a protein is folded, amino acids far apart in the sequence may be spatially close and have significant interactions. The formation of protein secondary structure especially the regions of β -sheets involves extensive interactions between residues that are not adjacent in the primary sequence. This generates complex, varying long-term dependencies in the sequences. In order to take into account long-range interactions as well as local interactions, the PSS predictors need to capture arbitrary ranges of dependencies, especially long-term dependencies. This is particularly clear in the prediction of β -sheets where hydrogen bonds can be formed between distant amino acids. To the best of our knowledge, very few successful algorithms have been reported to capture the long-range interactions for protein secondary structure prediction. Therefore, in our present work, we first investigate the effect of long-range interactions on protein secondary structure prediction, and then try to improve the prediction performance by capturing long ranges of dependencies between amino acids in the sequences.

A promising alternative to the fixed-size window approaches is recurrent neural network

CHAPTER 1. INTRODUCTION

(RNN), which is a powerful connectionist model for learning in sequential domains. Unlike feedforward neural networks, RNNs accept sequential data as inputs and employ state dynamics to store contextual information, thus they can adapt to variable width temporal dependencies. In this research, we use bidirectional recurrent neural networks (BRNNs) for PSS prediction to mitigate the limitations of the fixed-size window approaches. Unfortunately, conventional RNNs have been plagued by a major practical problem: the gradient of the total output error with respect to previous inputs quickly vanishes as the time lags between relevant inputs and errors increase [13]. Hence conventional RNNs fail to learn in the presence of time lags exceeding as few as 5-10 discrete time steps between relevant input events and target signals. In order to deal with the long-term dependencies, we propose and develop novel variants of BRNN architecture for PSS prediction.

Our preliminary study indicates that strong long-range correlations are present in the secondary structure (SS) sequences. The PHD method uses a structure-to-structure network (second level) to correlate predictions on adjacent amino acids from the sequence-to-structure network (first level). The second level of the structure-to-structure network proves to be successful in learning the local correlation among consecutive secondary structure patterns. Compared with other neural network-based predictors, the PHD method produces more similar length distribution of predicted segments and observed segments. However, by using an input window of size 13 for the first network and an input window of size 17 for the second network, the PHD method cannot capture the long-range interactions between amino acids or the long-range correlations between secondary structure elements. Except for the PHD method, most other existing prediction models are trained on single

positions of the sequence to classify mutually independent secondary structures. In this context, a further objective of our research is to take into account both long-range inter-residue interactions and SS-SS correlations when predicting protein secondary structures.

1.4 Thesis Structure

The chapters of this thesis are organized as follows. Chapter 2 gives a detailed review of the related work. Chapter 3 gives details of the PSS prediction data set used by Jones (called as PSIPRED set in this thesis), especially emphasizing the measures used (in chapters 5, 7, and 8) for prediction accuracy. Chapter 4 presents a statistical analysis of the long-range interactions between amino acids in a protein chain, and of their effect on protein secondary structure prediction. Chapter 5 introduces a novel prediction model based on a bidirectional long short-term memory algorithm. In chapter 6, we propose a novel architecture, called segmented-memory recurrent neural network, which is more capable of learning long-term dependencies. Chapter 7 describes a bidirectional segmented-memory recurrent neural network for protein secondary structure prediction. Chapter 8 shows that a cascaded bidirectional recurrent neural network achieves further improvement in prediction accuracy. Chapter 9 summarizes the work that has been done and identifies the potential directions for further studies and investigation in the future.

Chapter 2

Literature Review: Protein Structure Determination

2.1 Experimental Approach

The most direct method of determining the tertiary structure of a protein molecule is to observe it experimentally. Two experimental technologies are often used: X-ray diffraction and nuclear magnetic resonance (NMR).

Traditionally, X-ray crystallography has been used for solving the structure of proteins. However, it is useful only for the proteins that can be coaxed into a crystalline state. The method consists of placing a protein crystal in an x-ray beam and measuring the intensity of the diffracted rays. The main challenge is in creating a crystal of the target protein, a time consuming process which, at the moment, is impossible to automate and serialize. Crystallization is not always possible, especially for proteins whose native environment is not an aqueous solution e.g., transmembrane proteins. Even if the number of proteins whose primary sequences are known counts in the number of millions, only few thousands of them have been crystallized and their 3D structures are stored in the Protein Data Bank.

The alternative approaches based on Nuclear magnetic resonance (NMR) spectroscopy

CHAPTER 2. LITERATURE REVIEW: PROTEIN STRUCTURE DETERMINATION

use high magnetic fields and radio-frequency pulses to manipulate the spin states of nuclei - including ^1H , ^{13}C , and ^{15}N - that have nonzero spin angular momentum. For a molecule containing such nuclei, the result is an NMR spectrum with peaks whose positions and intensities reflect the chemical environment and nucleic positions within the molecule. As applied to protein structure analysis, NMR spectroscopy achieves accuracy comparable to that obtained with X-ray crystallography. In several respects, NMR spectroscopy offers a technique complementary to X-ray crystallography and neutron diffraction. An important consideration is that NMR structures are typically obtained from proteins in solution, with no requirement that the protein be crystallisable. Not only does this lead to a protein structure representation unconstrained by any crystal lattice, but it also allows structures to be determined for proteins that cannot be crystallized. The latter point is especially significant because a substantial fraction of all proteins are thought to contain long, disordered regions (> 40 residues) that may prevent crystallization. On the other hand, NMR historically has been limited in two important ways. First, protein structure determination by NMR methods has been limited to relatively small proteins, that is, those smaller than about 40 kD. Second, data collection for a single protein structure typically has required weeks, as compared with the minutes or hours needed for X-ray crystallography. Due to these two limitations, nuclear magnetic resonance cannot be applied at the genomic scale.

2.2 Protein Structure Prediction

Protein structure prediction aims to solve the problem of deriving the protein 3D structure from its 1D sequence using theoretical approaches, as opposed to experimental methods.

CHAPTER 2. LITERATURE REVIEW: PROTEIN STRUCTURE DETERMINATION

Experimental approaches such as X-ray crystallographic or Nuclear Magnetic Resonance (NMR) are not favorable because in addition to posing restrictions, they are also expensive, laborious and time consuming, and each has unique drawbacks. Therefore, it is becoming increasingly important to predict protein's tertiary structure from its amino acid sequence, using insights obtained from already known structures.

Protein structure prediction methods can be divided into three categories depending on the sequence similarity between the target (the protein whose structure is going to be predicted) and the template (the protein whose structure is used to model the target). They are homology modelling, fold recognition and ab initio prediction. In homology modelling, one or several template proteins of high sequence similarities with the target protein are identified. The target and the templates are then aligned using only the sequence information. The target structure is predicted using the structures of the templates from the aligned regions, along with the models for loops and unaligned regions. If no unambiguous templates can be found, fold recognition is attempted. Typically, the sequence-structure alignment (threading) is performed between the target and the template using both the sequence and structure information to identify the fold which the target is most likely to adopt. If no templates can be identified with confidence, ab initio methods are used to predict the target structure without explicitly using any templates. Homology modelling and fold recognition are generally called by a joint name of comparative approach. The ab initio methods are further classified into atomic simulations and secondary structure prediction. Figure 2.1 depicts the four principal techniques that can be employed to predict the protein structures.

Figure 2.1: Computational methods of protein structure prediction.

2.2.1 Comparative Approaches

The comparative approaches predict protein structure by referring to sequences of known structure. They can construct all-atom 3D models for those proteins that are related to at least one known protein structure. Even though comparative approaches are applicable only to the members of structurally characterized protein families, it is the most appropriate modeling method for protein structure prediction. The reason is that it results in the most accurate, detailed, and explicit models of protein structure. Earlier studies indicate that currently up to 40% of the known protein sequences have at least one segment related to one or more known structures [119]. Thus, the number of sequences that can be modeled relatively accurately by comparative modeling is already an order of magnitude larger than the number of experimentally determined protein structures [119]. Among the aforementioned techniques, comparative approaches are the most productive as they can increase the number of known structures by a factor of 10 [120].

CHAPTER 2. LITERATURE REVIEW: PROTEIN STRUCTURE DETERMINATION

Homology Modelling

At present, the prediction of an unknown protein 3D structure by homology-based comparative modelling is by far the most reliable technique. Homology modelling methods use experimentally determined protein structures (templates) to predict the conformation of another protein with a similar amino acid sequence (target). In homology modelling, there is usually strong sequence homology between the target and the templates. A protein sequence with at least 40% identity to a protein sequence of known structure can be automatically modelled, with an accuracy approaching that of a low-resolution X-ray structure or a medium-resolution nuclear magnetic resonance structure. Homology modelling is possible because a small change in the sequence usually results in a small change in the three-dimensional structure.

All homology modelling methods share three main stages. The first step is to align the target sequence with all the related proteins whose three-dimensional structures are known. In the second step, the alignment and the template structures are used to build a model for the target sequence. The parts of the target that lie in the conserved regions of the alignment (typically they are regular secondary structures like α helices and β strands) are modelled from the aligned template structures, and the non-conserved regions (usually loops) can be modelled using the loop library or ab initio prediction. By assembling the structural information at homologous regions from various proteins and optimizing it under spatial and conventional energy restraints, the full 3D structure of the target sequence can be predicted. The MODELLER [86] tool for example, predicts the three-dimensional structure by satisfying the spatial restraints, whereas the SWISS-MODEL [58] server

CHAPTER 2. LITERATURE REVIEW: PROTEIN STRUCTURE DETERMINATION

adopts the energy optimization method. In the third step, model evaluation is carried out to analyze the correctness of a homology model. The criteria for analysis of correctness can include:

- main chain conformations in acceptable regions of the Ramachandran map.
- planar peptide bonds.
- side chain conformations that correspond to those in the rotamer library.
- hydrogen-bonding of polar atoms if they are buried.
- proper environments for hydrophobic and hydrophilic residues.
- no bad atom-atom contacts.
- no holes inside the structure.

If necessary, the alignment and model building are repeated until a satisfactory model is obtained.

Fold Recognition/Threading

Homology modelling methods are successful only when a template protein structure can be found with a very high degree of sequence similarity to the target protein. If the sequence homology is less than 25%, homology modelling will fail. In the absence of a suitable homologue of known 3D structure, fold recognition methods provide another option for constructing useful tertiary structural models for a given sequence [21, 68, 28]. It has long been recognized that proteins often adopt similar folds despite no significant sequence

CHAPTER 2. LITERATURE REVIEW: PROTEIN STRUCTURE DETERMINATION

similarity and that nature is apparently restricted to a limited number of protein folds. Methods of protein fold recognition attempt to detect the structural similarity of proteins without significant sequence identity.

One way to detect structural similarity is to identify the remote homologs for the target protein via sequence comparison. Remote homologs are those proteins that share significant structure similarities but only very low sequence similarities. Advances have been made from the pairwise to multiple sequence comparison, from sequence-to-sequence, sequence-to-profile to profile-to-profile comparison. Pairwise search algorithm, such as dynamic programming [94, 136], FASTA [99, 100] and BLAST [2] do not work well when the sequence similarity is low. The ability to detect remote homologs has been greatly increased by using sequence profiles compiled from multiple sequence alignments of the target and its homologs. PSI-BLAST [3], today's most popular sequence search engine, searches the database for the most similar sequence to the target and generates a position-specific scoring matrix (PSSM) from their multiple sequence alignment. This sequence profile is then used to iteratively search for more remote homologs. SAM-T98 [70], a sequence-profile-based hidden Markov model approach also shows the remarkable ability of remote homolog detection.

Another way to detect structural similarity is to take full advantage of known protein structures. In effect, they turn the protein folding problem on its head: rather than predicting how a sequence will fold, they predict how well a fold will fit a sequence. The sequence-to-structure threading assesses the compatibility of a sequence with each known structure by a pairwise score function or single-body structural profile. An efficient thread-

ing tool such as PROSPECT [147] is preferred over homology modelling based tools when the target sequence has less than 25% pairwise sequence identity with any protein with known structure.

2.2.2 Ab Initio Methods

Despite their success in increasing the number of known structures, both comparative approaches are impractical when there are no homologues of known structures. Finding the homology between dissimilar sequences as in threading is also very difficult because it is NP-complete [77]. For these reasons, ab initio prediction of structure, using only the information of the amino acid sequences, is considered.

Given a protein's amino acid sequence, ab initio methods attempt to predict the structure of its native state without reference to any template structure. A brute force solution might be sampling the entire conformational space using molecular dynamic simulations. It is impractical since the conformational space of any protein having a reasonable size is astronomically large. The Levinthal's paradox [78] described the NP-complete problem of protein folding: for a 100-residue protein with three possible conformations per residue, it would take about 10^{27} years to fully search through all of its 3^{100} distinct conformations; yet, naturally-occurring proteins reliably fold to their native state on a time scale of seconds to minutes. Thus the central problem is to find an efficient method to sample the conformational space.

As a result of the complexity of molecular dynamic simulation, several ways are attempted to predict the structure using hierarchical methods. For example, Baldi and

Pollastri's protein-structure-prediction strategy [10, 11, 12, 103] decomposes the overall problem into three steps: (1) prediction of 1D structural features including secondary structure, relative solvent accessibility, coordination or contact number, the coupling of amino acids and strands within β -sheets from primary sequence; (2) prediction of 2D distance and/or contact maps from primary sequence and 1D structural features; and (3) prediction of 3D coordinates of all residues and atoms in the structure from 2D maps.

The two major categories of Ab Initio methods for protein structure prediction, i.e. molecular dynamics simulation and protein secondary structure prediction will be discussed in the next two sections.

2.3 Molecular Dynamics Simulation

In response to the Levithal's paradox, there are numerous attempts to simulate protein folds at atomistic level with simplified approaches. These approaches include molecular mechanics [85], Monte Carlo simulations [20] that search the energy space for a global minimum, and exhaustive/semi-exhaustive lattice based studies [135] which explore all or large amounts of conformational space using an initial approximation of the protein fold. Genetic algorithms [39] and deterministic global optimization schemes [73] are also used to minimize the protein potential energy function. These techniques rely on the thermodynamic hypothesis of protein folding [4] which suggests that the native structure of a protein sequence corresponds to its global minimum free energy state, and accordingly search for the global minimum. However, it was found that the conformation taken by in vivo proteins may not necessarily be the global minimum, but the one that is most

accessible [22]. Furthermore, the current best techniques are not able to predict proteins which are more than 150 residues in length [19].

2.4 Protein Secondary Structure Prediction

Protein folding is directed through pathways rather than random conformational searches (Table 2.1), so that protein sequences are able to fold quickly into unique native states in seconds as described by the Levinthal’s paradox. The folds in the pathways follow the structural hierarchy illustrated in Figure 2.2. Hence, an alternative approach to predict the 3D structure would be to initially reduce the complexity by decomposing it to a one-dimensional problem in the hierarchy, and to extract the crucial features from the amino acid sequences. As such, the secondary structure of proteins, which succeeds the amino acid sequence in the hierarchy, is predicted first as a stepping stone towards the three-dimensional structure.

Table 2.1: Steps of the protein folding pathways

Step 1:	Formation of local secondary structure
	<ul style="list-style-type: none">• Development of alpha-helices and beta-sheets which are connected through loops.• Duration: 5ms.
Step 2:	Hydrophobic collapse
	<ul style="list-style-type: none">• Much of the secondary structure has been formed.• Transition from secondary to tertiary structure.• Collapsed state known as a molten globule.
Step 3:	Formation of tertiary structure
	<ul style="list-style-type: none">• Packing of alpha-helices and beta-sheets• Duration: 5-1000 ms
Step 4:	Rigidification of protein structure
	<ul style="list-style-type: none">• Hydrogen bonding• Expulsion of water from protein’s interior.• Duration: 1-100 s.

CHAPTER 2. LITERATURE REVIEW: PROTEIN STRUCTURE DETERMINATION

Figure 2.2: Structural hierarchy of proteins.

2.4.1 History

First Generation: Single Residue Statistics

Shortly after Pauling et al. [98, 97] suggested an explanation for the formation of certain local conformational patterns like α -helices and β -strands, Szent-Gyrgyi and Cohen [139] attempted to correlate proline with the formation of α -helices. This idea was expanded by correlating the content of all amino acids with that of all structural elements [16, 17]. The field of protein secondary structure prediction had been opened. The main feature of the first generation methods was the use of single residue statistics for prediction. They assigned scores to individual amino acids by frequencies of occurrence in each structural state, and combined them in ways corresponding to conditional independence models [31, 33, 32, 49]. Preferences of particular amino acids for particular secondary structure states

CHAPTER 2. LITERATURE REVIEW: PROTEIN STRUCTURE DETERMINATION

were extracted from the given small databases. The low number of known structures results in biases toward certain amino acids for certain states. Another major drawback of the early methods is their assumption that each amino acid influences the secondary structure independently.

Second Generation: Segment Statistics

The growth of experimental information about protein structure enabled the second generation to parameterize the information contained in consecutive segments of residues. Typically, a window around a particular residue of interest is used to take into account local interactions between amino acids. The second generation of methods achieved improvements in accuracy by considering correlations among positions within the window explicitly (e.g. GOR [48, 112]), or using semi- and non-parametric statistical models implicitly (such as neural networks [108] and nearest-neighbor classifiers [148, 150]).

Third Generation: Using evolutionary information

The third generation methods improved prediction accuracy by the inclusion of evolutionary information via multiple alignments of homologous sequences [43, 117, 115, 122]. Several researchers achieved further improvement in the prediction accuracy by using local pairwise alignment of the query sequence with each related sequence rather than utilization of a multiple alignment [47, 67, 123].

Recent Developments

Interestingly, most recent improvements in accuracy have come from methods which are capable of considering nonlocal interactions in the sequence which occur outside a fixed

length window [34, 46, 102, 127].

In the last two decades, many computational techniques have been proposed in the literature to solve the protein secondary structure prediction problem. They can be approximately grouped into five categories: (1) statistical methods, (2) neural network methods, (3) nearest neighbor methods, (4) hidden markov model methods and (5) kernel methods.

2.4.2 Statistical Methods

Most approaches in the early era of protein secondary structure prediction were based on residue statistics. From the limited protein databases in 1960-70s, evidence was obtained for the preference of each amino acid for a particular secondary structure element. For example, the Chou-Fasman algorithm used a table of conformational propensities [31, 33, 32]. For each amino acid, Chou-Fasman algorithm's table of conformational propensities gives a value describing the given amino acid's propensity to be found in an α -helix, β -sheet, β -turns, or coil. The propensity scores were calculated by measuring the frequencies with which each amino acid appears for a given structure element of experimentally derived protein structures in a 29 protein data base. The frequencies were normalized by the prevalence of the amino acid in the data base.

The Garnier, Osguthorpe, and Robson (GOR) methods improve on the Chou-Fasman algorithm, and use information theory and Bayesian statistics for predicting the secondary structure of proteins. The GOR methods made use of an information function defined as:

$$I(S_j; R_j) = \log\left[\frac{P(S_j|R_j)}{P(S_j)}\right] \quad (2.1)$$

where S_j is one of the three secondary structural conformations (H, E, C) at position j , R_j

CHAPTER 2. LITERATURE REVIEW: PROTEIN STRUCTURE DETERMINATION

is one of the 20 amino acid residues at position j , $P(S_j|R_j)$ is the conditional probability for observing a secondary structure S_j when an amino acid R_j is present, $P(S_j)$ is the prior probability of observing S_j . GOR methods maximize the information difference, which is the difference between the likelihood of a conformational state given the residue and the likelihood of all other conformational states [48].

$$I(\Delta S_j; R_j) = I(S_j; R_j) - I(\bar{S}_j, R_j) \quad (2.2)$$

where $\bar{S}_j$ stands for the conformations other than S_j . Calculated for the three conformations, the highest value of Equation 2.2 for one of the conformations S_j will be the predicted conformation. GOR methods use a sliding window of adjacent residues to estimate the likelihood of a particular residue being one of the three types of secondary structures. Hence Equation 2.2 is extended to a local sequence along the polypeptide chain of 17 consecutive residues:

$$I(\Delta S_j; R_{j-8}, \dots, R_{j+8}) = I(S_j; R_{j-8}, \dots, R_{j+8}) - I(\bar{S}_j; R_{j-8}, \dots, R_{j+8}) \quad (2.3)$$

Several GOR techniques have evolved in the literature. They are referred to as GOR I [49], GOR II [50], GOR III [52], GOR IV [48] and GOR V [133]. Different GOR techniques use different approximations for the computation of information difference $I(\Delta S_j; R_{j-8}, \dots, R_{j+8})$. In GOR I and GOR II, the 17 positions in the window were treated as being independent, and so single-position information could be summed over the 17-residue window. Hence, they use the following approximation:

$$I(\Delta S_j; R_{j-8}, \dots, R_{j+8}) \approx \sum_{m=-8}^{m=8} I(\Delta S_j; R_{j+m}) \quad (2.4)$$

CHAPTER 2. LITERATURE REVIEW: PROTEIN STRUCTURE DETERMINATION

The next level of approximation was introduced by GOR III. In GOR III, this approach was refined by including pair frequencies derived from 16 pairs between each non-central and the central residue in the 17-residue window:

$$I(\Delta S_j; R_{j-8}, \dots, R_{j+8}) \approx I(\Delta S_j; R_j) + \left(\sum_{\substack{m \neq 0 \\ -8 \leq m \leq 8}} I(\Delta S_j; R_{j+m} | R_j) \right) \quad (2.5)$$

GOR IV and V use even more sophisticated approximations. GOR IV combines pair-wise information over all possible paired positions in a window (there are $17 \times 16/2$ possibilities), GOR V combines information theory, Bayesian statistics and evolutionary information from multiple sequence alignments.

The GOR methods use statistically sound theoretical principles and invoke no ad-hoc rules or artificial variables. Hence, they are of the most elegant methods with a high accuracy. However, a major problem of GOR methods is the limited size of the neighborhood used, which may not capture the intrinsic relationship of the input amino acid sequence and the output secondary structure sequence.

2.4.3 Nearest Neighbor Methods

Nearest neighbor methods predict the secondary structure state of the central residue of a test segment from the query protein, based on the similarity of the query sequence segment to homologous segments from proteins with known structure, even when the overall query protein sequence differs substantially from the reference proteins. A large collection of n -residue segments (exemplars) was derived from a database of known structure by sliding a window along the reference proteins. For each n -residue window over the query protein, k nearest-neighbor exemplars were identified using a certain similarity criterion. The

CHAPTER 2. LITERATURE REVIEW: PROTEIN STRUCTURE DETERMINATION

distribution of secondary structure states of the central residue in the k nearest-neighbor exemplars was then used to derive the propensities of the central residue in the query segment for the three states.

A key element in any nearest-neighbor based algorithm is the choice of scoring scheme for evaluation of segment similarity. Yi and Lander [148] have developed a hybrid scoring system that combined a sequence similarity matrix with the local structural environment scoring scheme described by Bowie et al. [21]. The Nearest Neighbor Secondary Structure Prediction (NNSSP) method by Salamov and Solovyev further improved Yi and Lander's scoring system by taking into consideration N and C-terminal position of α -helices and β -strands and also β -turns as distinctive types of secondary structure [122]. Another two important parameters are the number of nearest neighbors k and the window size n . The nearest-neighbor system operated best with $n=19$, and k between 50 and 200 [148]. Yi and Lander [148] have used a neural network to combine the results of six different nearest-neighbor predictors that employed various combinations of two different scoring systems and three different window sizes. Yi and Lander's nearest-neighbor system achieved an overall three-state accuracy of 68%, as assessed through jackknife testing [148]. The NNSSP method reached a prediction accuracy of 72.2% over the RS126 database, using multiple sequence alignments instead of single sequences [122].

The nearest-neighbor approaches to protein secondary structure prediction are simple, flexible and straightforwardly interpretable. The nearest-neighbor rule captures the biologically appealing idea of using structural homology between protein segments to aid in secondary structure prediction. Ultimately, the key to a dramatic improvement in predic-

CHAPTER 2. LITERATURE REVIEW: PROTEIN STRUCTURE DETERMINATION

tion performance is incorporating global, long-range information into the prediction scheme. Enlarging the window size is one attempt in this direction. Unfortunately, nearest-neighbor methods suffer from the same defect as other window-based approaches, and exhibit a decline in prediction accuracy with increasing window size.

Frishman and Argos' PREDATOR prediction combines the standard nearest-neighbor approach with additional propensity values that reflect non-local hydrogen-bonding interactions [46, 47]. The principal step in PREDATOR involves generating seven secondary structural propensities based on both local and non-local effects. These seven propensities rely on different concepts (hydrogen bonded pairing, sequence fragment similarity, and knowledge-based statistics) and complement each other with appropriate weighting. Three propensities are based on long-range interactions involving potential hydrogen bonding residues in antiparallel and parallel β -strands as well as α -helices. Three further propensities for helix, strand and coil were determined by a k-nearest neighbor approach with $k=25$ and 13-residue windows. A last propensity concerning β -turns was obtained by summing single-residue propensities in classic β -turn positions 1-4 using a four-residue window [64]. In order to combine the seven propensity values and predict the secondary structure at each residue, PREDATOR employs six if-then rules and makes the decisions based on seven thresholds - one for each propensity value. PREDATOR has 68% accuracy when using a single protein sequence as input, and 75% accuracy when using homologous aligned sequence input. However, PREDATOR is a fairly complex algorithm, and its results depend on the choice of the seven propensity value thresholds for which there are no clear methods guaranteeing optimality.

2.4.4 Bayesian Methods

Schmidler et al. [127] developed a probabilistic model of protein sequence/structure relationships in terms of structural segments, and formulated secondary structure prediction as a general Bayesian inference problem. In this Bayesian segmentation model, the secondary structure assignments were parameterized by the locations and types of the secondary structural segments. The goal is to infer the secondary structure assignment parameters for a new protein sequence. The authors defined a joint probability distribution for an amino acid sequence and its secondary structure assignment, then computed the conditional or posterior probability distribution over structural assignments given a new sequence via Bayesian inference. The sequence likelihood and the prior distribution of secondary structure assignments are needed to completely specify the joint distribution. An assumption was made that inter-segment residues are conditionally independent, so that the sequence likelihood can be expressed in the product of segment likelihoods. The exact form of the segment likelihood is structure-dependent, and this form for each structural type is specified by developing a probabilistic model of the given type of segment. The prior distribution was factored as a Markov process, in which each segment type depends on its nearest neighbors. After the efficient exact calculation of the posterior, prediction of secondary structure assignments was made based on maximizing this posterior distribution. Since their model is structurally similar to the class of semi-Markov source models described by Rabiner [109], they computed the maximum a posterior (MAP) value using a procedure analogous to the Viterbi algorithm for HMMs. An alternative estimator is the marginal mode predictor (Mod), which maximizes the marginal posterior distribution over segment

CHAPTER 2. LITERATURE REVIEW: PROTEIN STRUCTURE DETERMINATION

types at a single position in the sequence. The most commonly reported measure of accuracy for protein secondary structure prediction is the Q_3 value, the percentage correct on a per residue basis. Thus Mod segmentation was able to provide significant improvement over MAP segmentation.

The Bayesian method considers the information on the whole sequence instead of a local neighborhood information to take into account non-local interactions between amino acids, and provides a general framework for incorporating the growing body of scientific knowledge about protein structure into the prediction process. A distinctive feature of this approach is the ability to develop explicit probabilistic models for α -helices, β -strands, and other classes of secondary structures. Such models were able to capture core aspects of protein secondary structure formation : amino acid propensities, hydrophobicity patterns, side chain interactions, and helical capping signals. However, they are only one of many conceivable forms for the segment models. No algorithmic model selection has been proposed to optimize the selection of the best models of the segment likelihoods. In addition, the exact calculation of posterior probabilities is based on the fundamental assumption of the conditional independence between inter-segment residues that is clearly violated in the case of protein sequences. There are numerous structural motifs which rely on sequentially distant segments interacting in 3D space, including β -sheets, coiled coils, disulfide bonds, and many others. Further, this method is unable to use information from multiple sequence alignments. Due to the above disadvantages, the secondary structure prediction accuracy of the Bayesian segmentation still falls short of the best contemporary discriminative methods.

Chu et al.'s graphical model with a novel parametric likelihood function extended the

CHAPTER 2. LITERATURE REVIEW: PROTEIN STRUCTURE DETERMINATION

Bayesian segmental framework to exploit the information provided by alignment profiles [34]. The information of long range interaction in β -sheets can be directly incorporated into the modelling. A Markov Chain Monte Carlo sampling scheme was adopted to perform inference in this model. The generalization performance of this graphical model was claimed to be competitive with other contemporary methods.

2.4.5 Hidden Markov Model Methods

An Hidden Markov Model (HMM) is a probabilistic finite state machine used to model stochastic sequences. An HMM is defined by a finite set of states, each of which is associated with a probability distribution. In a particular state an outcome or observation can be generated, according to the associated probability distribution. Transitions among the states are governed by a set of probabilities called transition probabilities. One can associate a probability with a sequence according to how likely it is for an HMM to generate that sequence.

In the early 1990's , hidden Markov models (HMMs) have been applied to protein secondary structure prediction but have achieved limited accuracy [5, 138]. Recently, an HMM-based model called HMMSTR [27, 26] was reported to predict protein secondary structure with an accuracy of 74.3%. The HMMSTR was built based on the I-sites library [25] of sequence-structure motifs. The I-sites library consists of an extensive set of short sequence patterns that correlate strongly with certain recurrent local structural motifs ¹

¹In an unbranched, chain-like biological molecule, such as a protein or a strand of RNA, a structural motif is a three-dimensional structural element or fold within the chain, which appears also in a variety of other molecules. In the context of proteins, the term is sometimes used interchangeably with "structural domain," although a domain need not be a motif nor, if it contains a motif, need it be made up of only one.

CHAPTER 2. LITERATURE REVIEW: PROTEIN STRUCTURE DETERMINATION

in proteins. Each I-sites motif was represented as a chain of Markov states. Each state contains information about the sequence and structure attributes of a single position in the motif. Adjacent positions were connected by transitions from one state to the next. These linear chains of states were hierarchically merged until almost all motifs were contained in a single network of states connected by probabilistic transitions, resulting in an HMMSTR. Each state of the HMMSTR produces the predicted secondary structure according to the probability distribution specific to that state. The HMMSTR extends and generalizes the I-sites library by describing the non-random transitions between different sequence-structure motifs, and by representing overlapping motifs in a condensed form. In contrast to the linear HMMs used to model individual protein families, HMMSTR has a highly branched and multi-cyclic topology and simultaneously captures the recurrent local features of protein sequences and structures that transcend protein family boundaries.

2.4.6 Neural Network Methods

Neural networks were first used to predict protein secondary structure by Qian and Sejnowski [108]. In the influential work of Qian and Sejnowski, a local window of 13 consecutive residues is used as input to a fully connected Multilayer Perceptron with a single hidden layer. The 20 distinct amino acids were represented using orthogonal encoding in which each residue is assigned a unique binary vector i.e. $(1,0,0,\dots)$ or $(0,1,0,\dots)$. The output layer consists of three sigmoidal units with orthogonal encoding of the secondary structure classes for the residue located at the center of the input window.

Beginning with the seminal work of Qian and Sejnowski, many different feedforward

CHAPTER 2. LITERATURE REVIEW: PROTEIN STRUCTURE DETERMINATION

neural network architectures have been applied to predict protein secondary structure [137, 112, 117]. A major improvement in the prediction accuracy of these methods was made by Rost and Sander [114], who proposed a prediction scheme using a 3-level artificial neural network, known as PHD network. The first two levels consist of two MLPs in cascade, and the third level is a final arithmetic average over independently trained cascaded-MLP networks. The first MLP with a 13-residue window (sequence-to-structure network) predicts the secondary structure 3-state propensities for the central residue. The second MLP (structure-to-structure network) refines the raw predictions of the first MLP. It takes as inputs the 3-state propensities obtained from the first network and processes the information using a slightly longer window of 17 residues. The purpose of the second MLP is to capture the contextual information of neighboring secondary structures in determining the secondary structure of a particular residue.

Most of the subsequent work on predicting protein secondary structure using NNs has been based on architectures with a fixed, and relatively short input window of amino acids centered at the prediction site. A fixed small window avoids over-fitting problems, but it does not permit capturing variable long-range information. The limitations of feed-forward networks associated with small fixed-length input windows can be mitigated using other connectionist models such as recurrent neural networks (RNNs). Unlike feed-forward nets, RNNs employ state dynamics to store contextual information and they can adapt to variable width temporal dependencies. Pollastri et al. introduce bidirectional recurrent neural networks which can learn to make predictions based on variable ranges of dependencies [9, 7, 8, 105, 102]. Bidirectional RNNs (BRNNs) extend recurrent neural networks, in-

CHAPTER 2. LITERATURE REVIEW: PROTEIN STRUCTURE DETERMINATION

roducing bidirectional dynamics to capture both upstream and downstream information. Pollastri et al. trained several BRNNs of different sizes and different architectural details, and combined them to form an ensemble. The prediction algorithm is complete by incorporating mixtures of estimators and the use of evolutionary information expressed in terms of multiple alignments both at the input and output levels. Their BRNN-based prediction system claimed to achieve an overall performance close to 76% correct prediction.

Although an artificial neural network (ANN) is a powerful learning tool, it suffers from a number of problems. First, an ANN has to be regarded as a black box, as it is normally difficult to understand the internal functioning of the network. Second, there is a risk of overtraining the network, which leads to proper recognition of those patterns the network has been confronted with during training, but much less successful recognition of patterns that have not been seen. The problem of overtraining occurs if training on the samples continues for too long, and also if too many units are used in the network relative to the number of samples in the training set. The overtraining phenomenon decreases the ability of an ANN to generalize and detect true positives in the test set, sacrificing sensitivity for specificity. Third, there are known concerns in getting ANNs to effectively reach the global minimum of a very complex search space, i.e. learning the optimal parameters. The PHD and BRNN architecture are quite complex and uses a number of parameters in controlling the optimization algorithm. While Rost and Sandler tried a number of permutations on the parameter values, there is no guarantee that these somewhat ad hoc values permit optimal training for any given PHD architecture of a certain window size. Rost and Sandler made an effort to minimize the effects of parameter-dependent ANN operation by use of the third

layer. Pollastri et al. mitigated this problem by using ensemble averaging from several independently trained BRNNs.

2.4.7 Kernel Methods

Support Vector Machines are learning machines that can perform binary classification (pattern recognition) and real valued function approximation (regression estimation) tasks [24]. For binary classification, Support Vector Machines non-linearly map their n -dimensional input space into a high dimensional feature space. In this high dimensional feature space, a separating hyperplane is constructed to separate the positive from the negative examples. Let $d_+(d_-)$ be the shortest distance from the separating hyperplane to the closest positive (negative) example. Define the “margin” of a separating hyperplane to be $d_+ + d_-$. The support vector algorithm looks for the separating hyperplane with largest margin.

Hua and Sun [63] introduced the first use of SVMs to predict protein secondary structures. Since SVMs can effectively discriminate between two structural classes at a time, they constructed six binary SVM classifiers for H/ $\tilde{H}$, E/ $\tilde{E}$, C/ $\tilde{C}$, H/E, E/C and C/H, and then combined them to produce a tertiary classifier for the three secondary structure states. As with the neural network approaches, the SVM approach adopts the classical local coding scheme of the protein sequences with a sliding window. The window size has an effect on the prediction performance. A too short segment will lose some important information, while a too long segment will decrease the signal-to-noise ratio. Hua and Sun’s study suggests that the optimal window size is related to the average length of the secondary structure segments. In general, longer secondary structure segments require larger window

CHAPTER 2. LITERATURE REVIEW: PROTEIN STRUCTURE DETERMINATION

sizes. Hence, Hua and Sun used different window sizes for each of the six binary classifiers. Through sevenfold cross validation on a database of 513 non-homologous protein chains with multiple sequence alignments, their SVM method achieved higher segment overlap accuracy (SOV = 76.2%) than other existing prediction methods.

Support Vector Machine (SVM) is one of the most powerful kernel based learning machines which has favorable properties such as effective avoidance of over-fitting and structural risk minimization (the principle of structural risk minimization is an attempt to minimize the risk functional with respect to both the empirical risk and Vapnik-Chervonenkis confidence) [88]. However, there are a number of problems with SVM approaches. (1) The SVMs attain higher accuracy than a neural network when both classifiers are trained on small data sets but this advantage is lost as the size of training set is increased [142]. The linear growth in the number of support vectors with the size of the training set results in high empirical error of secondary structure prediction. (2) For $n > 2$ class classification problems, there are a lot of options as to how to combine binary classifiers to make a tertiary prediction. Many researchers found that the best way of combining classifiers was the vote classifier, which is in line with the SVM literature, but there may be problem with this approach. (3) Recent efforts have been made to predict the eight structural states assigned by DSSP [105], and this may be the direction of future developments in secondary structure prediction. The problem with the extension to eight-class classification will be the increase in the number of binary SVM classifiers.

2.4.8 Ensemble Methods

For a wide range of pattern recognition problems, model combination has proved efficient to increase the generalization performance of individual model. Many sophisticated systems have been developed for protein secondary structure prediction, based on different principles and different knowledge sources (amino acid sequences, physicochemical properties, homology, etc). We can expect significant benefits from combining different secondary structure predictions.

The simplest way to combine x different architectures is to compute an arithmetic average (jury decision):

$$\langle s_i \rangle_{\text{jury}} = \frac{1}{x} \sum_{a=1}^x s_i^a, \quad (2.6)$$

where s_i^a is the value of output unit i for architecture a . This combination has been claimed to be successful for non-network methods [15] and network methods [59, 114]. For example, Rost and Sander [114] reported a 2% improvement in predictive performance by using an ensemble of neural networks. They trained 10 separate neural network systems that vary in the choice of balanced/unbalanced training data and encoding scheme. The final prediction is simply the arithmetic average of the 10 ensemble predictions. A more elaborate approach was to train a network for combining different secondary structure predictions [150].

Cuff and Barton [36] combined four secondary structure prediction methods DSC, PHD, PREDATOR, and NNSSP by using a simple majority-wins method. They calculated the standard consensus at each position by examining the prediction for each method, and taking the most popular state. If there was a tie for a particular residue, the result from

CHAPTER 2. LITERATURE REVIEW: PROTEIN STRUCTURE DETERMINATION

the PHD method was taken. Their combination improved the average three-state prediction accuracy Q_3 by 1% from 71.9% (PHD) to 72.9% on the CB396 set, and from 73.5% (PHD) to 74.8% on the RS126 set. They further tried to weigh the predictions for each method by adding constants, and calculate the consensus from the weighted predictions in the same manner as the simple majority-wins method. The resulting Q_3 accuracy was no higher than that of the non-weighted majority-wins methods. Besides the majority-wins methods, they trained an artificial neural network on the outputs from the four methods, but yielded still lower accuracy 73.2% on the RS126 set.

Usually, the user can take advantage of the combination of prediction methods only when they provide systematically homogeneous scores. Guermeur's Multivariate Linear Regression Combiner (MLRC) [56] allows the combination of protein secondary structure prediction methods that provide inhomogeneous scores or exhibit large differences in prediction accuracy. It can be used with confidence on top of any set of prediction methods and ensures that the resulting prediction will be better than those of the individual methods. However, this combiner suffers from a drawback in the context of interest: it can only take class posterior probabilities as inputs.

Guermeur et al. [57], for the first time, combined protein secondary structure prediction methods using multi-class support vector machines (M-SVMs), which overcomes the overfitting problem faced when combining prediction methods. M-SVMs gave better performance than the ensemble methods traditionally used in the field. They also outperform the decomposition approaches based on bi-class SVMs.

Chapter 3

Protein Secondary Structure Prediction System Overview

3.1 Protein Secondary Structure Definition

The assignment of protein secondary structure categories can be performed by three programs, namely program Definition of Secondary Structure of Proteins (DSSP) [69], STRuctural IDentification (STRIDE) [45] and DEFINE-STRUCTURE (DEFINE) [111]. Cuff and Barton [36] have illustrated the differences among DSSP, STRIDE and DEFINE on the RS126 set. Assignment schemes impact prediction performance to some extent [36]. In this thesis, we concentrate on the widely used DSSP assignment. The DSSP program was designed by Kabsch and Sander [69] to standardize secondary structure assignment. Given atomic coordinates in Protein Data Bank (PDB) format, the DSSP program defines secondary structure of proteins, geometrical features and solvent exposure of proteins.

The DSSP program classifies each residue into 8 secondary structure classes: H (α helix), B (isolated β -bridge), E (extended β -strand), G (3_{10} -helix), I (π -helix), T (hydrogen bonded turn), S (bend) and C (not HBEGIT or S). Unstructured or poorly resolved regions of the protein are unassigned (X). PSS prediction methods are normally trained and assessed for

CHAPTER 3. PROTEIN SECONDARY STRUCTURE PREDICTION SYSTEM OVERVIEW

only 3 standard classes associated with helices (H), beta-strands (E), and coils (C), so the 8 classes must be reduced to 3 classes. There are four main methods to perform the reduction process.

- **Method A:** $H\{H, G\}, E\{E, B\}, C\{S, T, I, C\}$
- **Method B:** $H\{H\}, E\{E\}, C\{G, S, T, B, I, C\}$
- **Method C:** $H\{H, G, I\}, E\{E, B\}, C\{S, T, C\}$
- **Method D:** $H\{H, G\}, E\{E\}, C\{S, T, B, I, C\}$

Cuff and Barton [36] carried out a detailed study on the effect of various reduction methods on PSS prediction performance. Application of alternative 8- to 3-class reductions shows variation of over 3% on three-state overall prediction accuracy. It is suggested that we should compare prediction methods by the same reduction method. In this thesis, **method A** called CASP reduction is adopted, because it is considered to be the strictest definition, which usually results in lower prediction accuracy than the other definitions [91, 95].

3.2 Data Preparation

3.2.1 Non-Homologous Data Set

Jones's PSIPRED method [67] used a testing set of 187 unique protein folds for three-way cross-validation. The testing set was divided into three sets of 62, 62 and 63 chains. Based on structural similarity criteria, three training sets were produced for the three testing subsets respectively. Each training set is comprised of proteins that are not in the testing set and show no detectable sequence similarity with the proteins that are used for testing. Each

CHAPTER 3. PROTEIN SECONDARY STRUCTURE PREDICTION SYSTEM OVERVIEW

pair of training and testing set was screened for significant structural similarity. Any protein in the training set which had a domain fold in common with any of the domains found in the testing set was removed. In the final set, no similar folds were present in both the testing and training sets. Jones's three independent training and testing set pairs are referred to as the PSIPRED set in this thesis. We use the PSIPRED set for cross-validated training and testing of the prediction methods presented here. The PSIPRED set is available at <http://bioinf.cs.ucl.ac.uk/downloads/psipred/old/data/tdbdata.tar.gz>. The distribution of the three secondary structure classes in the PSIPRED set is summarized in Table 3.1. In this table, Combined test set is defined as the union of the three independent testing sets, i.e. Combined test set=Test set1 $\cup$ Test set 2 $\cup$ Test set 3.

Table 3.1: Three-class assignment statistics of the PSIPRED set. num_ t = the number of residues in secondary structure class $t \in \{C, H, E\}$; num_R = the total number of residues; num_P = the number of protein chains.

data set	num_C (percent)	num_H (percent)	num_E (percent)	num_R	num_P
Train set 1	93233 (44.50%)	69319 (33.08%)	46977 (22.42%)	209529	1156
Test set 1	7717 (43.52%)	6095 (34.37%)	3919 (22.10%)	17731	63
Train set 2	92216 (44.13%)	72573 (34.73%)	44158 (21.13%)	208947	1096
Test set 2	6348 (43.01%)	4687 (31.76%)	3724 (25.23%)	14759	62
Train set 3	99434 (44.34%)	74193 (33.08%)	50628 (22.58%)	224255	1092
Test set 3	6276 (41.65%)	5573 (36.98%)	3220 (21.37%)	15069	62
Combined test set	20341(42.77%)	16355(34.39%)	10863(22.84%)	47559	187

3.2.2 Multiple Sequence Alignment

Prediction from a multiple alignment profile of homologous protein sequences rather than a single sequence has long been recognized as a way to improve prediction accuracy [43, 112, 114, 115, 113]. In a multiple sequence alignment, different sequences are compared to

CHAPTER 3. PROTEIN SECONDARY STRUCTURE PREDICTION SYSTEM OVERVIEW

each other to find out which parts of the sequences are alike and which parts are dissimilar. The similarity of sequences is usually related to their evolutionary dependencies, because during evolution, residues with similar physico-chemical properties are conserved if they are important to the fold or function of the protein. The sequence alignment of homologous proteins accords with their structural alignment, and aligned residues usually have similar secondary structures. There are two kinds of alignment profiles: the multiple sequence alignment profiles (MSAPs), and position-specific score matrices (PSSMs). In this thesis, we use PSSMs on the PSIPRED set.

Starting from a sequence of unknown structure, the following steps are required to obtain the multiple sequence alignment profile (MSAP): (1) a database search for homologues to the target sequence through BLAST (Basic Local Alignment Search Tool) or iterated PSI-BLAST is carried out. (2) The selected sequences and the target sequence are then aligned. (3) The multiple sequence alignment is transformed into a sequence profile that consists of the frequencies of the 20 amino acids in each column of the multiple alignment.

The prediction accuracy of methods based on multiple sequence alignments has been found to correlate with the degree of divergence present in the aligned set of sequences. Alignments which incorporate sequences with significant yet low sequence similarity to the target protein produce more accurate predictions than those which incorporate sequences that are very closely related to the target [106]. Recently, a new search tool called PSI-BLAST is published for very sensitive sequence comparison based on the new gapped version of BLAST [3]. With suitable choices of parameters and filtering of the search data banks, PSI-BLAST greatly outperforms a standard Smith-Waterman [136] search in its

ability to detect distant homologues of a query sequence. In addition to this, PSI-BLAST generates sequence profiles as part of the search process. By compiling the PSI-BLAST-generated profile using the appropriate “debug” flag in the POSIT module, Jones [67] obtained a position specific scoring matrix (PSSM). The position specific scoring matrix has $20 \times L$ elements, where L is the length of the target sequence. Each element represents the log-likelihood of that particular residue substitution at that position in the template. The elements (typically in the range $\pm$ seven) are scaled to the required 0-1 range by using the standard logistic function $\frac{1}{1+e^{-x}}$, where x is the raw profile matrix value.

3.3 Recurrent Neural Networks and Bi-directionality

From the standpoint of computational linguistics, the procedure to construct a PSS predictor can be formulated as the problem of learning a sequential translation from strings in the amino acid alphabet into strings in the secondary structure alphabet. Thus PSS prediction can be treated as a special form of grammatical inference. In the field of grammatical inference, the main connectionist representation of languages is recurrent neural network (RNN). Unfortunately, two major limitations of conventional RNN prevent it from effectively solving the problem of PSS prediction. One is RNNs’ causality, and the other is that learning long-term dependencies with gradient descent is difficult. We adopt bi-directional neural network architectures to remove the causality assumption. To tackle the problem of long-term dependencies, we use the existing long short-term memory RNN, and furthermore we propose our segmented-memory RNN.

Recurrent neural network is a network of neurons with feedback connections. Unlike

CHAPTER 3. PROTEIN SECONDARY STRUCTURE PREDICTION SYSTEM OVERVIEW

feed-forward neural networks, RNNs employ state dynamics to store contextual information and they can adapt to variable width temporal dependencies. Figure 3.1 illustrates a simple recurrent neural network in which activations are copied from hidden layer to context layer on a one-for-one basis, with fixed weight of 1.0. The input vector is propagated through a weight layer $\mathbf{V}$, combined with the previous state activation through an additional recurrent weight layer $\mathbf{U}$. Suppose that there is a sequential input to be processed, and some clock which regulates presentation of the input to the network. Processing would then consist of the following sequence of events. The context layer is initially set to small random values. At each time step t , the input layer receives the first input in the sequence. Both the input layer and context layer activate the hidden layer; and then the hidden layer feed forward to activate the output layer. The hidden layer also feeds back to activate the context layer.

Figure 3.1: A simple recurrent neural network.

Regular unidirectional RNNs are causal in the sense that they can only make use of the input information up to the current time t . However, future input information coming up later than t is usually also useful. The secondary structural conformation of a residue

in a protein sequence may strongly depend on information located both upstream and downstream. To use all available input information, it is possible to use forward RNN and backward RNN (one for each time direction) and then merge the results. The resulting architecture is called bidirectional RNN (BRNN). BRNN extends recurrent neural networks, introducing non-causal bidirectional dynamics to capture both upstream and downstream information in sequences. In this thesis, we propose and develop several variants of BRNN for protein secondary structure prediction.

3.4 Comparison

All of our three prediction models are based on recurrent neural networks. Hence, we take two previous RNN-based architectures for comparison in our experiments of predicting protein secondary structure. They are the original bidirectional recurrent neural network and Pollastri's bidirectional recurrent neural network. Our three models and these two models are assessed on the same set of data, i.e. the PSIPRED set.

3.4.1 The Original Bidirectional Recurrent Neural Network

Bidirectional recurrent neural network (BRNN) was originally proposed for speech recognition [128]. The original BRNN architecture is illustrated in Figure 3.2. Its bidirectional dynamics are given below:

$$\begin{aligned} F_t &= \phi(F_{t-1}, I_t) \\ B_t &= \beta(B_{t+1}, I_t) \\ O_t &= \eta(F_t, B_t, I_t) \end{aligned} \tag{3.1}$$

Figure 3.2: The original bidirectional recurrent neural network. The left sub-network is forward RNN, consisting of layer I_t , F_{t-1} and F_t ; the right sub-network is backward RNN, consisting of layer I_t , B_{t+1} and B_t ; the top sub-network is feed-forward network, consisting of layer I_t , F_t , B_t and O_t .

where t denotes position within a sequence; the vector I_t encodes the external input at position t ; vector F_t stores contextual information contained at the left of position t , i.e. forward (upstream) context; vector B_t stores contextual information contained at the right of position t , i.e. backward (downstream) context; O_t is the output at position t , which depends on the local input I_t , the forward (upstream) hidden context F_t , and the backward (downstream) hidden context B_t ; the functions $\phi()$, $\beta()$ and $\eta()$ are realized respectively by the three subnetworks, i.e. the forward RNN, the backward RNN and the top feed-forward network.

3.4.2 Pollastri's Bidirectional Recurrent Neural Network

Recently, Pollastri proposed a bidirectional recurrent neural network (BRNN) for PSS prediction [102]. In this thesis, Pollastri's BRNN is abbreviated as PBRNN. The PBRNN

CHAPTER 3. PROTEIN SECONDARY STRUCTURE PREDICTION SYSTEM OVERVIEW

is a noncausal dynamic system that attempts to capture both upstream and downstream contextual information (Figure 3.3). As indicated by Figure 3.3, vector F_t (B_t) denotes the states of output layer of forward (backward) RNN, vector FH_t (BH_t) denotes the state of hidden layer of forward (backward) RNN, and vector O_t (I_t) denotes the state of output (input) layer of Pollastri's BRNN. The upstream and downstream contextual information is contained in the vectors F_t and B_t that are defined by the following recurrent bidirectional equations:

$$\begin{aligned} F_t &= \phi_1(FH_t) = \phi_1(\phi_2(F_{t-1}, I_t)) = \phi(F_{t-1}, I_t) \\ B_t &= \beta_1(BH_t) = \beta_1(\beta_2(B_{t+1}, I_t)) = \beta(B_{t+1}, I_t) \end{aligned} \quad (3.2)$$

where $\phi()$ and $\beta()$ are adaptive nonlinear transition functions. They are realized by two recurrent neural networks, namely the left and right sub-networks of PBRNN. The output at residue t is a vector of posterior probabilities of secondary structure classes for sequence position t given the protein's sequence of amino acids. In this model, the output is computed as:

$$O_t = \eta_1(F_t, B_t, H_t) = \eta_1(F_t, B_t, \eta_2(I_t)) = \eta(F_t, B_t, I_t) \quad (3.3)$$

where $\eta()$ is realized by an MLP, i.e. the top sub-network of PBRNN.

3.5 Measures of Prediction Accuracy

Several different measures of prediction accuracy have been suggested in the literature. Throughout this thesis, we use the standard performance measures reviewed in [6].

Figure 3.3: Pollastri's bidirectional recurrent neural network. The left sub-network is forward RNN, consisting of layer I_t , F_{t-1} , FH_t and F_t ; the right sub-network is backward RNN, consisting of layer I_t , B_{t+1} , BH_t and B_t ; the top sub-network is feed-forward network, consisting of layer I_t , H_t , F_t , B_t and O_t .

Three-state overall per-residue accuracy

The most common measure is the overall three-state prediction percentage Q3 defined as the ratio of correctly predicted residues to the total number of residues in the database under consideration. Unless we specify otherwise, Q3 percentages are measured on a per-residue basis.

A measure of the performance on secondary structure class $t \in \{H, E, C\}$ is the percentage $Q_{.t}$ of correctly predicted residues observed in class t .

$$Q_{.t} = \frac{a_t}{b_t} \quad (3.4)$$

CHAPTER 3. PROTEIN SECONDARY STRUCTURE PREDICTION SYSTEM OVERVIEW

where a_t is the number of correctly predicted residues in structure t , and b_t is the number of residues observed in type t . These measures can be helpful in detecting over-prediction (ex. non-helix is predicted as helix) and under-prediction (ex. helix is predicted as non-helix) of one or more types of secondary structures.

Matthews' correlation coefficients

A complementary measure of prediction accuracy is obtained from Matthews' correlation coefficients [87] for each of the three secondary structure classes: C_H, C_E, C_C.

$$C_t = \frac{p_t n_t - u_t o_t}{\sqrt{(p_t + u_t)(p_t + o_t)(n_t + u_t)(n_t + o_t)}} \quad (3.5)$$

where $t \in \{H, E, C\}$, p_t is the number of positive cases that were correctly predicted, n_t is the number of negative cases that were correctly rejected, u_t is the number of under-predicted residues (misses), and o_t is the number of over-predicted cases (false positive). Let M_{tl} be the number of residues observed in structure t and predicted in structure l , where $t, l \in \{H, E, C\}$. The above values can be written as:

$$a_t = M_{tt}, n_t = \sum_{l \neq t} \sum_{k \neq t} M_{kl}, u_t = \sum_{l \neq t} M_{tl}, o_t = \sum_{l \neq t} M_{lt} \quad (3.6)$$

Matthews' correlation coefficients indicate the success of predicting residues for each type of secondary structure, which are +1 if the predictions are all correct or -1 if all the predictions are false. The advantage of Matthews' correlation coefficients is seen in case of a random or trivial prediction. For example, a trivial prediction of helices for all residues in PSIPRED Test set 1 (Table 3.1) gives Q_H=100% and Q3=34.37%, but $C_{\{H,E,C\}}=0$. Similarly, $C_{\{H,E,C\}}$ is close to 0 for random predictions. In this sense, Matthews' correlation coefficients give more reliable estimates of the prediction accuracy.

Segment Overlap

The ultimate goal of secondary structure prediction is reliable prediction of tertiary structures that are important for function. The record of protein evolution reveals that considerable variation in the position and length of secondary structure segments can be accommodated within the same 3D fold. For assessing how well the main secondary structure elements are predicted, we need a convincing measure that has to reflect the potential to predict main features of the protein's tertiary structure given the secondary structure prediction. Rost et al. [116] carried out a detailed comparison of pairs of structurally homologous proteins with divergent sequences, showing that proteins with the same 3D fold differ by 12% in secondary structure. Originally, they defined a measure of Segment Overlap (SOV) [116], attempting to evaluate the similarity of 3D structure at the secondary structure level.

We adopt the segment overlap measure redefined by Zemla et al. [149]. In what follows, the subscript 1 or 2 labels either the proteins being compared or the observed (usually subscript 1) and predicted (usually subscript 2) structure; (s_1, s_2) denotes a pair of overlapping segments, where s_1 and s_2 are two segments that have in common at least one residue position in the same secondary structure; $S(i)$ denotes the set of all the overlapping pairs of segments (s_1, s_2) in state i ,

$$S(i) = \{(s_1, s_2) : s_1 \cap s_2 \neq \emptyset, s_1 \text{ and } s_2 \text{ are both in conformational state } i\} \quad (3.7)$$

$S'(i)$ denotes the set of all segments s_1 for which there is no overlapping segment s_2 in state

CHAPTER 3. PROTEIN SECONDARY STRUCTURE PREDICTION SYSTEM OVERVIEW

i ,

$$S'(i) = \{s_1 : \forall s_2, s_1 \cap s_2 = \emptyset, s_1 \text{ and } s_2 \text{ are both in conformational state } i\} \quad (3.8)$$

For secondary structure state $i \in \{H, E, C\}$, the segment overlap measurement is defined as:

$$SOV_{.i} = 100 \times \frac{1}{N(i)} \sum_{S(i)} \left[\frac{\minov(s_1, s_2) + \delta(s_1, s_2)}{\maxov(s_1, s_2)} \times \text{len}(s_1) \right] \quad (3.9)$$

with the normalization value $N(i)$ defined as:

$$N(i) = \sum_{S(i)} \text{len}(s_1) + \sum_{S'(i)} \text{len}(s_1) \quad (3.10)$$

where $\text{len}(s_1)$ is the number of residues in segment s_1 ; $\minov(s_1, s_2)$ is the length of the actual overlap of s_1 and s_2 , i.e. the number of residues which both segments have in common; $\maxov(s_1, s_2)$ is the total extent to which either of the two segments has a residue in state i ; $\delta(s_1, s_2)$ is defined as:

$$\delta(s_1, s_2) = \min[(\maxov(s_1, s_2) - \minov(s_1, s_2)); \minov(s_1, s_2); \text{int}(\text{len}(s_1)/2); \text{int}(\text{len}(s_2)/2)] \quad (3.11)$$

The SOV measure defined above is extended to evaluate multi-state secondary structure prediction as below:

$$SOV = 100 \times \left[\frac{1}{N} \sum_{i \in \{H, E, C\}} \sum_{S(i)} \frac{\minov(s_1, s_2) + \delta(s_1, s_2)}{\maxov(s_1, s_2)} \times \text{len}(s_1) \right] \quad (3.12)$$

where the normalization value N is a sum of $N(i)$ over all three conformational states.

$$N = \sum_{i \in \{H, E, C\}} N(i) \quad (3.13)$$

we use an example given in Figure 3.4 to illustrate the calculation of $SOV(E)$. In

CHAPTER 3. PROTEIN SECONDARY STRUCTURE PREDICTION SYSTEM OVERVIEW

Figure 3.4: Illustration of a SOV_E calculation.

the observed structure, the first β -strand β_O^1 belongs to the set $S'(E)$ because it does not overlap with any β -strand of the predicted structure; the second β -strand β_O^2 produces two overlapping pairs: (β_O^2, β_P^1) and (β_O^2, β_P^2) belonging to the set $S(E)$.

$$S(E) = \{(\beta_O^2, \beta_P^1), (\beta_O^2, \beta_P^2)\}$$

$$S'(E) = \{\beta_O^1\} \quad (3.14)$$

The lengths of the four β -strands in the observed structure and predicted structure are:

$$\text{len}(\beta_O^1) = 3, \text{len}(\beta_O^2) = 6, \text{len}(\beta_P^1) = 5, \text{len}(\beta_P^2) = 2 \quad (3.15)$$

$N(E)$, $\delta(\beta_O^2, \beta_P^1)$ and $\delta(\beta_O^2, \beta_P^2)$ are calculated by:

$$N(E) = \text{len}(\beta_O^2) + \text{len}(\beta_O^1) = 6 + 3$$

$$\delta(\beta_O^2, \beta_P^1) = \min((10 - 1); 1; \text{int}(6/2); \text{int}(5/2)) = 1$$

$$\delta(\beta_O^2, \beta_P^2) = \min((6 - 2); 2; \text{int}(6/2); \text{int}(2/2)) = 1 \quad (3.16)$$

The value of SOV(E) is calculated as follows:

$$\text{SOV}_E = 100 \times \frac{1}{6 + 6 + 3} \times \left(\frac{1+1}{10} + \frac{2+1}{6} \right) \times 6 = 28.0 \quad (3.17)$$

3.6 Summary of This Chapter

This thesis reports several novel variants of bidirectional recurrent neural network. All of them are proposed for the same problem of protein secondary structure prediction. They are all evaluated on the same set of data and using the same measures of prediction accuracy. In addition, our methods are compared against other two existing methods, i.e. the original BRNN and Pollastri's BRNN. To give an overview of the methods presented in this thesis, this chapter defines protein secondary structure prediction, describes the data used in the experiments, introduces two existing methods that are taken for comparison, and summarizes the performance measures. This chapter summarizes the common aspects of all the methods which are to appear in the upcoming separate chapters.

Chapter 4

Statistical Analysis of Long-Range Interactions in Proteins

We carry out a comprehensive study of long-range interactions on a large data set of non-homologous proteins. Our study reveals that the long-range interactions between amino acids far apart are common in protein folding, and play an important role on the formation of secondary structure. Using residue-wise contact order (RWCO) to describe long-range interactions, we further evaluate the effect of long-range interactions on secondary structure prediction. We select five most popular prediction methods and collect their prediction results on data sets of non-homologous proteins. All the five prediction methods show a significant negative correlation between prediction accuracy and RWCO.

4.1 Data Acquisition

4.1.1 EVA set

EVA (EValuation of Automatic protein structure prediction) is a web-based server that provides a continuous, fully automated, and statistically significant analysis of structure prediction servers. Everyday, EVA downloads the newest protein structures from Protein Data Bank (PDB) [14]. The structures are added to the MySQL databases. The sequences

CHAPTER 4. STATISTICAL ANALYSIS OF LONG-RANGE INTERACTIONS IN PROTEINS

are extracted for every protein chain, and are sent to each prediction server by META-PredictProtein (META-PP) [71]. Then META-PP collects the results and sends them back to EVA. The central EVA site at Columbia is online at <http://cubic.bioc.columbia.edu/eva/>.

The EVA set, the largest sequence unique subset of PDB, fulfills the criterion that no pair in the subset has more than 33% identical residues over more than 100 residues aligned. All proteins in the EVA set were sequence unique. That is, at the date of the submission of the new proteins, they do NOT have *significant sequence similarity* to proteins of known structure. The term *significant sequence similarity* refers to the following operational definition: Two proteins A and B are considered to be significantly similar if we can safely predict that B has the same structure as A by simply comparing their sequences.

We extracted the EVA set from PDB based on the current EVA list of 3449 chains (http://cubic.bioc.columbia.edu/eva/res/unique_list.html). In order to extract the 3D coordinates of C_α atoms, together with the secondary structures, we ran the DSSP (Define Secondary Structure of Proteins) program [69] on all the PDB files of the EVA proteins, excluding those for which DSSP crashes due, for instance, to missing entries or format errors. In the current EVA list, there are some theoretical model structures (Table 9.2 in the Appendix) and outdated PDB IDs (each structure in the PDB is represented by a 4 character identifier) (Table 9.1 in the Appendix). After removing those redundant entries (Table 9.3 in the Appendix) and those for which DSSP could not produce an output, we obtained the final set consisting of 3374 protein chains, which forms the source for our present study in this chapter.

4.1.2 Collection of Secondary Structure Predictions

Twenty-one PSS prediction servers participate in the EVA evaluation. They are apssp, phdpsi, jpred, phd, psipred, prof.king, profsec, sspro2, jufo, phdprof, prof, prof0, prof1, prospect, pssp, sable, samt99_sec, scratch, sspro, sspro1 and apssp2 (available online at http://cubic.bioc.columbia.edu/eva/doc/explain_methods.html). For each protein in the EVA unique list, EVA evaluated these PSS prediction servers on the deposition date of that protein. The prediction results from these prediction servers are downloadable at <http://cubic.bioc.columbia.edu/eva/doc/ftp.html>. Out of these 21 servers, we selected apssp [110], phdpsi [106], jpred [37], phd [113] and psipred [67] to evaluate the effect of long-range interactions on prediction accuracies. Details for these five servers are given in Table 9.4 in the Appendix.

4.2 Inter-Residue Contacts and Long-Range Interactions

The folding of a polypeptide chain into a compact, unique 3D structure is directed and stabilized by intra-molecular interactions between the constituent amino acid residues along the chain. The residues in a protein molecule are represented by the residue centers, i.e. their C_α atoms. The complicated atomic structure of a protein can be represented as the ordered succession of residue centers (C_α atoms) described by their X, Y, Z coordinates. We used the three-dimensional C_α coordinates to calculate the residue-residue (C_α - C_α) distance matrices:

$$r_{i,j} = \text{sqrt}[(X_{C_\alpha}^i - X_{C_\alpha}^j)^2 + (Y_{C_\alpha}^i - Y_{C_\alpha}^j)^2 + (Z_{C_\alpha}^i - Z_{C_\alpha}^j)^2] \quad (4.1)$$

CHAPTER 4. STATISTICAL ANALYSIS OF LONG-RANGE INTERACTIONS IN PROTEINS

Two residues are considered to be in contact if the C_α atoms from each residue locate closer than a threshold value. Different threshold values, for example, 4.5Å, 6Å, 8Å, 10Å and 12Å have been used by different researchers to define inter-residue contact. The selection of the threshold distance does not significantly affect the correlations described in this work.

Long-range interactions are defined to be those contacts between residues that are far separated along the primary sequence. Most existing local window-based approaches do not exploit relevant information contained in residues which are more than 8 residues away from the central residue to be predicted. For example, the GOR (Garnier-Osguthorpe-Robson) techniques take into account only the local subsequence around the residue of interest by using an approximation:

$$I(S_i; R_1, R_2, \dots, R_i, \dots, R_{last}) \simeq I(S_i; R_{i-8}, \dots, R_i, \dots, R_{i+8}) \quad (4.2)$$

where R_i represents the amino acid at position i and S_i is the secondary structure type of R_i . They neglected the interactions between residues separated by more than 8 positions along the amino acid sequence. In the famous PHD network, the structure-to-structure net with an input window of size 17 does not exploit information outside the range $[t-8, t+8]$ for predicting the SS type at position t . Therefore, in our work, we define two interacting residues to be in long-range contact if they are separated by at least 9 residues along the chain.

4.3 Correlations and Mutual Information

The average strength of correlations between two discrete random variables X and Y is the mutual information, an information-theoretic measure of the knowledge that each variable

carries about the other.

$$I(X, Y) = \sum_{x,y} p(x, y) \log \frac{p(x, y)}{p(x)p(y)} \quad (4.3)$$

where $p(x, y)$ is the joint probability distribution function of X and Y , and $p(x)$ and $p(y)$ are the marginal probability distribution functions of X and Y respectively. When calculated with base 2 logarithms, mutual information is expressed in units of bits. A high mutual information value is a result of strong correlation, whereas a mutual information value of zero indicates uncorrelated variables. A basic property of the mutual information is that

$$I(X, Y) = H(X) - H(X/Y) = H(Y) - H(Y/X) \quad (4.4)$$

where $H(X) = -\sum_x p(x) \log p(x)$ is the entropy of X , $H(Y) = -\sum_y p(y) \log p(y)$ is the entropy of Y , $H(X/Y) = \sum_{x,y} p(x, y) \log \frac{p(y)}{p(x,y)}$ is the conditional entropy of X given random variable Y , and $H(Y/X) = \sum_{x,y} p(x, y) \log \frac{p(x)}{p(x,y)}$ is the conditional entropy of Y given random variable X . Mutual information cannot exceed the entropy of either variable, and reaches this upper bound when the variables are perfectly correlated, i.e. $p(x, y) = p(x) = p(y)$ and hence $H(X/Y) = 0$ and $H(Y/X) = 0$.

4.4 Importance of Long-Range Interactions in Protein Folding

Gromiha and Selvaraj [132, 54, 55, 76] defined long-range interactions to be those residue-contacts for which the sequence separation of contacting residues is larger than 4 residues. Their work revealed that long-range interactions play an active role in the folding of globular, $(\alpha/\beta)_8$ barrel and membrane proteins. For globular proteins, they studied the influence of long-range interactions in different structural classes of proteins and found that 85% of

CHAPTER 4. STATISTICAL ANALYSIS OF LONG-RANGE INTERACTIONS IN PROTEINS

residues are involved in long-range contacts. The all- α class proteins have more long-range contacts in the 4-10 range and the all- β class proteins have more long-range contacts in the 11-20 range. The range 4-10 is favored by the $\alpha + \beta$ class of proteins. In the α/β class, the α -helices and β -strands occur alternatively and some residue distances are necessary to form β -strand and barrel, which leads to having contacts in the 21-30 range. A similar trend was also observed in their study of $(\alpha/\beta)_8$ barrel proteins. In the case of membrane proteins (a membrane protein is a protein molecule that is attached to, or associated with the membrane of a cell or an organelle), the long-range interactions have an important effect on the stabilization of helix-helix interactions in transmembrane helical (TMH) proteins and on the close packing of β -strands in transmembrane strand (TMS) proteins. Specifically, the TMS proteins prefer the 11-20 range, and the higher long-range contacts are influenced up to the range 21-30.

Gromiha and Selvaraj's study put emphasis on the different influence of long-range interactions in different folding types of globular and membrane proteins. Their observation was obtained from relatively small data sets. Besides, the definitions of inter-residue interaction and long-range interaction given by them are different from the definitions given in this thesis. In our study, we used the large EVA set of 3374 non-homologous proteins. We calculated the numbers of residue-contacts and the percentage of long-range contacts in the EVA protein set. Different partitioned distances from $4.5\text{\AA}$ to $12\text{\AA}$ are used to define inter-residue contact. In our definition of long-range interactions, the sequence separation between contacting residues must be larger than 8. In this experiment, we use different threshold values of sequence separation ranging from 9 to 20. For threshold values larger

CHAPTER 4. STATISTICAL ANALYSIS OF LONG-RANGE INTERACTIONS IN PROTEINS

than 20, the current limited data set is insufficient to generate statistically significant results. The results for different combinations of parameters are recorded in Table 4.1.

Table 4.1: The ratio of long-range interactions to the total number of inter-residue contacts. contact_num refers to the number of residue-contacts, and each column represents the percentage of long-range contacts defined by different threshold values.

cutoff distance	4.5Å	6 Å	8Å	10Å	12Å
contact_num	52424	900798	2236370	5231096	9455424
> 9	74.42%	46.80%	55.66%	60.94%	68.25%
> 10	71.56%	45.56%	54.33%	59.64%	66.85%
> 11	69.17%	44.41%	53.06%	58.41%	65.60%
> 12	67.65%	43.26%	51.86%	57.27%	64.42%
> 13	65.69%	42.20%	50.75%	56.20%	63.29%
> 14	63.60%	41.18%	49.68%	55.18%	62.18%
> 15	61.52%	40.17%	48.64%	54.17%	61.10%
> 16	60.17%	39.21%	47.60%	53.16%	60.05%
> 17	58.61%	38.27%	46.60%	52.17%	58.99%
> 18	56.99%	37.33%	45.63%	51.19%	57.94%
> 19	55.49%	36.46%	44.66%	50.21%	56.89%
> 20	54.15%	35.60%	43.70%	49.22%	55.83%

As observed from Table 4.1, long-range interactions are important in protein folding. Approximately one half of inter-residue contacts are long-ranged. The fraction of long-range interactions is smallest when 6Å is selected as the threshold value for defining residue contact. Therefore, in the rest of this chapter, we fixed the threshold value for residue contact to be 6Å.

4.5 Influence of Long-Range Interactions on SS Formation

There are several concrete examples of secondary structures whose formation is influenced by long-range interactions between amino acids [65, 66, 89, 92, 96, 151]. For example,

CHAPTER 4. STATISTICAL ANALYSIS OF LONG-RANGE INTERACTIONS IN PROTEINS

small fragments of the same sequence are found in different secondary structures [65, 66, 96, 151]. An 11-amino-acid sequence designed by Minor and Kim [89] folds as an α -helix when in one position but as a β -sheet when in another position of the primary sequence of the IgG-binding domain of protein GB1. Minor and Kim's experimental results demonstrated that non-local interactions can determine the secondary structure of peptide sequences of substantial length. Most recently, Crooks and Brenner [35] showed that local sequence information is insufficient to determine secondary structure, implying indirectly that nonlocal interactions are important for secondary structure formation. In particular, they found that correlations between neighboring amino acids are essentially uninformative and only one-fourth of the total information needed to determine the secondary structure is available from local inter-sequence correlations. As they estimated, the entropy density of secondary structure sequences is 0.60 bits per residue, while the local mutual information between primary and secondary structure is only 0.16 bits per residue. The scarcity of local sequence information places severe constraints on any prediction algorithm that purports to extract secondary structure information from local sequence correlations.

On the other hand, some other researchers denied the significant influence of long-range interactions on secondary structure specification [41, 96]. For example, Pan et al. [96] analyzed the relationship between the fraction of n -mers with a unique central secondary structural state and the length of n -mers in a low-homologous database. They computed the fraction of determinable n -mers as below:

$$F_{dt} = N_{dt} / (N_{dt} + N_{undt}) \quad (4.5)$$

CHAPTER 4. STATISTICAL ANALYSIS OF LONG-RANGE INTERACTIONS IN PROTEINS

where N_{dt} is the sum of appearance frequencies of determinable n -mers, and N_{undt} is the sum of appearance frequencies of undeterminable n -mers. From the selected database, they calculated the values of F_{dt} for $1 \leq n \leq 11$. It was found that the fraction of determinable n -mers increases with the increasing length of the n -mers. When $n > 11$, N_{dt} and N_{undt} were too small to supply statistically relevant information for analysis, due to the limited size of the database used. An approximation was made that the values of F_{dt} can be calculated by an exponential function. According to that exponential function, the values of F_{dt} is greater than 0.9 when $n=14$ and 17. Thus, they concluded that the secondary structural state of position i is mainly determined by a continuous subsequence fragment from $i - n$ to $i + n$ where $n = 7, 8$.

Is protein secondary structure primarily determined by local interactions between residues closely spaced along the amino acid backbone, or also strongly influenced by long-range interactions between residues far apart along the chain ? As we know, quantitative measurement of sequence-structure correlations can elucidate the relative importance of different interactions to protein structure formation and facilitate the rational design of structure prediction algorithms. To answer the above question, we examine the average strength of correlations among amino acid (AA) identity and secondary structure (SS) element, both for contacting residues closely located along the amino acid chain and for residues proximate in 3D space but distantly separated along the chain.

In Table 4.2, we use mutual information to quantify the protein AA-AA, AA-SS and SS-SS correlations in residue-residue contacts. All contacts have a C_α - C_α distance of 6 Å or less; the contacting residues are separated in the polypeptide chain by 2 or more residues; R_i is

CHAPTER 4. STATISTICAL ANALYSIS OF LONG-RANGE INTERACTIONS IN PROTEINS

Table 4.2: Summary of mutual information in residue-residue contacts.

Mutual Information	residue-residue contacts ($ i - j \geq 2$)	local contacts ($2 \leq i - j \leq 8$)	long-range interactions ($ i - j > 8$)
$MI(R_i, R_j)$	0.007289	0.006261	0.017497
$MI(R_i, S_j)$	0.024069	0.014110	0.033919
$MI(S_i, S_j)$	0.459712	0.265390	0.262924

the amino acid at position i and S_i is the corresponding secondary structure. The amount of information contained in contacting amino acids is modest: $MI(R_i, R_j) = 0.007289$. In principle, direct interactions between amino acids determine the native structure of a protein; however, these interactions do not lead to strong inter-residue AA-AA correlations. The AA-SS correlation in residue-residue contacts is also weak: $MI(R_i, S_j) = 0.024069$. Given that residue i is in contact with residue j , the amino acid type at position i does not have significant effect on the formation of secondary structure at position j . Interestingly, the secondary structural states are strongly correlated, resulting in relatively high mutual information between S_i and S_j : $MI(S_i, S_j) = 0.459712$. Not much difference is found between the values of $MI(R_i, R_j)$, $MI(R_i, S_j)$ and $MI(S_i, S_j)$ for $2 \leq |i - j| \leq 8$ and those for $|i - j| > 8$. The mutual information in long-range interactions is comparable with the mutual information in local contacts. It implies that long-range interactions play a non-negligible role on the formation of secondary structure.

4.6 Negative Effect of Long-Range Interactions on Prediction Accuracy

Most current protein secondary structure prediction methods assign a secondary structure to the center of a local segment and thus usually do not explicitly consider long-range interactions between amino acids. There have been several experimental examples where a local segment alone is not sufficient to determine its central secondary structure [54, 71, 89, 92]. Interestingly, some earlier studies suggested that the current insufficient accuracy of secondary structure prediction does not result from the neglect of long-range interactions [41, 96]. For example, Fiser et al. [41] compared the prediction accuracy for residues involved in significant long-range interactions with the prediction accuracy for the other residues having local interactions only. Based on their comparison results, they claimed that the role of long-range interactions in defining the secondary structures is overestimated. So far, there have not been many systematic studies on the effect of long-range interactions on PSS prediction accuracy. In our present study, we investigate the effect of long-range interactions on the prediction accuracy of secondary structures, and come to a conclusion that the accuracy gets worse for residues whose interacting partners are located farther apart along the amino acid sequence. Specifically, we use Residue Contact Order (RCO) and Residue-Wise Contact Order (RWCO) to describe the extent of long-range interactions and examine the relationship between the RCO/RWCO and the prediction accuracy on the EVA set.

4.6.1 Quantity of Long-Range Interactions

We numerically specify long-range interactions in proteins before we study their effect on PSS prediction accuracy.

Contact order

The contact order is the average sequence separation between contacting residues in the native state of a protein.

$$CO = \frac{1}{N} \sum_{i=1}^{L-3} \sum_{j=i+3}^L |i - j| \delta_{ij} \quad (4.6)$$

where N is the total number of contacts in the protein; L is the length of the protein; $\delta_{ij} (1 \leq i, j \leq L)$ represents the contact map of the protein, i.e. $\delta_{ij} = 1$ if residue i and j are in contact, and 0 otherwise.

Relative contact order

Relative contact order is the average sequence distance between all pairs of contacting residues normalized by the total sequence length. Relative contact order reflects the relative importance of local and non-local contacts to a protein's native structure.

$$CO_{rel} = \frac{1}{N \times L} \sum_{i=1}^{L-3} \sum_{j=i+3}^L |i - j| \delta_{ij} \quad (4.7)$$

The contact order and relative contact order were originally introduced to quantify the complexity of the native topology of proteins and to investigate the correlation between the native structure and the folding rate. As such, both the contact order and the relative contact order are per-protein quantity. In the next two subsections, we extend the definition of contact order and relative contact order to make them a per-residue quantity.

Residue contact order

Kihara [71] introduced for the first time the residue contact order (RCO) to evaluate the long-range interactions for each residue. The RCO indicates the average sequence separation between contacting residues to a residue of interest. The RCO of residue i is formulated as below:

$$RCO_i = \frac{1}{n} \sum_{j:|j-i|>2}^L |i-j|\delta_{ij} \quad (4.8)$$

where n is the number of contacts between residue i and the other residues. We do not take into account residue-residue pairs for which $|i-j|$ is smaller than 3. In this way, we exclude trivial contacts between nearest- and next-nearest residues along the sequence.

Residue-wise contact order

Residue-wise contact order was originally introduced by Kinjo and Nishikawa [72] as a new kind of one-dimensional protein structure to represent the extent of long-range contacts. The residue-wise contact order (RWCO) of the i -th residue of a protein is defined by

$$RWCO_i = \frac{1}{L} \sum_{j:|j-i|>2}^L |i-j|C_{i,j} \quad (4.9)$$

where L is the length of the amino acid sequence of the protein, and $C_{i,j}$ defines the contact between residues i and j with a sigmoid function:

$$C_{i,j} = \frac{1}{1 + \exp(r_{i,j} - d_c)} \quad (4.10)$$

where $r_{i,j}$ is the distance between the C_α atoms of the i -th and j -th residues; d_c is the cut-off distance for residue-contact definition; residue i and j are considered to be in contact

if $r_{i,j} < d_c$. The RWCO is a generalization of the residue contact order, and is also a per-residue quantity.

Kihara [71] has examined the effect of proteins' global features on the accuracy of protein secondary structure prediction. He found that the prediction accuracy is not affected by the length, the contact order, and the relative contact order of proteins. In what follows, we will investigate the influence of long-range interactions on the prediction accuracy by using the RCO and RWCO as measures of long-range interactions.

4.6.2 α -Helices are Better Predicted than β -Sheets

Using the EVA set, we calculated the RCO and RWCO values for residues in different classes of secondary structures. When calculating RCO, δ_{ij} is equal to 1 if residue i and j are in contact, and 0 otherwise; a cut-off distance value of $6\text{\AA}$ is selected for defining inter-residue contacts. When calculating RWCO, d_c is fixed to be $6\text{\AA}$. As indicated by the results in Table 4.4 and Table 4.3, the RCO and RWCO values of residues in class E are larger than those of residues in class H. Our observation accords with the experimental fact that the configurations of α -helices are continuous, whereas β -sheet structures typically range over several discontinuous sections in an amino acid sequence. Hence, the formation of β -strands is more dominated by long-range interactions than that of α -helices.

Table 4.3: RCO of residues in three secondary structure classes. num_R = the number of residues in three different secondary structure classes.

class	num_R	max_RCO	min_RCO	average_RCO
H	144182	838.5	0	11.6377
E	85319	595	0	41.9357
C	164782	1001	0	26.1306

CHAPTER 4. STATISTICAL ANALYSIS OF LONG-RANGE INTERACTIONS IN PROTEINS

Table 4.4: RWCO of residues in three secondary structure classes. num_R = the number of residues in three different secondary structure classes.

class	num_R	max_RWCO	min_RWCO	average_RWCO
H	144182	5.38045	0.00104984	0.280893
E	85319	5.5523	0.000118382	0.718816
C	164782	6.12381	0	0.360048

Existing approaches can predict α -helices with accuracies between 70% and 80%. However, the problem of predicting β -sheet regions has not been treated at a comparable level. The overall prediction accuracies of apssp, phd, phdpsi, jpred, psipred and sspro2 on the EVA proteins are summarized in Table 4.5. The EVA set most probably includes sequences that were used to train these programs, which would inflate the accuracy beyond that shown in the original publications. As shown in table 4.5, α -helices are better predicted than β -strands by each of the five prediction methods evaluated here.

Table 4.5: Performance of several prediction methods. num_P = the number of proteins; num_R = the number of residues.

method	num_P	num_R	Q3	Q_C	Q_H	Q_E
apssp	1264	247762	77.05	82.70	77.58	65.38
phdpsi	1780	359152	74.11	76.97	77.18	63.17
jpred	1301	258007	74.23	84.61	71.34	59.01
phd	1883	380026	70.62	72.56	74.12	60.94
psipred	1667	337333	78.21	79.08	81.97	70.18

4.6.3 Relationship between RCO and Prediction Accuracy

Residues with RCO values larger than 8 are considered to have long-range contacts. Table 4.6 records the prediction accuracies for residues with $RCO \leq 8$ and for those with $RCO > 8$. We observe from the table that prediction accuracies drop if the residue has contacts with

CHAPTER 4. STATISTICAL ANALYSIS OF LONG-RANGE INTERACTIONS IN PROTEINS

distant residues that are not located in the local input window. The reason is that these methods use a local input window of size 17 to capture information contained in the residues that are no more than 8 residues apart from the residue at the center.

Table 4.6: The prediction accuracy for residues with local interactions only and for those with not only local interactions but also long-range interactions. num_R = the number of residues with different ranges of RCO values.

method	RCO $\leq$ 8		RCO $>$ 8	
	num_R	Q3	num_R	Q3
apssp	141951	78.92%	105811	74.54%
phdpsi	204357	76.50%	154795	70.95%
jpred	148020	76.55%	109987	71.12%
phd	214573	73.18%	165453	67.31%
psipred	190802	80.00%	146531	75.88%

4.6.4 Prediction Accuracy with Respect to RWCO

The correlations between RWCO and PSS prediction accuracies of the five methods are depicted in Figures 4.1- 4.5. When calculating RWCO, we selected $d_c = 6\text{\AA}$ as the cut-off distance for residue-contact definition. All the residues in the testing data set are sorted in increasing order of RWCO, and then split into subsets, each of which contains around 5000 residues. The average three-state accuracy Q3 on each subset is calculated and plotted. Figures 4.1- 4.5 indicate that the secondary structure prediction accuracy drops as the RWCO value becomes higher.

A correlation coefficient is a number between -1 and 1 which indicates the strength and direction of a linear relationship between two random variables. There are a number of different correlation coefficients that might be appropriate depending on the kinds of variables being studied. The best known is the Pearson’s correlation coefficient, which is

CHAPTER 4. STATISTICAL ANALYSIS OF LONG-RANGE INTERACTIONS IN PROTEINS

Figure 4.1: The correlation between the Q3 prediction accuracy of phd and RWCO.

Figure 4.2: The correlation between the Q3 prediction accuracy of phdpsi and RWCO.

Figure 4.3: The correlation between the Q3 prediction accuracy of psipred and RWCO.

Figure 4.4: The correlation between the Q3 prediction accuracy of apsssp and RWCO.

obtained by dividing the covariance of the two variables by the product of their standard deviations.

$$r = \frac{\sum_{i=1}^n (x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^n (x_i - \bar{x})^2 \sum_{i=1}^n (y_i - \bar{y})^2}} \quad (4.11)$$

We compute the Pearson's Correlation Coefficient between RWCO and prediction accuracy Q3 using the data points in Figure 4.1- 4.5. As illustrated in Table 4.7, all the prediction methods used show significant negative correlation coefficient, which verifies our observations that the prediction accuracy gets worse as RWCO increases.

CHAPTER 4. STATISTICAL ANALYSIS OF LONG-RANGE INTERACTIONS IN PROTEINS

Figure 4.5: The correlation between the Q3 accuracy of jpred prediction and RWCO.

Table 4.7: The correlation coefficient of the residue-wise contact order and the prediction accuracy.

method	correlation coefficient
apssp	-0.793196
jpred	-0.834448
phd	-0.694418
phdpsi	-0.646376
psipred	-0.563313

4.7 Potential Approaches to Capture Long-Range Interactions

4.7.1 Long-Term Dependencies

Interactions between amino acids are an important factor in protein folding. The folded protein secondary structure entails pairing between contacting amino acids, establishing relationships that are called *dependencies* in linguistics. *Long-term dependencies* in protein sequences are defined to be relationships established by long-range interactions. Our studies have suggested that the insufficient accuracy of current PSS prediction methods is commonly attributed to the neglect of long-range interactions. In order to take into account both local and long-range interactions, we need a flexible predictor that is capable

of learning short-term dependencies as well as long-term dependencies. In the field of computational linguistics, there are two typical representations of languages, namely formal grammar and recurrent neural network. They both can accept the whole protein sequences of variable length as inputs rather than local segments of fixed length.

4.7.2 Stochastic Grammar

Stochastic context-free grammars (CFGs) have been successfully used as model for RNA secondary structure prediction [121]. From a computational linguistics standpoint, the language of RNA is dominated by nested pairwise correlations that can be easily generated by the context-free grammars. As an RNA sequence folds, hydrogen bonds form between the Watson-Crik base-pairs. CFGs allow for productions of the form $S \rightarrow ASU$, $S \rightarrow USA$, $S \rightarrow GSC$ and $S \rightarrow CSG$ which describe the Watson-Crik base-pairing structure in RNA. In the case of proteins, the correlations among amino acid entities are beyond context-free. The next category of languages in the Chomsky hierarchy, the context-sensitive languages, are intractable because for them parsing is an NP-complete problem. Recently, a class of mildly context-sensitive grammars (MCSGs), which exceed CFGs but still can be parsed in polynomial time, have found applications for the description of secondary and higher structures of some biological sequences [1, 74, 141]. However, the parsing and learning algorithms are computationally inefficient.

Inferring production rules of protein grammar based on protein sequences with known structures falls into the field of grammatical inference. Grammatical inference has been extensively studied in the purely symbolic paradigm for over 30 years. Gold's and Pinker's

previous studies concluded that grammatical inference is an exceptionally complex task [53, 101]. A lot of research on natural language processing has indicated that language learning is really hard. Learning the language of proteins is even more difficult due to proteins' complicated structures and the scarcity of prior knowledge. Therefore, at this stage, predicting protein secondary structure using classical grammatical methods is very difficult.

4.7.3 Recurrent Neural Networks

The task of protein secondary structure prediction can be viewed as finding the mapping from amino acid sequences to secondary structure sequences. Recurrent neural networks (RNNs) can be trained to map input sequences to output sequences, for applications in sequence recognition/production or time-series prediction. RNNs are well suited for the above applications because they have internal states that can store and update context, i.e. information computed from the past inputs and useful to produce target outputs. RNNs do not have the parsing problem of grammatical methods, and can be efficiently trained by gradient descent algorithm. However, RNN is a causal dynamic system. A dynamic system is causal if the output only depends on the current and previous input values and not on any future input values. While the problem of PSS prediction is non-causal because the pattern of secondary structure associated with an amino acid in a sequence may strongly depend on both upstream and downstream information. A noncausal generalization of RNNs called bidirectional RNN (BRNN) is able to circumvent this limitation. Unfortunately, gradient descent learning algorithm is not sufficiently powerful to discover long-term dependencies.

BRNN is usually trained by gradient descent, hence it has difficulties in learning long-term dependencies in protein sequences. In the next three chapters, we are going to propose several variants of BRNN to deal with this problem.

4.8 Discussion

We have addressed the importance of long-range interactions on PSS formation, and the negative effect of long-range interactions on the current window-based approaches of PSS prediction. To overcome this limitation of window-based approaches, we should exploit relevant information contained in the whole protein sequence instead of fix-sized local windows. In order to make use of the whole protein sequence, and exploit both local and distant information for PSS prediction, we consider two candidate approaches, namely stochastic grammar and recurrent neural network. They have been suggested to have different strengths and weaknesses. Integration of them allows combining the advantages of both symbolic and connectionist approaches. At this stage, we focus on recurrent neural networks and develop PSS predictors based on RNNs. In the main part of this thesis, we carry out detailed investigation on this issue.

Chapter 5

Bidirectional Long Short-Term Memory for Protein Secondary Structure Prediction

Recurrent neural networks (RNNs) use recurrent connections to model the flow of time directly, in this way, they are able to incorporate context into sequence processing tasks. Standard RNNs (by which we mean RNNs containing hidden layers of recurrently connected neurons) have two major drawbacks. Firstly, since they process inputs in temporal order, their outputs tend to be dependent on previous context only. By adding a delay between the outputs and the targets, RNNs introduce future context, but do not usually make full use of backward dependencies. Secondly they are known to have difficulty in learning time dependencies more than a few time steps long. An elegant solution to the first problem is provided by bidirectional networks. For the second problem, an alternative RNN architecture and algorithm, Long Short-Term Memory, has been shown to be capable of learning long time dependencies. In this chapter, we constructed a bidirectional LSTM for the first time to predict protein secondary structure.

5.1 Learning Long-Term Dependencies is Difficult

5.1.1 Vanishing Gradients and Long-Term Dependencies

In principle, the recurrent connections permit recurrent neural networks to encode arbitrary temporal dependencies. However, the standard structural framework of recurrent neural networks does not adequately model long-term dependencies. In practice, standard RNNs perform poorly on long-term dependency problems, i.e. those problems for which the desired output depends on inputs presented at times far in the past [13]. Many researchers have reported this limitation of standard RNNs [13, 18, 60, 62, 81, 23].

The two typical learning algorithms for recurrent neural networks are *back propagation through time* (BPTT) [144] and *real time recurrent learning* (RTRL) [146]. Both BPTT and RTRL are based on computing the gradients of a cost function with respect to the weights of the network. Unfortunately, the gradient-based algorithms face an increasingly difficult problem as the duration of the dependencies to be captured increases [13]. With BPTT or RTRL, error signals flowing backwards in time tend to (1) blow up or (2) decay. Case (1) may cause oscillating weights, while in case (2) learning to store contextual information over long time intervals takes a prohibitive amount of training time or does not work at all.

Bengio et al. [13] have provided theoretical and experimental evidence showing that learning long-term dependencies with gradient descent is difficult. They argued that for many practical applications the goal of RNNs is robust information latching, i.e., the long-term storage of definite bits of information into the state variables in the presence of noise. More specifically, they proved that the necessary conditions of robust information latching

CHAPTER 5. BIDIRECTIONAL LONG SHORT-TERM MEMORY FOR PROTEIN SECONDARY STRUCTURE PREDICTION

bring a problem of vanishing gradients. In other words, when the robust latching conditions are satisfied, the vanishing gradient effect occurs. In this section, we briefly describe the theoretical analysis given by Bengio et al. [13].

Bengio et al. considered a non-autonomous discrete-time system having state transition function:

$$a_t = M(a_{t-1}) + u_t \quad (5.1)$$

where a_t and u_t are respectively the system state and the external input at time t , and M is the nonlinear map from state at time $t - 1$ to state at time t . In order to latch one bit of state information, one wants to restrict the system state a_t to a subset of its domain. To make sure that a_t remains in such a region, the system dynamics can be chosen such that this region is the basin of attraction of an attractor. In [13], Bengio used *hyperbolic attractors* to latch bits of information and gave the following definitions.

Definition 4 A set of points E is said to be invariant under a map M if $E = M(E)$.

Definition 5 A hyperbolic attractor is a set of points X invariant under the differentiable map M , such that $\forall a \in X$, all eigenvalues of $M'(a)$ are less than 1 in absolute value.

Definition 6 The basin of attraction of an attractor X is the set $\beta(X)$ of points converging to X under the map M , i.e., $\beta(X) = \{a : \forall \epsilon, \exists l, \exists x \in X \text{ s.t. } \|M^l(a) - x\| < \epsilon\}$.

Definition 7 The reduced attracting set of a hyperbolic attractor X is defined to be the set $\Gamma(X)$ of points y in the basin of attraction of X , such that $\forall l \geq 1$, all the eigenvalues of $(M^l)'(y)$ are less than 1.

Based on the above definitions, Bengio presented theoretical evidence showing that the system is robustly latched to a hyperbolic attractor X as long as a_t remains in the reduced attracting set $\Gamma(X)$ of X [13].

As illustrated in Figure 5.1, the fixed point is the hyperbolic attractor X ; the shadow region is the basin of attraction of attractor X , i.e. $\beta(X)$; the dark shadow region is

CHAPTER 5. BIDIRECTIONAL LONG SHORT-TERM MEMORY FOR PROTEIN SECONDARY STRUCTURE PREDICTION

the reduced attracting set $\Gamma(X)$; and $|M'|$ is the norm of the Jacobian (matrix of partial derivatives) of the map M . Actually, the space of $\beta(X)$ is decomposed into two regions: one where $|M'| > 1$ and the other where $|M'| < 1$. In regions where $|M'| > 1$, arbitrarily small perturbation can push the state a_t out of the basin of attraction, so it is impossible to robustly latch information (Figure 5.1(a)). In regions where $|M'| < \lambda < 1$, there is a level of perturbation (depending on λ) below which the state will remain in the reduced attracting set $\Gamma(X)$ and will gradually go closer to a certain volume around the attractor X (Figure 5.1(b)).

Figure 5.1: Information latching and vanishing gradients. [13]

Unfortunately, in the regions ($|M'| < 1$) where one can robustly latch information, one has the problem of vanishing gradients. The derivative of cost function E at time t with respect to parameter W is given by:

$$\frac{\partial E(t)}{\partial W} = \sum_{\tau \leq t} \frac{\partial E(t)}{\partial a(\tau)} \frac{\partial a(\tau)}{\partial W} = \sum_{\tau \leq t} \frac{\partial E(t)}{\partial a(t)} \frac{\partial a(t)}{\partial a(\tau)} \frac{\partial a(\tau)}{\partial W} \quad (5.2)$$

The partial derivative of $a(t)$ with respect to $a(\tau)$ ($\tau \leq t$) is simply the product of the

CHAPTER 5. BIDIRECTIONAL LONG SHORT-TERM MEMORY FOR PROTEIN SECONDARY STRUCTURE PREDICTION

map derivatives between τ and t :

$$\frac{\partial a(t)}{\partial a(\tau)} = \frac{\partial a(t)}{\partial a(t-1)} \frac{\partial a(t-1)}{\partial a(t-2)} \cdots \frac{\partial a(\tau+1)}{\partial a(\tau)} = \prod_{\tau < t' \leq t} \frac{\partial a(t')}{\partial a(t'-1)} \quad (5.3)$$

In the regions where $|M'| < 1$, the norm of each map derivative $\frac{\partial a(t')}{\partial a(t'-1)} (\tau < t' \leq t)$ is less than 1. Hence, $\frac{\partial a(t)}{\partial a(\tau)}$ converges exponentially fast to zero as $t - \tau$ increases. That is, for a term of the sum with $\tau \ll t$, we have

$$\left| \frac{\partial a(t)}{\partial a(\tau)} \right| \rightarrow 0 \quad (5.4)$$

Thus,

$$\left| \frac{\partial E(t)}{\partial a(\tau)} \frac{\partial a(\tau)}{\partial W} \right| \rightarrow 0 \quad (5.5)$$

This term tends to become very small in comparison to the terms for which τ is close to t . The gradients decay with the increase of $t - \tau$. A small change in W would have effect mostly on the near past (τ close to t). Thus the consequence of vanishing gradients is that short-term information completely dominates long-term information, making the task of capturing distant information very hard.

5.1.2 Several Remedies

Several approaches have been suggested to circumvent the problem of vanishing gradients. Some consider alternative network architectures, such as Nonlinear Auto-Regressive models with eXogenous (NARX) recurrent neural network [81, 80], Hierarchical Recurrent Neural Network [60], Long Short-Term Memory [62] and Latched Recurrent Neural Network [23]. Others try alternative optimization algorithms, such as simulated annealing algorithm [13], cellular genetic algorithm [75], expectation-maximization algorithm [84] and least-squares-based optimization in a layer-by-layer fashion [30]. In this section, we briefly

CHAPTER 5. BIDIRECTIONAL LONG SHORT-TERM MEMORY FOR PROTEIN SECONDARY STRUCTURE PREDICTION

review the alternative approaches that have been proposed to improve learning long-term dependencies.

Alternative optimization algorithms: non-gradient methods and evolutionary algorithms

Bengio et al. [13] investigated several alternative algorithms such as simulated annealing, multi-grid random search, time-weighted pseudo-newton optimization and discrete error propagation. A powerful global search method, genetic algorithm, has also been used to train recurrent neural networks [124]. Recently, Ma and Ji [84] have proposed a probabilistic model based on the expectation-maximization (EM) algorithm for training RNNs.

Alternative Network Architectures

Adding connections with time delays to the RNNs helps gradient descent algorithms in learning long-term dependencies. Two typical examples are Nonlinear Auto-Regressive models with eXogenous (NARX) recurrent neural networks [79, 81] and hierarchical recurrent neural networks [60]. The time-delayed connections in NARX provide a more direct link between the activity of a neuron and the output of the network at a later moment. Gradient flow in this architecture can be improved because embedded memories efficiently introduce “shortcuts” in the error propagation path through time. NARX can retain information for two to three times as long as conventional recurrent neural networks. Hierarchical recurrent neural networks use delays and multiple time scales to achieve a multi-resolution representation of context. The temporal dependencies are structured hierarchically, which implies that long-term dependencies are represented by variables with

CHAPTER 5. BIDIRECTIONAL LONG SHORT-TERM MEMORY FOR PROTEIN SECONDARY STRUCTURE PREDICTION

long time scales. This scheme allows context information and credit information to be respectively propagated forward and backward more easily. However, in the absence of prior knowledge concerning the problem to solve, it is difficult to choose the appropriate location and the delay associated to a time-delayed connection [18]. Boné et al. proposed a constructive approach that starts with an RNN having no time-delayed connections and progressively adds a few such connections heuristically [18].

Recently, a novel recurrent architecture called *Long Short-Term Memory* (LSTM) was designed to overcome the problem of vanishing gradient [62]. By enforcing constant error flow through internal states of special units, LSTM can learn to bridge long time lags. More recently, Šter addressed the problem with *Latched Recurrent Neural Network* (LRNN). The LRNN ignores unimportant input information by using a register of latches as the input layer of the network [23].

5.2 Long Short-Term Memory

The Long Short-Term Memory (LSTM) is a very general and promising adaptive sequence processing device with a wide field of potential applications. In LSTM, constant error flow is enforced through “constant error carousels” within special units, and multiplicative gate units learn to open and close access to the constant error flow. LSTM is not affected by the problem of vanishing gradients. They can learn to bridge minimal time lags in excess of 1000 discrete time steps without loss of short time lag capabilities. Moreover, LSTM’s learning algorithm is more efficient than previous RNN learning algorithms such as real time recurrent learning (RTRL) and back propagation through time (BPTT). LSTM has

CHAPTER 5. BIDIRECTIONAL LONG SHORT-TERM MEMORY FOR PROTEIN SECONDARY STRUCTURE PREDICTION

been shown to outperform other RNNs on many artificially generated sequential processing tasks involving long time lags [62]. This has encouraged researchers to apply LSTM to more realistic problems, such as speech recognition, grammatical inference, and time series prediction.

5.2.1 Memory Cells and Gate Units

The central feature of LSTM is the memory block, which contains one or more memory cells and a pair of adaptive, multiplicative gating units, i.e. input gate and output gate (Figure 5.2). Each memory cell has its core a recurrently self-connected linear unit called “Constant Error Carousel” (CEC). CEC is protected by the input and output gates: the input gate learns to protect the memory cells from irrelevant inputs, the output gate learns to turn off a cell block that is generating irrelevant output. If the memory block has two or more memory cells, the gating units will be shared by all of them.

Figure 5.2: Architecture of memory cell c_j^v (the box) and its gate units in_j , out_j .

Figure 5.3 gives an example of LSTM RNN architecture with one input layer, one hidden layer, and one output layer. The basic unit in the hidden layer of an LSTM RNN is the memory block that contains memory cells and corresponding gate units. The hidden layer is fully self-connected. Each memory cell and each gate unit receives incoming connections

CHAPTER 5. BIDIRECTIONAL LONG SHORT-TERM MEMORY FOR PROTEIN SECONDARY STRUCTURE PREDICTION

from all memory cells and gate units. The output layer's only incoming connections originate at the memory cells in the hidden layer. The input layer has forward connections to all units in the hidden layer.

Figure 5.3: Long short-term memory. A LSTM network with 2 input units, 2 output units, 1 memory cell block of size 2 and 1 memory cell block of size 1. in1 marks the input gate of block 1, out1 marks the output gate of block 1, and cell 1/block 1 marks the first memory cell of block 1. The hidden layer is fully connected. Each gate unit and each memory cell are connected to all units in the hidden layer.

As shown in Figure 5.2, c_j^v stands for the v -th memory cell of the j -th memory block, in_j indexes the input gates and out_j indexes the output gates of the j -th memory block. in_j 's activation at time t is denoted by $y^{in_j}(t)$, out_j 's by $y^{out_j}(t)$. We have

$$y^{in_j}(t) = f_{in_j}(net_{in_j}(t)) \quad (5.6)$$

$$y^{out_j}(t) = f_{out_j}(net_{out_j}(t)) \quad (5.7)$$

where $net_{in_j}(t)$ is the net input of the input gate in_j , and $net_{out_j}(t)$ is the net input of the output gate out_j . The net input of a unit is defined as a linear function of the individual inputs to that unit.

$$net_{in_j}(t) = \sum_u w_{in_j u} y^u(t-1) \quad (5.8)$$

$$net_{out_j}(t) = \sum_u w_{out_j u} y^u(t-1) \quad (5.9)$$

CHAPTER 5. BIDIRECTIONAL LONG SHORT-TERM MEMORY FOR PROTEIN SECONDARY STRUCTURE PREDICTION

where the summation index u may stand for input units, gate units, memory cells, or even conventional hidden units if there are any. At time t , memory cell c_j^v 's output $y^{c_j^v}(t)$ is computed as

$$y^{c_j^v}(t) = y^{out_j}(t)h(s_{c_j^v}(t)) \quad (5.10)$$

where the internal state $s_{c_j^v}(t)$ is

$$s_{c_j^v}(0) = 0, s_{c_j^v}(t) = s_{c_j^v}(t-1) + y^{in_j}(t)g(net_{c_j^v}(t)) \text{ for } t > 0 \quad (5.11)$$

where $net_{c_j^v}$ is the net input of the v -th memory cell of the j -th memory block.

$$net_{c_j^v}(t) = \sum_u w_{c_j^v u} y^u(t-1) \quad (5.12)$$

The net input and the activation of output unit k are:

$$\begin{aligned} net_k(t) &= \sum_j \sum_v^{S_j} w_{kc_j^v} y^{c_j^v}(t) \\ y^k(t) &= f_k(net_k(t)) \end{aligned} \quad (5.13)$$

where S_j is the size of the j -th memory block, i.e. the number of memory cells in the j -th memory block.

In the above equations, the squashing functions f , h and g are defined by:

$$f(x) = \frac{1}{1 + \exp(-x)} \quad (5.14)$$

$$h(x) = \frac{2}{1 + \exp(-x)} - 1 \quad (5.15)$$

$$g(x) = \frac{4}{1 + \exp(-x)} - 2 \quad (5.16)$$

CHAPTER 5. BIDIRECTIONAL LONG SHORT-TERM MEMORY FOR PROTEIN SECONDARY STRUCTURE PREDICTION

5.2.2 Long Short-Term Memory Learning Algorithm

The quadratic error between the actual output and the target output of LSTM at time t is given by:

$$E(t) = \sum_k \frac{1}{2} (t^k(t) - y^k(t))^2 \quad (5.17)$$

where k indexes the output units, $t^k(t)$ is the k -th output unit's target at time t , $y^k(t)$ is the actual output of the k -th output unit at time t .

We define unit l 's error at time t by:

$$e_l(t) := -\frac{\partial E(t)}{\partial \text{net}_l(t)} \quad (5.18)$$

If $l = k$, $e_k(t)$ is the k -th output unit' error at time t :

$$e_k(t) = f'_k(\text{net}_k(t))(t^k(t) - y^k(t)) \quad (5.19)$$

If $l = \text{out}_j$, $e_{\text{out}_j}(t)$ is the error of output gate out_j at time t :

$$e_{\text{out}_j}(t) = f'_{\text{out}_j}(\text{net}_{\text{out}_j}(t)) \left(\sum_{v=1}^{S_j} h(s_{c_j^v}(t)) \sum_k w_{kc_j^v} e_k(t) \right) \quad (5.20)$$

If $l = c_j^v$, $e_{s_{c_j^v}}(t)$ is the error of internal state $s_{c_j^v}$ at time t :

$$e_{s_{c_j^v}}(t) = -\frac{\partial E(t)}{\partial s_{c_j^v}(t)} = f_{\text{out}_j}(\text{net}_{\text{out}_j}(t)) h'(s_{c_j^v}(t)) \sum_k w_{kc_j^v} e_k(t) \quad (5.21)$$

In equations 5.19 5.20 and 5.21, k ranges over output units.

The contribution at time t to w_{lm} 's gradient-based update with learning rate α is defined by:

$$\Delta w_{lm}(t) = -\alpha \frac{\partial E(t)}{\partial w_{lm}} \quad (5.22)$$

CHAPTER 5. BIDIRECTIONAL LONG SHORT-TERM MEMORY FOR PROTEIN SECONDARY STRUCTURE PREDICTION

For weights connecting to output units ($l = k$):

$$\Delta w_{kc_j^v}(t) = \alpha e_k(t) y^{c_j^v}(t) \quad (5.23)$$

For weights connecting to the memory cell ($l = c_j^v$):

$$\Delta w_{c_j^v m}(t) = \alpha e_{s_{c_j^v}}(t) \frac{\partial s_{c_j^v}(t)}{\partial w_{c_j^v m}} \quad (5.24)$$

For weights connecting to the gates ($l = in_j$ or $l = out_j$):

$$\Delta w_{in_j m}(t) = \alpha \sum_{v=1}^{S_j} e_{s_{c_j^v}}(t) \frac{\partial s_{c_j^v}(t)}{\partial w_{in_j m}} \quad (5.25)$$

$$\Delta w_{out_j m}(t) = \alpha e_{out_j}(t) y^m(t-1) \quad (5.26)$$

In equations 5.24 and 5.25, the derivatives of the internal states with respect to weights $w_{c_j^v m}$ and $w_{in_j m}$ are calculated as follows:

$$\frac{\partial s_{c_j^v}(t)}{\partial w_{in_j m}} = \frac{\partial s_{c_j^v}(t-1)}{\partial w_{in_j m}} + g(net_{c_j^v}(t)) f'_{in_j}(net_{in_j}(t)) y^m(t-1) \quad (5.27)$$

$$\frac{\partial s_{c_j^v}(t)}{\partial w_{c_j^v m}} = \frac{\partial s_{c_j^v}(t-1)}{\partial w_{c_j^v m}} + g'(net_{c_j^v}(t)) f_{in_j}(net_{in_j}(t)) y^m(t-1) \quad (5.28)$$

5.3 Bidirectional Long Short-Term Memory

Being unable to take into account future inputs, LSTM is a causal system and hence performs poorly on protein secondary structure prediction [38]. The highest Q3 accuracy achieved by causal LSTM is around 60% [38], which is quite low compared with other algorithms. In our work, we construct a bidirectional LSTM to capture both upstream and downstream information (Figure 5.4). The forward LSTM and backward LSTM update

CHAPTER 5. BIDIRECTIONAL LONG SHORT-TERM MEMORY FOR PROTEIN SECONDARY STRUCTURE PREDICTION

and store upstream context and downstream context respectively. The final output is obtained from a feed-forward subnetwork which combines upstream context, the residue of interest and downstream context.

Figure 5.4: Bidirectional long short-term memory for protein secondary structure prediction. The current input vector $I(t)$ encodes the amino acid at position t . The feed-forward sub-network combines the forward LSTM, the backward LSTM and the current input $I(t)$ to produce an output for position t .

Let t index the position along the amino acid sequence. Starting from $t = 0$, the forward LSTM reads the entire sequence from left to right and updates the states of memory cells and gate units.

$$y^{inj}(0) = 0, y^{inj}(t) = f_{inj}(net_{inj}(t)) = f_{inj}\left(\sum_u w_{inju} y^u(t-1)\right) \text{ for } 1 \leq t \leq L \quad (5.29)$$

$$y^{outj}(0) = 0, y^{outj}(t) = f_{outj}(net_{outj}(t)) = f_{outj}\left(\sum_u w_{outju} y^u(t-1)\right) \text{ for } 1 \leq t \leq L \quad (5.30)$$

CHAPTER 5. BIDIRECTIONAL LONG SHORT-TERM MEMORY FOR PROTEIN SECONDARY STRUCTURE PREDICTION

$$\begin{aligned}
net_{c_j^v}(t) &= \sum_u w_{c_j^v u} y^u(t-1) \text{ for } 1 \leq t \leq L \\
s_{c_j^v}(0) &= 0, s_{c_j^v}(t) = s_{c_j^v}(t-1) + y^{in_j}(t)g(net_{c_j^v}(t)) \text{ for } 1 \leq t \leq L \\
y^{c_j^v}(0) &= 0, y^{c_j^v}(t) = y^{out_j}(t)h(s_{c_j^v}(t)) \text{ for } 1 \leq t \leq L
\end{aligned} \tag{5.31}$$

In the backward LSTM, the states of memory cells and gate units are initialized at $t = L+1$ and updated along the sequence from right to left.

$$\begin{aligned}
y^{in_j}(L+1) &= 0, y^{in_j}(t) = f_{in_j}(net_{in_j}(t)) = f_{in_j}\left(\sum_u w_{in_j u} y^u(t+1)\right) \text{ for } 1 \leq t \leq L \tag{5.32} \\
y^{out_j}(L+1) &= 0, y^{out_j}(t) = f_{out_j}(net_{out_j}(t)) = f_{out_j}\left(\sum_u w_{out_j u} y^u(t+1)\right) \text{ for } 1 \leq t \leq L
\end{aligned} \tag{5.33}$$

$$\begin{aligned}
net_{c_j^v}(t) &= \sum_u w_{c_j^v u} y^u(t+1) \text{ for } 1 \leq t \leq L \\
s_{c_j^v}(L+1) &= 0, s_{c_j^v}(t) = s_{c_j^v}(t+1) + y^{in_j}(t)g(net_{c_j^v}(t)) \text{ for } 1 \leq t \leq L \\
y^{c_j^v}(L+1) &= 0, y^{c_j^v}(t) = y^{out_j}(t)h(s_{c_j^v}(t)) \text{ for } 1 \leq t \leq L
\end{aligned} \tag{5.34}$$

The final output for position t is obtained by combining the states of non-gate units in the forward and backward LSTMs and the current inputs at position t .

$$O^k(t) = f\left(\sum_u w_{ku} y^u(t) + \sum_m w_{km} I^m(t)\right) \tag{5.35}$$

where u indexes any non-gate unit of forward LSTM and backward LSTM, $I^m(t)$ is the m -th unit of the current input vector.

5.4 Experimental Results

5.4.1 Results of BLSTM

In our experiments, we use the PSIPRED set to evaluate the performance of BLSTM on protein secondary structure prediction. We trained several BLSTMs of different numbers

CHAPTER 5. BIDIRECTIONAL LONG SHORT-TERM MEMORY FOR PROTEIN SECONDARY STRUCTURE PREDICTION

of memory blocks and different sizes of memory blocks. Each BLSTM has 20 input units and 3 output units. Forward LSTM and backward LSTM have equal numbers of memory blocks, and each memory block has equal numbers of memory cells. The main results are summarized in Table 5.1.

Table 5.1: The performance of the BLSTMs on protein secondary structure prediction. In the first row, m refers to the number of memory blocks in the LSTM subnetworks, n refers to the number of memory cells in each memory block, and W refers to the total number of weights.

m	n	W	Q_3 (%)			
			Test set 1	Test set 2	Test set 3	Combined test set
8	2	3484	69.07	69.26	70.57	69.60
8	3	5004	70.70	69.41	71.72	70.62
8	4	6780	70.14	70.47	71.23	70.59
9	2	4200	71.09	71.15	71.15	71.13
9	3	6072	70.13	69.73	71.02	70.29
9	4	8268	70.15	70.07	72.29	70.80
10	2	4980	70.75	69.88	72.83	71.14
10	3	7240	70.79	70.45	71.94	71.05
10	4	9900	70.49	70.41	72.06	70.96
11	2	5824	70.98	70.32	73.04	71.43
11	3	8508	70.20	70.94	72.56	71.18
11	4	11676	70.50	70.72	72.51	71.20
12	2	6732	70.68	69.37	73.02	71.02
12	3	9876	70.29	70.61	72.01	70.94
12	4	13596	71.44	70.81	72.33	71.53
13	4	15660	71.43	70.59	72.32	71.45

The highest Q_3 accuracies on different testing sets are achieved by BLSTMs of different sizes. For each testing set, we select the BLSTM of optimal size which yields the highest Q_3 , and give the values of other measures of prediction accuracy in Table 5.2. With the forward/backward LSTM containing 12 memory blocks and each memory block containing 4 memory cells, the BLSTM obtains a best Q_3 accuracy of 71.53% and an SOV of 59.6577 on the Combined test set.

CHAPTER 5. BIDIRECTIONAL LONG SHORT-TERM MEMORY FOR PROTEIN SECONDARY STRUCTURE PREDICTION

Table 5.2: Best results of BLSTM on different testing data sets.

Test set	Test set 1	Test set 2	Test set 3	Combined test set
BLSTM Size	m=12, n=4	m=9, n=2	m=11, n=2	m=12, n=4
Q3(%)	71.44	71.15	73.04	71.53
Q_C(%)	81.35	78.83	77.23	81.08
Q_H(%)	71.50	71.22	74.18	70.74
Q_E(%)	51.82	57.98	62.89	54.83
SOV	59.5777	58.7323	61.3896	59.6577
SOV_C	60.7727	60.6650	60.1571	62.0553
SOV_H	62.9389	58.0459	62.5217	60.1411
SOV_E	51.2242	56.1998	61.8442	54.0219
C_C	0.51529	0.51574	0.53636	0.51659
C_H	0.63195	0.62331	0.62688	0.62566
C_E	0.49679	0.52001	0.57646	0.52261

5.4.2 Results of PBRNN

As a comparison test, several PBRNNs of different architectural sizes is also assessed on the PSIPRED set. The prediction accuracies Q3 and SOV of each PBRNN are recorded in Table 5.3. Among the different PBRNNs evaluated here, we select the one that provides the highest Q3 accuracy, and give its detailed results in Table 5.4. In Table 5.3 and Table 5.4, the PBRNN has 20 input units (i.e. Inum=20) and 3 output units (i.e. Onum=3); Hnum = the number of units in hidden layer H; Fnum (Bnum) = the number of units in layer F (B); FHnum (BHnum) = the number of units in layer FH (BH); Fnum = Bnum and FHnum = BHnum. When Hnum=15, Fnum (Bnum)=10 and FHnum (BHnum)=15, PBRNN obtains its best accuracies of Q3=72.21% and SOV=60.8649 on the entire PSIPRED testing set, i.e. the Combined test set.

CHAPTER 5. BIDIRECTIONAL LONG SHORT-TERM MEMORY FOR PROTEIN SECONDARY STRUCTURE PREDICTION

Table 5.3: Experiment results of PBRNN on protein secondary structure prediction

Hnum	Fnum (Bnum)	FHnum (BHnum)	Q3(%) (SOV)			
			Test set 1	Test set 2	Test set 3	Combined test set
10	10	10	71.05 (58.5027)	68.81 (56.3425)	72.00 (62.1146)	70.66 (58.9017)
10	10	15	70.51 (58.8775)	71.18 (61.0322)	71.50 (61.7579)	71.03 (60.4617)
15	10	10	66.84 (52.0003)	65.61 (55.2007)	70.76 (60.3180)	67.70 (55.4951)
15	10	15	71.95 (59.8224)	71.30 (60.3440)	73.41 (62.5845)	72.21 (60.8649)

Table 5.4: The best results of Pollastri’s BRNN (PBRNN) on the PSIPRED data sets. The architectural size of the PBRNN is: Inum=20, Hnum=15, Onum=3, Fnum (Bnum)=10, and FHnum (BHnum)=15.

Test set	Test set 1	Test set 2	Test set 3	Combined test set
Q3(%)	71.95	71.30	73.41	72.21
Q_C(%)	80.69	79.74	78.49	79.72
Q_H(%)	77.16	71.43	77.09	75.49
Q_E(%)	46.62	56.74	57.14	53.21
SOV	59.8224	60.3440	62.5845	60.8649
SOV_C	58.7954	61.1245	60.1526	59.9516
SOV_H	69.6493	62.6777	70.0166	67.7118
SOV_E	46.0937	55.7286	54.6762	51.9527
C_C	0.52384	0.51965	0.53916	0.52753
C_H	0.63744	0.63038	0.64808	0.63950
C_E	0.50012	0.51407	0.55812	0.52283

5.4.3 Results of the Original BRNN

Being a comparison model, the original BRNNs of different architectural sizes are also trained and tested on the PSIPRED training/testing data set. The results are given in Table 5.5. The original BRNNs of optimal sizes and their best results are provided in Table 5.6. The original BRNN with 20 units in layer F (B) achieves its highest Q3 of 73.67% and its highest SOV of 62.96 on the Combined test set.

5.4.4 Comparison Among BLSTM, PBRNN and the Original BRNN

We summarize the best results of PBRNN, BRNN and BLSTM in Table 5.7. From this table, we observe that BLSTM performs worse than the PBRNN and the original BRNN.

5.5 Discussion

Bidirectional LSTM achieves much higher accuracies than the traditional causal LSTM on PSS prediction. However, the best results of BLSTM are worse than those of the PBRNN and the original BRNN. One reason is that it is hard to detect the sparse and weak long-range signals while ignoring the additional noise found over large distances. And another reason is the limitations of LSTM itself. LSTM is the state of the art in recurrent neural networks. It works well on many artificially generated sequential processing tasks. It can find periodicity in complicated signals, hence it is applicable to cognitive domains such as speech recognition and music composition. However, LSTM's superiority does not carry over to certain time series benchmark problems: the Mackey-Glass series and the Chaotic FIR laser emission series [51]. On these two problems, LSTM performs worse than the

CHAPTER 5. BIDIRECTIONAL LONG SHORT-TERM MEMORY FOR PROTEIN SECONDARY STRUCTURE PREDICTION

Table 5.5: Experimental results of the original BRNN on protein secondary structure prediction. The original BRNN examined has 20 input units and 3 output units, that is Inum=20 and Onum=3. In this table, Fnum (Bnum) represents the number of units in layer F (B), and Fnum is equal to Bnum; Wnum is the total number of weights.

Test set	Test set 1			Test set 2		
Fnum (Bnum)	10	15	20	10	15	20
Wnum	720	1200	1780	720	1200	1780
Q3(%)	71.38	73.26	73.61	72.03	73.00	72.94
Q_C(%)	78.86	79.90	82.45	78.59	81.40	80.17
Q_H(%)	74.55	75.50	74.31	74.42	71.03	73.48
Q_E(%)	51.70	56.70	55.09	57.84	61.17	59.94
SOV	59.84	61.22	62.21	62.34	61.67	62.42
SOV_C	58.04	59.97	61.52	62.73	64.10	63.46
SOV_H	66.50	65.69	67.76	65.32	59.45	62.44
SOV_E	52.21	56.03	54.33	57.62	60.48	60.50
C_C	0.5111	0.5383	0.5432	0.5271	0.5334	0.5292
C_H	0.6306	0.6646	0.6761	0.6487	0.6679	0.6656
C_E	0.4972	0.5301	0.5298	0.5225	0.5510	0.5518

Test set	Test set 3			Combined test set		
Fnum (Bnum)	10	15	20	10	15	20
Wnum	720	1200	1780	720	1200	1780
Q3(%)	72.24	74.33	74.47	71.85	73.52	73.67
Q_C(%)	80.04	81.93	80.24	79.14	80.99	81.06
Q_H(%)	73.80	75.29	75.08	74.26	74.15	74.33
Q_E(%)	54.35	57.86	62.17	54.59	58.58	58.85
SOV	61.23	63.16	64.35	61.06	61.97	62.96
SOV_C	62.21	63.03	63.76	60.80	62.21	62.83
SOV_H	64.13	65.78	66.31	65.35	63.86	65.72
SOV_E	53.68	58.40	61.88	54.52	58.28	58.68
C_C	0.5320	0.5527	0.5594	0.5227	0.5412	0.5441
C_H	0.6208	0.6580	0.6520	0.6334	0.6640	0.6656
C_E	0.5365	0.5864	0.5937	0.5184	0.5545	0.5573

CHAPTER 5. BIDIRECTIONAL LONG SHORT-TERM MEMORY FOR PROTEIN SECONDARY STRUCTURE PREDICTION

Table 5.6: The best results of the original BRNN on different testing sets. The original BRNN has 20 input units and 3 output units, that is $I_{num}=20$ and $O_{num}=3$. In this table, F_{num} (B_{num}) represents the number of units in layer F (B), and F_{num} is equal to B_{num} . W_{num} is the total number of weights.

Test set	Test set 1	Test set 2	Test set 3	Combined test set
F_{num} (B_{num})	20	15	20	20
W_{num}	1780	1200	1780	1780
Q3(%)	73.61	73.00	74.47	73.67
Q_C(%)	82.45	81.40	80.24	81.06
Q_H(%)	74.31	71.03	75.08	74.33
Q_E(%)	55.09	61.17	62.17	58.85
SOV	62.21	61.67	64.35	62.96
SOV_C	61.52	64.10	63.76	62.83
SOV_H	67.76	59.45	66.31	65.72
SOV_E	54.33	60.48	61.88	58.68
C_C	0.5432	0.5334	0.5594	0.5441
C_H	0.6761	0.6679	0.6520	0.6656
C_E	0.5298	0.5510	0.5937	0.5573

MLP. LSTM learned to tune into the fundamental oscillation of each series but was unable to accurately follow the signal. It could not predict the collapse of emission in the FIR-laser task. Gers' results suggest to use LSTM only on tasks where traditional time window based approaches must fail [51]. For the problem of protein secondary structure prediction, many window-based MLPs have yielded impressive results. The LSTM used here is a Gradient-trained LSTM. In late 2006, an EVOLINO-trained LSTM is newly proposed to circumvent the limitations of Gradient-LSTM. EVOLINO-LSTM is able to learn the Mackey-Glass time series. Therefore, we suggest to improve PSS prediction by using a bidirectional EVOLINO-LSTM.

CHAPTER 5. BIDIRECTIONAL LONG SHORT-TERM MEMORY FOR PROTEIN SECONDARY STRUCTURE PREDICTION

Table 5.7: Comparison among PBRNN, BRNN and BLSTM.

measure	Test	PBRNN	BRNN	BLSTM
Q3(%)	Test set 1	71.95	73.61	71.44
	Test set 2	71.30	72.94	70.81
	Test set 3	73.41	74.47	72.33
	Combined test set	72.21	73.67	71.53
Q-C(%)	Test set 1	80.69	82.45	81.35
	Test set 2	79.74	80.17	83.74
	Test set 3	78.49	80.24	78.04
	Combined test set	79.72	81.06	81.08
Q-H(%)	Test set 1	77.16	74.31	71.50
	Test set 2	71.43	73.48	67.19
	Test set 3	77.09	75.08	72.91
	Combined test set	75.49	74.33	70.74
Q-E(%)	Test set 1	46.62	55.09	51.82
	Test set 2	56.74	59.94	53.33
	Test set 3	57.14	62.17	60.22
	Combined test set	53.21	58.85	54.83
SOV	Test set 1	59.8224	62.2104	59.5777
	Test set 2	60.3440	62.4237	58.2764
	Test set 3	62.5845	64.3487	61.1379
	Combined test set	60.8649	62.9593	59.6577
SOV_C	Test set 1	58.7954	61.5206	60.7727
	Test set 2	61.1245	63.4552	62.6127
	Test set 3	60.1526	63.7589	63.0892
	Combined test set	59.9516	62.8277	62.0553
SOV_H	Test set 1	69.6493	67.7551	62.9389
	Test set 2	62.6777	62.4404	56.7479
	Test set 3	70.0166	66.3065	60.1386
	Combined test set	67.7118	65.7217	60.1411
SOV_E	Test set 1	46.0937	54.3339	51.2242
	Test set 2	55.7286	60.4976	52.6909
	Test set 3	54.6762	61.8782	59.0713
	Combined test set	51.9527	58.6804	54.0219
C_C	Test set 1	0.52384	0.54317	0.51529
	Test set 2	0.51965	0.52916	0.50867
	Test set 3	0.53916	0.55937	0.52669
	Combined test set	0.52753	0.54407	0.51659
C_H	Test set 1	0.63744	0.67606	0.63195
	Test set 2	0.63038	0.66561	0.62223
	Test set 3	0.64808	0.65197	0.61903
	Combined test set	0.63950	0.66564	0.62566
C_E	Test set 1	0.50012	0.52975	0.49679
	Test set 2	0.51407	0.55182	0.51801
	Test set 3	0.55812	0.59369	0.55820
	Combined test set	0.52283	0.55726	0.52261

Chapter 6

Segmented-Memory Recurrent Neural Network for Long-Term Dependencies

In this chapter, we propose a novel recurrent architecture called Segmented-Memory Recurrent Neural Network (SMRNN) and develop a gradient-based learning strategy to construct the SMRNN. The SMRNN tries to memorize long sequences in segments like human beings do. The input sequence is broken into segments and then presented to the SMRNN one symbol per cycle. The SMRNN memorizes each segment first and then cascades them to form the final sequence. The SMRNN uses separate internal states to store symbol-level context, as well as segment-level context. The symbol-level context is updated for each symbol presented for input. The segment-level context is updated after each segment. Our theoretical analysis and experimental results indicate that SMRNN outperforms conventional RNNs on the long-term dependency problems.

6.1 Segmented-Memory Recurrent Neural Network

6.1.1 Segmented Memory

As discussed in the previous chapter, the memory of recurrent neural networks is limited. If the network reads in a long sequence at a time, it tends to forget the head part of the sequence. Human memory is similar in that we also have difficulty memorizing very long sequences. As we observe, when people are memorizing some long numbers or long sentences, they tend to do so in segments. During the process of human memorization of a long sequence, people tend to break the sequence into a few segments, whereby people memorize each segment first and then cascade them to form the final sequence [44, 61, 118, 134, 145]. Here are several examples. When we are learning a new word, we often divide it into syllables, e.g. “dependency” into “de-pen-den-cy”, and memorize them sequentially. Similarly, we memorize a sentence phrase by phrase. Another typical example is the ratio of the circumference to the diameter of a circle, i.e. $\pi = 3.1415926535897932384626433832795....$ We can firstly try to memorize the first segment of 10 numbers, then the second segment of the same length and so on.

The process of memorizing a sequence in segments is illustrated in Figure 6.1. The

Figure 6.1: Illustration of segmented memory.

CHAPTER 6. SEGMENTED-MEMORY RECURRENT NEURAL NETWORK FOR LONG-TERM DEPENDENCIES

substrings in parentheses represent segments of length $d1$, $d2$ and $d3$; gray arrows indicate the update of contextual information associated to symbols and black arrows indicate the update of contextual information associated to segments; numbers under the arrows indicate the sequencing of memorization. $d1$, $d2$ and $d3$ are not necessarily equal to one another. The segment length can be fixed or vary from segment to segment.

Based on the observation on human memorization, we believe that RNNs are more capable of capturing long-term dependencies if they have segmented-memory and imitate the way of human memorization. Following this intuitive idea, we propose a novel recurrent network called *segmented-memory recurrent neural network* (abbreviated as SMRNN).

6.1.2 Network Architecture

The architecture of SMRNN is illustrated in Figure 6.2. The SMRNN has two hidden layers,

Figure 6.2: Segmented-memory recurrent neural network.

namely layer H1 and layer H2. The layer H1 represents symbol-level state and the layer H2 represents segment-level state. Layer H1 and layer H2 both have recurrent connections

CHAPTER 6. SEGMENTED-MEMORY RECURRENT NEURAL NETWORK FOR LONG-TERM DEPENDENCIES

among themselves. The symbol-level state and segment-level state at the previous cycle are fed back and stored in context layer S1 and context layer S2 respectively. Context layer S1 and context layer S2 store contextual information at symbol level and at segment level respectively. Most importantly, we introduce into the network a new attribute vector *interval* $\mathbf{d} = (d_1, d_2, \dots, d_i, \dots, d_n)$ to denote the segment length that can be fixed or variable.

6.1.3 Illustration of Dynamics

We now explain the dynamics of SMRNN with an example (Figure 6.3). Figure 6.3 illustrates that the symbol-level state is updated on a symbol-by-symbol basis, whereas the segment-level state is changed in a segment-by-segment fashion.

Figure 6.3: Dynamics of segmented-memory recurrent neural network.

The input sequence $u^1, u^2, \dots, u^{d1+d2+d3}$ in Figure 6.3 is divided into segments of length $d1$, $d2$ and $d3$. At position t that is not the beginning of a segment, symbol u^t from the input sequence together with the previous symbol-level state x^{t-1} are fed to hidden layer H1 to update the symbol-level state x^t ; at position t that is the beginning of a segment, x^t is updated by combining x^0 and u^t . At position t that is the end of a segment, the symbol-level state x^t together with the previous segment-level state y^{t-1} are forwarded to the next layer H2 to update the segment-level state y^t ; at position t that is not the end

CHAPTER 6. SEGMENTED-MEMORY RECURRENT NEURAL NETWORK FOR LONG-TERM DEPENDENCIES

of a segment, the segment-level state y^t remains unchanged, i.e. $y^t = y^{t-1}$. This process continues until it reaches the end of the input sequence. Next, the segment-level state is forwarded to the output layer to generate the final output. In other words, the SMRNN reads one symbol per cycle as conventional RNNs do; the state of H1 is updated at the coming of each single symbol, while the state of H2 is updated only at the end of each segment. The segment-level state layer behaves as if it sequentially cascades segments to obtain the final sequence as people often do: Every time when people finish one segment, they usually go over the subsequence from the first segment to the segment which they have recently memorized. This process possibly makes sure that they have remembered all the previous segments in correct order (Figure 6.1).

6.1.4 Formulation of Dynamics

In this section, we formulate the dynamics of SMRNN to implement the segmented memory illustrated in Figure 6.1.

The initial symbol-level state is computed by $x_k^0 = g(\sigma_k^0)$, where g is a sigmoid function and σ_k^0 is a parameter to be optimized during training. Suppose a sequence of symbols is fed to SMRNN one symbol per cycle, the symbol-level state is updated at the coming of each symbol. At the beginning of each segment, vector $\mathbf{x}^t$ is obtained from initial state $\mathbf{x}^0$ and input $\mathbf{u}^t$; while at other positions, $\mathbf{x}^t$ is obtained from previous state $\mathbf{x}^{t-1}$ and input $\mathbf{u}^t$. Let u_i^t be the i -th input at cycle t , the symbol-level state at cycle t is calculated by:

For $k = 1, \dots, n_X$

$$x_k^t = \begin{cases} g(\sum_{j=1}^{n_X} W_{kj}^{xx} x_j^0 + \sum_{i=1}^{n_U} W_{ki}^{xu} u_i^t) & \text{if } t = SH \\ g(\sum_{j=1}^{n_X} W_{kj}^{xx} x_j^{t-1} + \sum_{i=1}^{n_U} W_{ki}^{xu} u_i^t) & \text{otherwise} \end{cases} \quad (6.1)$$

CHAPTER 6. SEGMENTED-MEMORY RECURRENT NEURAL NETWORK FOR LONG-TERM DEPENDENCIES

where SH (Segment Head) refers to the beginning of a segment.

The segment-level state is initialized at the beginning of the input sequence by $y_k^0 = g(v_k^0)$, where v_k^0 is a parameter to be trained. Due to the insertion of intervals into the memory of context, the segment-level state is updated only at the end of each segment.

For $k = 1, \dots, n_Y$,

$$y_k^t = \begin{cases} g(\sum_{j=1}^{n_Y} W_{kj}^{yy} y_j^{t-1} + \sum_{i=1}^{n_X} W_{ki}^{yx} x_i^t) & \text{if } t = ST \\ y_k^{t-1} & \text{otherwise} \end{cases} \quad (6.2)$$

where ST (Segment Tail) denotes the end of a segment.

Contextual information contained in vector $\mathbf{y}^t$ is forwarded to the output layer to produce an output.

For $k = 1, \dots, n_Z$,

$$z_k^t = g(\sum_{j=1}^{n_Y} W_{kj}^{zy} y_j^t) \quad (6.3)$$

For the sake of clarity, we summarize the significance of the symbols in the above equations.

- y_j^{t-1} denotes the previous state of the j -th unit at hidden layer H2. It is stored in the j -th unit at context layer S2.
- x_j^{t-1} denotes the previous state of the j -th unit at hidden layer H1. It is stored in the j -th unit at context layer S1.
- n_Z , n_Y , n_X and n_U denote the numbers of neurons at output layer, hidden layer H2 (context layer S2), hidden layer H1 (context layer S1) and input layer respectively.
- W_{kj}^{zy} denotes the connection from the j th neuron at hidden layer H2 to the k th neuron at output layer.

CHAPTER 6. SEGMENTED-MEMORY RECURRENT NEURAL NETWORK FOR LONG-TERM DEPENDENCIES

- W_{kj}^{yy} denotes the connection from the j th neuron at context layer S2 to the k th neuron at hidden layer H2.
- W_{ki}^{yx} denotes the connection from the i th neuron at hidden layer H1 to the k th neuron at hidden layer H2.
- W_{kj}^{xx} denotes the connection from the j th neuron at context layer S1 to the k th neuron at hidden layer H1.
- W_{ki}^{xu} denotes the connection from the i th neuron at input layer to the k th neuron at hidden layer H1.
- $g(x) = 1/(1 + \exp(-x))$.

Usually the number of input neurons is equal to the size of the input symbol set. The number of output units is chosen according to different applications. Symbols in the sequence are presented to the input layer with one-hot coding [23]. When a_l (the l -th symbol in the input symbol set) is read, the k -th element of the input vector is $I_k = \delta(k, l)$ ($\delta(k, l)$ is 1 if $k = l$ and 0 otherwise). Normally the synaptic input equals the sum of inputs multiplied by weights. Hence, if an input element is zero, the weights associated to that input unit will not be updated during training. Thus we apply a contractive mapping $f(x) = \epsilon + (1 - 2\epsilon)x$, with ϵ a small positive number, to input units.

6.1.5 SMRNN Learning Strategy

The segmented-memory recurrent neural network is trained using an extension of the Real Time Recurrent Learning algorithm. Usually, the weights are initialized with random values

CHAPTER 6. SEGMENTED-MEMORY RECURRENT NEURAL NETWORK FOR LONG-TERM DEPENDENCIES

and then learned, but the initial states of the hidden neurons x_k and y_k do not change during training. As remarked in [42], it is not reasonable to make the initial states fixed, and the behavior of the network improves if they are also learned. In order to keep x_k^t and y_k^t within the range $[0, 1]$ during gradient descent, we define the synaptic inputs of x_k^t and y_k^t to be $\sigma_k^t = \sum_{j=1}^{n_x} W_{kj}^{xx} x_j^{t-1} + \sum_{i=1}^{n_U} W_{ki}^{xu} u_i^t$ and $v_k^t = \sum_{j=1}^{n_Y} W_{kj}^{yy} y_j^{t-1} + \sum_{i=1}^{n_X} W_{ki}^{yx} x_i^t$ such that $x_k^t = g(\sigma_k^t)$ and $y_k^t = g(v_k^t)$, and take σ_k^0 and v_k^0 as the parameters to be optimized.

Gradient descent learning is typically based on minimizing the sum-of-squared-error cost function:

$$E^t = \sum_{k=1}^{n_z} \frac{1}{2} (z_k^t - \bar{z}_k^t)^2, \quad (6.4)$$

where $\bar{z}_k^t$ is the desired output and z_k^t is the actual output. Every parameter P , including W_{kj}^{zy} , W_{kj}^{yy} , W_{ki}^{yx} , W_{kj}^{xx} , W_{ki}^{xu} , σ_k^0 and v_k^0 , is initialized with small random values then updated according to gradient descent:

$$\Delta P = -\alpha \frac{\partial E^t}{\partial P} + \eta \Delta' P, \quad (6.5)$$

with a learning rate α and a momentum term η . The value $\Delta' P$ is the variation of P in the previous iteration.

The derivative of E^t with respect to W_{kj}^{zy} can be calculated in one single step:

$$\frac{\partial E^t}{\partial W_{kj}^{zy}} = (z_k^t - \bar{z}_k^t) z_k^t (1 - z_k^t) y_j^t. \quad (6.6)$$

While the derivatives with respect to the other parameters (denoted by P) need much more computation:

$$\frac{\partial E^t}{\partial P} = \sum_{a=1}^{n_z} (z_a^t - \bar{z}_a^t) z_a^t (1 - z_a^t) \sum_{b=1}^{n_Y} W_{ab}^{zy} y_b^t (1 - y_b^t) \frac{\partial v_b^t}{\partial P}. \quad (6.7)$$

CHAPTER 6. SEGMENTED-MEMORY RECURRENT NEURAL NETWORK FOR LONG-TERM DEPENDENCIES

The derivative of synaptic input v_b^t with respect to parameter P is computed in a recurrent way. At cycle $t=0$, the initial derivative with respect to v_k^0 is calculated by $\frac{\partial v_b^0}{\partial v_k^0} = \delta(b, k)$ where $\delta(b, k)$ denotes the Kronecker delta function ($\delta(b, k)$ is equal to 1 if $b = k$ and 0 otherwise); for the other parameters, the initial derivatives are set to be zero, that is, $\frac{\partial v_b^0}{\partial P} = 0$. At cycle $t > 0$, the derivatives of v_b^t with respect to W_{kj}^{yy} , W_{ki}^{yx} and v_k^0 are calculated respectively using the following equations.

$$\frac{\partial v_b^t}{\partial W_{kj}^{yy}} = \delta(b, k) y_j^{t-1} + \sum_{a=1}^{n_Y} W_{ba}^{yy} y_a^{t-1} (1 - y_a^{t-1}) \frac{\partial v_a^{t-1}}{\partial W_{kj}^{yy}}, \quad (6.8)$$

$$\frac{\partial v_b^t}{\partial W_{ki}^{yx}} = \delta(b, k) x_i^t + \sum_{a=1}^{n_Y} W_{ba}^{yy} y_a^{t-1} (1 - y_a^{t-1}) \frac{\partial v_a^{t-1}}{\partial W_{ki}^{yx}}, \quad (6.9)$$

$$\frac{\partial v_b^t}{\partial v_k^0} = \sum_{a=1}^{n_Y} W_{ba}^{yy} y_a^{t-1} (1 - y_a^{t-1}) \frac{\partial v_a^{t-1}}{\partial v_k^0}, \quad (6.10)$$

The iteration for calculating the derivatives of v_b^t with respect to σ_k^0 , W_{kj}^{xx} and W_{ki}^{xu} that are all denoted by P , is given as follows:

$$\frac{\partial v_b^t}{\partial P} = \sum_{a=1}^{n_Y} W_{ba}^{yy} y_a^{t-1} (1 - y_a^{t-1}) \frac{\partial v_a^{t-1}}{\partial P} + \sum_{a=1}^{n_X} W_{ba}^{yx} x_a^t (1 - x_a^t) \frac{\partial \sigma_a^t}{\partial P}. \quad (6.11)$$

The derivatives of σ_a^t with respect to σ_k^0 , W_{kj}^{xx} and W_{ki}^{xu} are also calculated by iteration.

$$\frac{\partial \sigma_a^0}{\partial \sigma_k^0} = \delta(a, k), \quad (6.12)$$

$$\frac{\partial \sigma_a^t}{\partial \sigma_k^0} = \sum_{b=1}^{n_X} W_{ab}^{xx} x_b^{t-1} (1 - x_b^{t-1}) \frac{\partial \sigma_b^{t-1}}{\partial \sigma_k^0}, \quad (6.13)$$

$$\frac{\partial \sigma_a^0}{\partial W_{kj}^{xx}} = 0, \quad (6.14)$$

$$\frac{\partial \sigma_a^t}{\partial W_{kj}^{xx}} = \sum_{b=1}^{n_x} W_{ab}^{xx} x_b^{t-1} (1 - x_b^{t-1}) \frac{\partial \sigma_b^{t-1}}{\partial W_{kj}^{xx}} + \delta(a, k) x_j^{t-1}, \quad (6.15)$$

$$\frac{\partial \sigma_a^0}{\partial W_{ki}^{xu}} = 0, \quad (6.16)$$

$$\frac{\partial \sigma_a^t}{\partial W_{ki}^{xu}} = \sum_{b=1}^{n_x} W_{ab}^{xx} x_b^{t-1} (1 - x_b^{t-1}) \frac{\partial \sigma_b^{t-1}}{\partial W_{ki}^{xu}} + \delta(a, k) u_i^t. \quad (6.17)$$

6.2 An Analytical Explanation of SMRNN Behavior

In this section, we theoretically explain how segmented-memory mitigates the problem of vanishing gradient, by studying the gradient flows in the Elman's network and in the SMRNN. For simplicity and better representation, we consider a simple Elman's network with single node in each layer (Figure 6.4), and a simplified SMRNN with single node in each layer and with fixed interval d (Figure 6.5).

The Elman's network is of a standard RNN architecture. It is a two-layer back-propagation network, with the addition of a feedback connection from the output of the hidden layer to its input. In the Elman's network, the state of hidden neuron $y(t)$ is updated by:

$$y(t) = g(W^{yy}y(t-1) + W^{yx}x(t)) \quad (6.18)$$

Assuming that the error function E is calculated at position T of the input sequence, the

CHAPTER 6. SEGMENTED-MEMORY RECURRENT NEURAL NETWORK FOR LONG-TERM DEPENDENCIES

Figure 6.4: A simple Elman's network.

Figure 6.5: A simple SMRNN.

derivatives of the error function with respect to W^{yy} and W^{yx} are given by:

$$\frac{\partial E(T)}{\partial W} = \sum_{\tau \leq T} \frac{\partial E(T)}{\partial y(\tau)} \frac{\partial y(\tau)}{\partial W} = \sum_{\tau \leq T} \frac{\partial E(T)}{\partial y(T)} \frac{\partial y(T)}{\partial y(\tau)} \frac{\partial y(\tau)}{\partial W} \quad (6.19)$$

where W represents W^{yy} or W^{yx} . The partial derivative of $y(T)$ with respect to $y(\tau)$ is calculated by

$$\frac{\partial y(T)}{\partial y(\tau)} = \frac{\partial y(T)}{\partial y(T-1)} \frac{\partial y(T-1)}{\partial y(T-2)} \cdots \frac{\partial y(\tau+1)}{\partial y(\tau)} = \prod_{t=\tau+1}^T \frac{\partial y(t)}{\partial y(t-1)} \quad (6.20)$$

As stated by Bengio, Y. [13], the norm of each factor on the right-hand side must be less than 1 in order for a recurrent architecture to robustly latch information. Hence, the left-hand side converges exponentially fast to zero as $T - \tau$ increases. That is, for a term of the sum with $\tau \ll T$, we have

$$\left| \frac{\partial y(T)}{\partial y(\tau)} \right| \rightarrow 0 \quad (6.21)$$

For the SMRNN with fixed interval d , the input sequence is divided into segments of equal length d and presented to the SMRNN one symbol per cycle. The symbol-level state

CHAPTER 6. SEGMENTED-MEMORY RECURRENT NEURAL NETWORK FOR LONG-TERM DEPENDENCIES

at cycle t is calculated by:

$$x(t) = \begin{cases} g(W^{xx}x(0) + W^{xu}u(t)) , & \text{if } t = nd + 1 \\ g(W^{xx}x(t-1) + W^{xu}u(t)) , & \text{otherwise} \end{cases} \quad (6.22)$$

where $x(0) = g(\sigma(0))$. At cycle $t = 0$, $y(0)$ is initialized by $y(0) = g(v(0))$; at cycle $t \geq 1$, the segment-level state is updated by

$$y(t) = \begin{cases} g(W^{yy}y(t-1) + W^{yx}x(t)) , & \text{if } t = nd \text{ or } t = L \\ y(t-1) , & \text{otherwise} \end{cases} \quad (6.23)$$

where L is the length of the input sequence.

Similarly, the derivatives of the error function with respect to W^{yy} and W^{yx} in the SMRNN are computed by equation 6.19 and the partial derivative $\frac{\partial y(T)}{\partial y(\tau)}$ is calculated by equation 6.20.

According to equation 6.23, the derivative $\frac{\partial y(t)}{\partial y(t-1)}$ is equal to 1 when $t \neq nd$ and $t \neq L$. Suppose $T = n_1d + m_1$ with $1 \leq m_1 \leq d$ and $\tau = n_2d + m_2$ with $1 \leq m_2 \leq d$, the partial derivative $\frac{\partial y(T)}{\partial y(\tau)}$ has a special form for SMRNN:

$$\begin{aligned} \frac{\partial y(T)}{\partial y(\tau)} &= \frac{\partial y(T)}{\partial y(T-1)} \frac{\partial y(n_1d)}{\partial y(n_1d-1)} \cdots \frac{\partial y((n_2+1)d)}{\partial y((n_2+1)d-1)} \\ &= \frac{\partial y(T)}{\partial y(T-1)} \prod_{n=n_2+1}^{n_1} \frac{\partial y(nd)}{\partial y(nd-1)} \end{aligned} \quad (6.24)$$

Similarly, the necessary condition for the SMRNN to robustly latch information is $|\frac{\partial y(nd)}{\partial y(nd-1)}| <$

$\lambda < 1$. So when $\tau \ll T$, the norm of $\frac{\partial y(T)}{\partial y(\tau)}$ also converges to zero.

$$\left| \frac{\partial y(T)}{\partial y(\tau)} \right| = \left| \frac{\partial y(T)}{\partial y(T-1)} \right| \prod_{n=n_2+1}^{n_1} \left| \frac{\partial y(nd)}{\partial y(nd-1)} \right| < \left| \frac{\partial y(T)}{\partial y(T-1)} \right| \lambda^{n_1-n_2} \rightarrow 0 \quad (6.25)$$

As $T - \tau$ increases, the ratio of the partial derivative of SMRNN to that of Elman's network also converges to zero. That is,

$$\frac{\left| \frac{\partial y(T)}{\partial y(\tau)} \right|_{Elman}}{\left| \frac{\partial y(T)}{\partial y(\tau)} \right|_{SMRNN}} = \frac{\left| \prod_{t=\tau+1}^T \frac{\partial y(t)}{\partial y(t-1)} \right|}{\left| \frac{\partial y(T)}{\partial y(T-1)} \prod_{n=n_2+1}^{n_1} \frac{\partial y(nd)}{\partial y(nd-1)} \right|}$$

CHAPTER 6. SEGMENTED-MEMORY RECURRENT NEURAL NETWORK FOR LONG-TERM DEPENDENCIES

$$= \prod_{t=\tau+1, t \neq nd}^{T-1} \left| \frac{\partial y(t)}{\partial y(t-1)} \right| < \lambda^{T-\tau-(n_1-n_2)} \rightarrow 0 \quad (6.26)$$

Therefore, the partial derivative $\frac{\partial y(T)}{\partial y(\tau)}$ in SMRNN vanishes slower than that of the Elman's network.

In SMRNN, there are two more groups of parameters, namely W^{xx} and W^{xu} . The derivatives with respect to W^{xx} and W^{xu} are given by:

$$\frac{\partial E(T)}{\partial W} = \sum_{\tau \leq T} \frac{\partial E(T)}{\partial x(\tau)} \frac{\partial x(\tau)}{\partial W} = \sum_{\tau \leq T} \frac{\partial E(T)}{\partial y(T)} \frac{\partial y(T)}{\partial x(\tau)} \frac{\partial x(\tau)}{\partial W} \quad (6.27)$$

where W represents W^{xx} or W^{xu} . The partial derivative $\frac{\partial y(T)}{\partial x(\tau)}$ is given by

$$\frac{\partial y(T)}{\partial x(\tau)} = \sum_{t=\tau}^T \frac{\partial y(T)}{\partial y(t)} \frac{\partial y(t)}{\partial x(t)} \frac{\partial x(t)}{\partial x(\tau)} \quad (6.28)$$

According to equations 6.22 and 6.23, when $t = nd+1$, $\frac{\partial x(t)}{\partial x(t-1)} = 0$; when $t \neq nd$ and $t \neq L$, the factor $\frac{\partial y(t)}{\partial x(t)} = 0$ and $\frac{\partial y(t)}{\partial y(t-1)} = 1$. Thus for different ranges of interval d , the partial derivative $\frac{\partial y(T)}{\partial x(\tau)}$ has different forms as below:

1) $d \geq T$: $\frac{\partial y(t)}{\partial x(t)} \neq 0$ only when $t = T$, therefore,

$$\begin{aligned} \frac{\partial y(T)}{\partial x(\tau)} &= \frac{\partial y(T)}{\partial y(T)} \frac{\partial y(T)}{\partial x(T)} \frac{\partial x(T)}{\partial x(\tau)} = \frac{\partial y(T)}{\partial x(T)} \frac{\partial x(T)}{\partial x(\tau)} \\ &= \frac{\partial y(T)}{\partial x(T)} \prod_{t'=\tau+1}^T \frac{\partial x(t')}{\partial x(t'-1)} \end{aligned} \quad (6.29)$$

with $T - \tau + 1$ factors in the product.

2) $1 < d < T$: $\frac{\partial x(t)}{\partial x(\tau)} \neq 0$ for $n_2d + d \leq t \leq \tau$ and $\frac{\partial y(t)}{\partial x(t)} \neq 0$ for $t = nd$, therefore the term of the sum is not zero only when $t = n_2d + d$:

$$\frac{\partial y(T)}{\partial x(\tau)} = \frac{\partial y(T)}{\partial y(n_2d + d)} \frac{\partial y(n_2d + d)}{\partial x(n_2d + d)} \frac{\partial x(n_2d + d)}{\partial x(\tau)}$$

CHAPTER 6. SEGMENTED-MEMORY RECURRENT NEURAL NETWORK FOR LONG-TERM DEPENDENCIES

$$= \frac{\partial y(n_2d + d)}{\partial x(n_2d + d)} \frac{\partial y(T)}{\partial y(T-1)} \prod_{n=n_2+2}^{n_1} \frac{\partial y(nd)}{\partial y(nd-1)} \prod_{t'=\tau+1}^{n_2d+d} \frac{\partial x(t')}{\partial x(t'-1)} \quad (6.30)$$

with $n_1 - n_2 + d - m_2$ factors in the product.

3) $d = 1$: $\frac{\partial x(t)}{\partial x(\tau)} \neq 0$ for $t = \tau$ only, hence

$$\begin{aligned} \frac{\partial y(T)}{\partial x(\tau)} &= \frac{\partial y(T)}{\partial y(\tau)} \frac{\partial y(\tau)}{\partial x(\tau)} \frac{\partial x(\tau)}{\partial x(\tau)} = \frac{\partial y(T)}{\partial y(\tau)} \frac{\partial y(\tau)}{\partial x(\tau)} \\ &= \frac{\partial y(\tau)}{\partial x(\tau)} \prod_{t'=\tau+1}^T \frac{\partial y(t')}{\partial y(t'-1)} \end{aligned} \quad (6.31)$$

with $T - \tau + 1$ factors in the product.

The equation 6.30 has less factors on its right-hand side than equations 6.29 and 6.31. Therefore, the gradient $\frac{\partial y(T)}{\partial x(\tau)}$ decays most slowly when $1 < d < T$. With $d > T$, SMRNN is an RNN with multiple hidden layers and without segmented-memory; with $d = 1$, SMRNN is an RNN with multiple feedback connections and without segmented-memory. The fact that SMRNN with $1 < d < T$ is superior implies that the advantage of SMRNN over standard RNNs is segmented-memory instead of multiple hidden layers and multiple feedback connections.

On the right-hand side of equation 6.30, there is a tradeoff between the number of $\frac{\partial y(nd)}{\partial y(nd-1)}$ and that of $\frac{\partial x(t')}{\partial x(t'-1)}$. Large d leads to less $\frac{\partial y(nd)}{\partial y(nd-1)}$ but more $\frac{\partial x(t')}{\partial x(t'-1)}$; small d results in more $\frac{\partial y(nd)}{\partial y(nd-1)}$ but less $\frac{\partial x(t')}{\partial x(t'-1)}$. The length of interval d has an effect on the performance of SMRNN. SMRNN fails to bridge long time lags when the interval d is too small or too large.

Based on the above analysis, we conclude that the gradients in SMRNN vanish at a slower pace than the gradients in standard RNNs. This advantage allows SMRNN to deal

with long-term dependencies effectively. We will further validate the analysis in this section with experimental results in the next section.

6.3 Implementation and Evaluation

For testing the performance of SMRNN, we choose two well-known bench-mark problems. They are (i) the information latching problem, (ii) the “2-sequence problem”.

6.3.1 The Information Latching Problem

Bengio, Y. [13] introduced approaches to study the strengths and weaknesses of the capability of a model to capture dependencies of output on much earlier inputs. Specifically, they designed a minimal task, i.e. the information latching problem, as a test that must be passed in order for a dynamic system to latch information robustly [13]. The term “information latching” in this context refers to the long-term storage of definite bits of information into the state variables of a dynamic system.

The information latching problem is to classify two different classes of sequences. The class of sequence $X_1, X_2, \dots, X_T$ depends only on the first L values of the sequence:

$$C(X_1, X_2, \dots, X_T) = C(X_1, X_2, \dots, X_L) \in \{0, 1\} (L < T) , \quad (6.32)$$

Sequence $X_1, X_2, \dots, X_T$ is classified as member if its prefix $X_1, X_2, \dots, X_L$ is identical to a predefined string $Y_1, Y_2, \dots, Y_L$, and nonmember otherwise. That is $C(X_1, X_2, \dots, X_T) = 1$ if $X_t = Y_t$ for $t = 1, \dots, L$, and $C(X_1, X_2, \dots, X_T) = 0$ otherwise. The values $X_{L+1}, \dots, X_T$ are irrelevant for determining the class of the sequence, however, they may affect the evolution of the dynamic system and eventually erase the internally stored information

CHAPTER 6. SEGMENTED-MEMORY RECURRENT NEURAL NETWORK FOR LONG-TERM DEPENDENCIES

about the initial values of the input. We suppose L fixed and allow sequences of arbitrary length $T \gg L$. Thus, the problem can be solved only if the dynamic system is able to store information about the initial input values for an arbitrary duration. This is the simplest form of long-term computation that one may ask a recurrent network to carry out.

In our experiments, we keep L fixed and vary T in increments of five or ten. By increasing T , we will be able to control the span of long-term dependencies. For each value of T , we randomly generate two sets of sequences, one for training and the other for testing. In each set, the number of member sequences is equal to that of nonmembers. Each sequence consists of letters of the English alphabet. The training sequence is presented to the recurrent neural networks, one symbol at a time. At the end of each training sequence, the error function is computed by $E = \frac{1}{2}(z - \bar{z})^2$, where z is the actual output and $\bar{z}$ is the target output. This error function has a minimum when the actual output is the probability that the sequence is in the class. The accuracy is defined as the ratio of correctly classified testing sequences to the total number of testing sequences.

Choosing Error Threshold

In order to choose an appropriate threshold that decides when to stop training, a simple test is conducted on the Elman's network. From the results illustrated in Table 6.1, we observe that if the threshold is set to be 0.01, which is too small, the network tends to get over-fitting. Thus the error threshold is set to be 0.1.

CHAPTER 6. SEGMENTED-MEMORY RECURRENT NEURAL NETWORK FOR LONG-TERM DEPENDENCIES

Table 6.1: The performance of the Elman’s network on the information latching problem by using different error threshold values.

L	T	training set size	testing set size	threshold	accuracy
10	15	30	10	0.1	80%
10	15	30	10	0.01	70%
50	55	100	40	0.1	72.5%
50	55	100	40	0.01	65%

Effect of Training Data Set Size

We test the effect of training data set size on an SMRNN with interval $d = 15$ and each hidden layer having 10 neurons. Sequences of length 70 are used. The results are given in Table 6.2. Table entries refer to means of 10 trials. As illustrated in this table, SMRNN learns faster and better when more samples are used for training, regardless of the testing data set size.

Table 6.2: Information latching by SMRNN using different sizes of training data set.

training set size	testing set size	epoches	accuracies
50	50	2196.3	65.6%
100	100	1789.2	84.80%
150	150	1503.5	97.47%

In our experiments, we found that the longer sequences are used, the more training samples are required to achieve a satisfactory level of generalization. Thus under circumstances whereby the network has small training error and relatively low testing accuracy, it is necessary to enlarge the training data set.

6.3.2 SMRNN vs. Elman’s Network on the Information Latching Problem

Researchers have proposed several architectural approaches for the long-term dependency problems. They compared their network architectures against Elman’s network, which is a standard architecture of recurrent neural network [60, 62, 81, 80, 23]. In our experiments, we also take Elman’s network for comparison.

We carry out the first experiment on an SMRNN with each hidden layer having 10 units and the interval $d = 15$. In this experiment, we keep L fixed and vary T in increments of ten. As illustrated in Table 6.3, for sequences of length 60-200, SMRNN is able to learn to classify the testing sequences with high accuracies.

Table 6.3: Information latching in SMRNN with interval $d = 15$.

L	T	training set size	testing set size	accuracy
50	60	50	50	96%
50	70	80	80	92.5%
50	80	100	100	97%
50	90	150	150	100%
50	100	150	150	98%
50	110	300	300	96.3%
50	120	400	400	99%
50	130	500	500	99.6%
50	140	500	500	100%
50	150	600	600	100%
50	160	800	800	99.75%
50	170	1000	1000	99.9%
50	180	1200	1200	100%
50	190	1300	1300	99.9%
50	200	1400	1400	100%

Being a comparison test, Elman’s network [40] is also trained with the RTRL algorithm for the same task. The results are given in Table 6.4. From this table, we observe that

Table 6.4: Information latching in Elman’s network.

hidden nodes	L	T	train set size	test set size	accuracy
10	50	55	30	30	30%
10	50	55	100	100	72.5%
10	50	55	200	200	92.5%
10	50	60	100	100	40%
10	50	60	200	200	85%
10	50	65	200	200	75%
10	50	65	300	300	60%
10	50	65	400	400	47.5%
15	50	65	200	200	97.5%
15	50	70	200	200	52.5%
20	50	70	200	200	46.5%
20	50	70	400	400	51.8%
20	50	70	600	600	50.2%

Elman’s network has difficulties in learning to classify sequences of length ≥ 70 . Further experiments using larger sized Elman’s network and larger training set show no improvement in the performance. The inferior performance of Elman’s network does not result from small architecture size or insufficiency of training samples. Instead, Elman’s network has low accuracies due to its capability limitations. A comparison between Table 6.3 and Table 6.4 indicates that SMRNN is able to capture much longer ranges of dependencies than Elman’s network.

6.3.3 SMRNN vs. NARX on the Information Latching Problem

It has been recently recognized that embedded memory in recurrent neural network architectures helps learning long-term temporal dependencies. One of such recurrent architectures with embedded memory is Nonlinear AutoRegressive with eXogeneous (NARX) network [81, 80]. Table 6.5 shows the performance of an NARX network with embedded

memory of order 2 on the information latching problem. In the cases of $L=10$, the NARX

Table 6.5: Information latching in NARX

L	T	training set size	testing set size	accuracy
10	15	50	50	60%
10	15	60	60	85%
10	15	70	70	97.1%
10	20	50	50	82%
10	20	90	90	98.9%
10	20	100	100	100%
10	25	70	70	55.7%
10	25	80	80	77.5%
10	25	90	90	95.6%
50	55	20	20	40%
50	55	40	40	52.5%
50	55	50	50	66%
50	55	60	60	85%
50	60	40	40	45%
50	60	50	50	58%
50	60	80	80	77.5%

fails when the sequences are longer than 25; in the cases of $L=50$, the NARX fails when the sequences are longer than 60. NARX with embedded memory of order 2 is inferior to the SMRNN in their abilities to deal with long time lags.

6.3.4 SMRNN vs. LSTM on the “2-Sequence Problem”

Hochreiter & Schmidhuber have proposed the Long Short-Term Memory algorithm, which is able to bridge long time lags [62]. In order to compare SMRNN with LSTM, we carry out experiments on the “2-sequence problem” which has been used to evaluate the LSTM algorithm. The “2-sequence problem” is to observe and then classify input sequences into two classes, each occurring with probability 0.5. Given a constant T , the sequence length is randomly selected between T and $T+T/10$. Only the first N real-valued sequence elements

CHAPTER 6. SEGMENTED-MEMORY RECURRENT NEURAL NETWORK FOR LONG-TERM DEPENDENCIES

convey relevant information about the class. Sequence elements at position $t > N$ are generated by a Gaussian with mean zero and standard deviation 0.2. Case $N=1$: the first sequence element is 1.0 for class 1 and -1.0 for class 2. Case $N=3$: the first three elements are 1.0 for class 1 and -1.0 for class 2. The target output at the sequence end is 1.0 for class 1 and 0.0 for class 2. Correct classification is defined as “absolute output error at sequence end below 0.2”. We stop training according to the following criteria: none of 256 sequences from a randomly chosen test set is misclassified.

Table 6.6: Results of LSTM on the “2-sequence problem”([62]). N is the number of information-conveying elements at sequence begin. T is minimal sequence length. The column headed by stop gives the number of sequence presentations required to achieve stopping criterion. The right-most column lists the fraction of misclassified post-training sequences from a test set consisting of 2560 sequences.

N	T	stop	# weights	fraction misclassified
1	100	64330	102	0.000117
3	100	39850	102	0.000195

As shown in Table 6.6, the LSTM achieved quite low “fraction misclassified” of 0.000117 for $(N=1, T=100)$ and 0.000195 for $(N=3, T=100)$ respectively. The results (Table 6.7) achieved by SMRNN are comparable with the LSTM.

6.3.5 Impact of the Interval Length

In the process of human memorization of long sequences in segments, the length of the segments plays a critical role. A short interval results in a large number of segments, which leads to difficulties in memorizing all the segment entities and cascading them into the final sequence. A long interval makes the task of memorizing the segment itself difficult. However, there is such an interval that may optimize the memorization process. Similarly,

CHAPTER 6. SEGMENTED-MEMORY RECURRENT NEURAL NETWORK FOR LONG-TERM DEPENDENCIES

Table 6.7: Results of SMRNN on the “2-sequence problem”. stop=the number of epoches needed to converge. HN=number of hidden nodes. TrainNum=number of training samples. fraction misclassified=the fraction of misclassified post-training sequences from a test set consisting of 2560 sequences.

N	T	stop	HN	TrainNum	interval d	fraction misclassified
1	30	3926	5	100	15	0.019531
1	40	5504	5	200	20	0.021484
1	50	7510	10	300	25	0.003516
1	60	5194	10	400	30	0.028906
1	70	2808	15	900	35	0.010547
1	80	3064	15	1000	40	0.012891
1	90	4468	15	1200	45	0.011328
1	100	6433	15	1500	50	0.016016
3	100	5187	15	1500	50	0.007031

the interval length has effect on the performance of SMRNN. When the interval becomes shorter, training takes more time because it requires more computation in each iteration and needs more iterations to converge. On the other hand, if the interval is too long, the SMRNN approximates to a conventional RNN without segmented-memory and hence cannot memorize the long segment itself at a time.

We make the impact of the interval length on SMRNN more explicit by a comparison among SMRNNs with different intervals (Table 6.8). When the interval is too short (≤ 5) or too long (≥ 80), SMRNN fails to capture the long-range dependencies in sequences of length $T=100$.

With the interval length larger than the length of input sequences, SMRNN is actually a traditional multi-layer RNN without segmented-memory. We carry out experiments on an SMRNN with interval $d = 500$ and obtain results given in Table 6.9. As observed from Table 6.3 and Table 6.9, SMRNN (which is a multi-layer RNN with segmented-memory)

CHAPTER 6. SEGMENTED-MEMORY RECURRENT NEURAL NETWORK FOR LONG-TERM DEPENDENCIES

Table 6.8: The results of several SMRNNs with different intervals on the information latching problem.

	interval d	accuracy	epoches	training error
	5	45%	15017	> 1.0
L=50	10	99.5%	1475	< 0.1
T=100	15	98%	2106	< 0.1
TrainNum=200	30	99.5%	2022	< 0.1
TestNum=200	50	99.5%	849	< 0.1
	80	47%	13657	> 0.5

is far more powerful than multi-layer RNN without segmented-memory. Thus, we suggest that the reason why SMRNN outperforms Elman’s network is segmented-memory instead of multiple hidden layers.

Table 6.9: Information latching in SMRNN with interval $d = 500$

L	T	training set size	testing set size	accuracy
50	55	200	200	97.5%
50	60	300	300	98.3%
50	65	300	300	99.7%
50	70	400	400	53.75%

With the interval $d = 1$, the SMRNN is an RNN with multiple feedback connections but without segmented-memory. Finally, we examine the performance of SMRNN with interval $d = 1$ on the information latching task. From the results illustrated in Table 6.10, we observe that SMRNN with interval $d = 1$ performs worse than SMRNN with interval $d = 15$. The results substantiate that segmented-memory rather than multiple feedback connections helps to deal with long time lags.

Table 6.10: Information latching in SMRNN with interval $d = 1$

L	T	training set size	testing set size	accuracy
50	60	100	100	42.5%
50	70	200	200	45%
50	80	200	200	47.5%
50	90	300	300	50%
50	100	300	300	46.7%
50	110	300	300	48.3%
50	120	300	300	51.7%

6.4 Discussion

Segmented-memory recurrent neural network is a novel architecture. In the previous sections, we perform detailed evaluation and present experimental results showing that SMRNN outperforms conventional RNNs without segmented-memory on long-term dependency problems. Several different approaches such as NARX and LSTM have been proposed by other researchers to solve the long-term dependency problem in recurrent networks. We further compare our SMRNN with NARX and LSTM. The comparison results indicate that SMRNN is superior to NARX and comparable to LSTM. In addition to the encouraging experimental results, we provide theoretical evidence showing that SMRNN is more efficient in learning long-term dependencies than conventional RNNs. Furthermore, we carry out investigation into the effect of the length of interval on learning speed and convergence performance.

By inserting intervals into the memory of contextual information, segmented-memory recurrent neural networks can learn to bridge long time lags even in noisy, highly unpredictable environments, without loss of short time lag capabilities. As shown by experiments

CHAPTER 6. SEGMENTED-MEMORY RECURRENT NEURAL NETWORK FOR LONG-TERM DEPENDENCIES

on information latching, SMRNN converges fast and generalizes well for short sequences as well. Another advantage of SMRNN is its flexibility. The interval length can be either fixed or varying. The length of interval d is allowed to be different among different sequences and even among different parts of one sequence. We can freely adjust the length of interval without any modification on the network architecture, without changing the number of neurons and the connections among them.

SMRNN is actually an Elman-style network with segmented-memory. Segmented-memory can be embedded into other recurrent architectures to improve their long time lag capabilities. For example, NARX recurrent neural network has been proposed to deal with long-term dependencies. An NARX with segmented-memory is supposed to be more capable of representing long-term dependencies.

There is a trade-off between the efficient training of gradient descent and long-range information latching. Other existing solutions for long-term dependency problems either perform just as poorly (gradient-based training), or require far more computation (non-gradient training). For SMRNN, the training algorithm is essentially gradient descent algorithm, hence SMRNN does not circumvent the problem of vanishing gradients, but just alleviates the problem. Actually, SMRNN also faces an increasingly difficult problem as the duration of dependencies to be captured increases.

Segmented-memory recurrent neural network is not the first method that involves sequence decomposition. For instance, Schmidhuber's hierarchical chunker system [125] uses the principle of history compression to compress sequences. The chunker tries to detect causal dependencies in the temporal input sequences and learns to attend to unexpected

CHAPTER 6. SEGMENTED-MEMORY RECURRENT NEURAL NETWORK FOR LONG-TERM DEPENDENCIES

inputs instead of focussing on every input. By doing so, the system is able to reduce the descriptions of input sequences without the loss of information. However, the sequence chunker relies on predictable local regularities in the input sequences. The performance of chunker systems deteriorates as the noise level increases and the input sequences become less compressible. In some problems like the information latching problem and the “2-sequence problem”, the input sequences do not have local regularities. For such problems, the chunker fails to capture the long time-lagged dependencies.

Chapter 7

Bidirectional Segmented-Memory Recurrent Neural Network for Protein Secondary Structure Prediction

Pollastri's bidirectional recurrent neural networks (PBRNN) have the advantage that it considers the information on the whole sequence instead of local neighborhood information of a residue to predict its secondary structure type. Their algorithmic idea begins to address the problem of long-term dependencies and encourages further investigation. However, learning long-term dependencies with gradient descent is difficult. The generalized back propagation algorithm for training Pollastri's BRNN is essentially a gradient descent algorithm. In PBRNN, error propagation in both the forward and backward chains is subject to exponential decay; so it has limitations for learning the remote information efficiently [102]. In the practice of PSS prediction, PBRNN can reliably utilize input information located within about ± 15 amino acids around the residue of interest. PBRNN fails to discover relevant information contained even in more distant portions of the protein sequence. However, interacting amino acids can be separated by much more than 15

CHAPTER 7. BIDIRECTIONAL SEGMENTED-MEMORY RECURRENT NEURAL NETWORK FOR PROTEIN SECONDARY STRUCTURE PREDICTION

residues in between. As discussed in the previous chapter, segmented-memory recurrent neural network (SMRNN) outperforms conventional RNNs on long-term dependency problems. Our experimental results show that the performance of SMRNN is better than that of Elman-style networks and NARX networks. This advantage enables SMRNN to learn the long-term dependencies present in protein sequences for secondary structure prediction. However, SMRNN is causal. A dynamic system is causal if the output only depends on the current and previous input values and not on any future input values. Obviously, the problem of PSS prediction is non-causal because the pattern of secondary structure associated with an amino acid in a sequence may strongly depend on both upstream and downstream information. To remove the causality assumption, we develop a *bidirectional (noncausal) segmented-memory recurrent neural network* (BSMRNN) to capture longer ranges of bidirectional dependencies in protein chains for secondary structure prediction.

7.1 Bidirectional Segmented-Memory Recurrent Neural Network

7.1.1 Architecture

The architecture of bidirectional segmented-memory recurrent neural network is illustrated in Figure 7.1. The left sub-network in white is the forward SMRNN N_ϕ ; the right sub-network in gray is the backward SMRNN N_ψ ; the top sub-network in black is a feed-forward MLP N_ζ . The input vector I_t encodes the amino acid at position t . The vector F_t stores contextual information contained in the upstream portion to the left of position t ; the vector B_t stores contextual information contained in the downstream portion to the right

CHAPTER 7. BIDIRECTIONAL SEGMENTED-MEMORY RECURRENT NEURAL NETWORK FOR PROTEIN SECONDARY STRUCTURE PREDICTION

of position t . The states of F_t and B_t are updated using the following equations:

$$FS_t = \begin{cases} \phi_1(FS_0, I_t) & \text{if } t = SH \\ \phi_2(FS_{t-1}, I_t) & \text{otherwise} \end{cases} \quad (7.1)$$

$$F_t = \begin{cases} \phi_3(F_{t-1}, FS_t) & \text{if } t = ST \\ F_{t-1} & \text{otherwise} \end{cases} \quad (7.2)$$

$$BS_t = \begin{cases} \psi_1(BS_{L+1}, I_t) & \text{if } t = ST \\ \psi_2(BS_{t+1}, I_t) & \text{otherwise} \end{cases} \quad (7.3)$$

$$B_t = \begin{cases} \psi_3(B_{t+1}, BS_t) & \text{if } t = SH \\ B_{t+1} & \text{otherwise} \end{cases} \quad (7.4)$$

where SH (Segment Head) refers to the beginning of a segment, ST (Segment Tail) denotes the end of a segment, $\phi_i (i = 1, 2, 3)$ and $\psi_i (i = 1, 2, 3)$ are nonlinear functions implemented by forward SMRNN N_ϕ and backward SMRNN N_ψ respectively (left and right subnetworks in Figure 7.1). The vector O_t , i.e. the output at residue t , is a vector of posterior probabilities of secondary structure classes for sequence position t given the protein's sequence of amino acids. In this model, the output O_t is computed as:

$$O_t = \zeta(F_t, B_t, I_t) \quad (7.5)$$

where $\zeta()$ is realized by MLP N_ζ (top subnetwork in Figure 7.1).

7.1.2 Inference and Learning

Given an amino acid sequence $X_1, X_2, \dots, X_t, \dots, X_L$, the BSMRNN can estimate the posterior probabilities of secondary structure classes for each sequence position t . Starting from $FS_0 = 0$ and $F_0 = 0$, the forward SMRNN reads in the substring $X_1, \dots, X_{t-1}, X_t$ from left to right and updates its symbol-level state $FS_1, FS_2, \dots, FS_t$ and segment-level state $F_1, F_2, \dots, F_t$. Similarly, starting from $BS_{L+1} = 0$ and $B_{L+1} = 0$, the backward SMRNN scans $X_t, X_{t+1}, \dots, X_L$ from right to left and updates its symbol-level state

CHAPTER 7. BIDIRECTIONAL SEGMENTED-MEMORY RECURRENT NEURAL NETWORK FOR PROTEIN SECONDARY STRUCTURE PREDICTION

Figure 7.1: Bidirectional segmented-memory recurrent neural network.

$BS_L, BS_{L-1}, \dots, BS_t$ and segment-level state $B_L, B_{L-1}, \dots, B_t$. After the forward and backward propagations have taken place, the output at position t is then obtained by combining the two hidden representations of context and the current input.

The learning algorithm is based on gradient descent. The error signal is propagated over time in both directions. The weights in subnetwork N_ζ are updated in the same way as standard MLP. The derivatives of the error function with respect to states F_t and B_t are calculated and injected into N_ϕ and N_ψ respectively. Then the derivatives for weights in N_ϕ and N_ψ are calculated using the same formulas as those of causal SMRNN (refer to the previous chapter).

7.2 Experiments and Discussion

All the BSMRNNs examined in our experiments has 20 input units, 3 output units and 20 units in the hidden layer H of the top subnetwork. We use equal-sized forward SMRNN

CHAPTER 7. BIDIRECTIONAL SEGMENTED-MEMORY RECURRENT NEURAL NETWORK FOR PROTEIN SECONDARY STRUCTURE PREDICTION

and backward SMRNN, which have 10 units in the segment-level state vectors and 10 units in the symbol-level state vectors. We trained several different BSMRNNs by varying the length of interval. Training is stopped using a fixed threshold on the reduction in error or when the maximum (1000) number of training epoches is reached.

The results on the PSIPRED data sets are given in Table 7.1 where d denotes the length of intervals. The interval length slightly affects the performance of the BSMRNN, resulting in small differences among the results obtained by different BSMRNNs with different interval length. The best results on each testing data sets are summarized in Table 7.2. As shown in this table, when interval $d = 15$, BSMRNN achieves its highest Q3 on Test set 1, Test set 3 and Combined test set; for Test set 2, the Q3 is highest when BSMRNN's interval $d = 5$. Our BSMRNN with interval length $d = 15$ achieves a Q3 accuracy of 74.10% and an SOV value of 63.4083 on the Combined test set.

We compare our bidirectional segmented-memory RNN with Pollastri's BRNN (PBRNN) and the original BRNN (Table 7.3). Our BSMRNN greatly improves accuracies over Pollastri's BRNN, and slightly outperforms the original BRNN.

- BSMRNN vs. BRNN:
 - BSMRNN achieves higher Q3 on Test set 1, Test set 2, Test set 3 and Combined test set;
 - BSMRNN achieves higher SOV on Test set 1, Test set 2 and Combined test set.
 - BSMRNN achieves higher C_C, C_H and C_E on Combined test set.
- BSMRNN vs. PBRNN:

CHAPTER 7. BIDIRECTIONAL SEGMENTED-MEMORY RECURRENT NEURAL NETWORK FOR PROTEIN SECONDARY STRUCTURE PREDICTION

Table 7.1: Experimental results of BSMRNN on the PSIPRED data sets. The BSMRNN size is: Inum=20, Hnum=20, Onum=3, Fnum (Bnum) =10, FSnum (BSnum) =10.

measure	Testing set	d=5	d=10	d=15	d=20	d=50
Q3(%)	Test set 1	73.21	73.27	74.06	73.98	73.87
	Test set 2	74.35	73.35	73.49	72.76	72.82
	Test set 3	73.78	74.34	74.73	74.69	74.40
	Combined test set	73.75	73.63	74.10	73.83	73.71
Q_C(%)	Test set 1	82.51	80.28	83.34	81.07	81.56
	Test set 2	80.15	83.27	88.28	75.55	86.78
	Test set 3	85.53	85.15	81.25	84.94	86.63
	Combined test set	82.70	82.71	84.23	80.54	84.75
Q_H(%)	Test set 1	73.29	75.36	75.06	79.33	75.83
	Test set 2	74.74	72.86	67.89	73.52	73.65
	Test set 3	72.21	74.09	77.10	74.13	70.34
	Combined test set	73.34	74.21	73.70	75.89	73.34
Q_E(%)	Test set 1	54.78	56.21	54.22	51.72	55.65
	Test set 2	63.99	57.04	55.34	67.05	47.99
	Test set 3	53.60	53.73	57.91	55.68	57.58
	Combined test set	57.59	55.76	55.70	58.15	53.59
SOV	Test set 1	61.1984	61.5338	63.1209	62.5240	61.8621
	Test set 2	64.8453	64.1466	63.6414	63.7022	61.2812
	Test set 3	64.9221	65.0561	63.5198	65.5474	63.8070
	Combined test set	63.4825	63.4383	63.4083	63.8343	62.2905
SOV_C	Test set 1	60.8058	59.2913	61.2115	59.8743	60.0694
	Test set 2	63.1817	64.4725	66.3006	61.3884	64.7633
	Test set 3	65.8418	65.6103	61.7533	65.8217	66.2478
	Combined test set	63.0524	62.7813	62.9076	62.1170	63.3600
SOV_H	Test set 1	65.2671	68.4384	69.9816	72.3675	67.3213
	Test set 2	68.2648	70.0223	64.6789	66.0595	64.5585
	Test set 3	68.6970	69.2871	68.3406	70.3042	63.2198
	Combined test set	67.2417	69.1739	67.8956	69.8159	65.1197
SOV_E	Test set 1	54.8008	54.9930	55.8135	51.9923	56.3921
	Test set 2	63.4012	56.2687	57.5977	64.8694	50.4951
	Test set 3	56.1223	56.1662	58.1880	56.5677	60.0269
	Combined test set	58.1800	55.7850	57.1363	57.7620	55.4232
C_C	Test set 1	0.53497	0.54047	0.54588	0.54888	0.54921
	Test set 2	0.54661	0.53524	0.54517	0.53106	0.53437
	Test set 3	0.55215	0.55870	0.55624	0.55575	0.55926
	Combined test set	0.54337	0.54404	0.54821	0.54413	0.54675
C_H	Test set 1	0.67276	0.67078	0.68872	0.67938	0.67923
	Test set 2	0.70108	0.68242	0.69494	0.66625	0.67964
	Test set 3	0.64678	0.65477	0.66774	0.67255	0.66079
	Combined test set	0.67317	0.66956	0.68344	0.67314	0.67313
C_E	Test set 1	0.52413	0.51947	0.53621	0.53206	0.53016
	Test set 2	0.56806	0.54950	0.55039	0.54919	0.53328
	Test set 3	0.57997	0.59049	0.58915	0.59279	0.59747
	Combined test set	0.55559	0.54984	0.55736	0.55454	0.54984

CHAPTER 7. BIDIRECTIONAL SEGMENTED-MEMORY RECURRENT NEURAL NETWORK FOR PROTEIN SECONDARY STRUCTURE PREDICTION

Table 7.2: Performance of BSMRNNs with optimal interval lengths.

Testing set	Test set 1	Test set 2	Test set 3	Combined test set
FS (BS)	10	10	10	10
F (B)	10	10	10	10
interval	d=15	d=5	d=15	d=15
Q3(%)	74.06	74.35	74.73	74.10
Q_C(%)	83.34	80.15	81.25	84.23
Q_H(%)	75.06	74.74	77.10	73.70
Q_E(%)	54.22	63.99	57.92	55.70
SOV	63.1209	64.8453	63.5198	63.4083
SOV_C	61.2115	63.1817	61.7533	62.9076
SOV_H	69.9816	68.2648	68.3406	67.8956
SOV_E	55.8135	63.4012	58.1880	57.1363
C_C	0.54588	0.54661	0.55624	0.54821
C_H	0.68872	0.70108	0.66774	0.68344
C_E	0.53621	0.56806	0.58915	0.55736

- Q3: BSMRNN achieves about 2-3% higher Q3 than PBRNN on each of the testing subsets (Test set 1, Test set 2, Test set 3) and on the entire testing set (Combined test set).
- SOV, SOV_C, SOV_H, SOV_E: On the Combined test set, BSMRNN improves SOV by 5% over PBRNN. On each set, BSMRNN’s segment overlap values including SOV, SOV_C, SOV_H and SOV_E are higher than those of PBRNN.
- C_C, C_H, C_E: BSMRNN achieves higher correlation coefficients than PBRNN on each set of Test set 1, Test set 2, Test set 3 and Combined test set.

CHAPTER 7. BIDIRECTIONAL SEGMENTED-MEMORY RECURRENT NEURAL NETWORK FOR PROTEIN SECONDARY STRUCTURE PREDICTION

Table 7.3: Comparison among PBRNN, BRNN and BSMRNN.

measure	Testing set	PBRNN	BRNN	BSMRNN
Q3(%)	Test set 1	71.95	73.61	74.06
	Test set 2	71.30	72.94	73.49
	Test set 3	73.41	74.47	74.73
	Combined test set	72.21	73.67	74.10
Q_C(%)	Test set 1	80.69	82.45	83.34
	Test set 2	79.74	80.17	88.28
	Test set 3	78.49	80.24	81.25
	Combined test set	79.72	81.06	84.23
Q_H(%)	Test set 1	77.16	74.31	75.06
	Test set 2	71.43	73.48	67.89
	Test set 3	77.09	75.08	77.10
	Combined test set	75.49	74.33	73.70
Q_E(%)	Test set 1	46.62	55.09	54.22
	Test set 2	56.74	59.94	55.34
	Test set 3	57.14	62.17	57.92
	Combined test set	53.21	58.85	55.70
SOV	Test set 1	59.8224	62.2104	63.1209
	Test set 2	60.3440	62.4237	63.6414
	Test set 3	62.5845	64.3487	63.5198
	Combined test set	60.8649	62.9593	63.4083
SOV_C	Test set 1	58.7954	61.5206	61.2115
	Test set 2	61.1245	63.4552	66.3006
	Test set 3	60.1526	63.7589	61.7533
	Combined test set	59.9516	62.8277	62.9076
SOV_H	Test set 1	69.6493	67.7551	69.9816
	Test set 2	62.6777	62.4404	64.6789
	Test set 3	70.0166	66.3065	68.3406
	Combined test set	67.7118	65.7217	67.8956
SOV_E	Test set 1	46.0937	54.3339	55.8135
	Test set 2	55.7286	60.4976	57.5977
	Test set 3	54.6762	61.8782	58.1880
	Combined test set	51.9527	58.6804	57.1363
C_C	Test set 1	0.52384	0.54317	0.54588
	Test set 2	0.51965	0.52916	0.54517
	Test set 3	0.53916	0.55937	0.55624
	Combined test set	0.52753	0.54407	0.54821
C_H	Test set 1	0.63744	0.67606	0.68872
	Test set 2	0.63038	0.66561	0.69494
	Test set 3	0.64808	0.65197	0.66774
	Combined test set	0.63950	0.66564	0.68344
C_E	Test set 1	0.50012	0.52975	0.53621
	Test set 2	0.51407	0.55182	0.55039
	Test set 3	0.55812	0.59369	0.58915
	Combined test set	0.52283	0.55726	0.55736

Chapter 8

Cascaded Bidirectional Recurrent Neural Networks for Protein Secondary Structure Prediction

The formation of secondary structure (SS) elements is only to a certain degree due to sequential local interactions between amino acids [93, 140]. In chapter 4, we have addressed the negative effect of long-range interactions on protein secondary structure prediction accuracy of window-based approaches. In order to improve protein secondary structure prediction, one should use local information as well as distant information that is not contained in local input windows. Pollastri's BRNN, our BSMRNN and BLSTM have tried to overcome the drawbacks of local fixed-sized window approaches. However, they are all trained on single positions of the sequence to classify mutually independent secondary structures. A comprehensive study on a large data set of non-homologous proteins suggests that secondary structures are strongly and non-locally correlated along the chain [29]. Hence, Pollastri's BRNN, our BSMRNN and BLSTM do not consider the correlations among consecutive secondary structure patterns. The famous PHD method has successfully used a structure-to-structure network to correlate predictions on adjacent amino acids

[114]. However, by using local input windows, PHD is inefficient in learning long-range dependencies between amino acid entities and those between secondary structure elements along the chain. Therefore, we propose a cascaded BRNN to take into account both SS-SS correlations and long-range interactions for PSS prediction.

8.1 Protein Sequence-Structure Correlations

The three-dimensional structure of a protein is determined by its amino acid sequence via the process of protein folding. Correlations between protein structures and amino acid sequences have been extensively studied and widely used for protein structure prediction. Specially, secondary structure predictors generally use correlations between a primary structure sequence and the corresponding secondary structure sequence. In this section, we investigate the correlations between amino acid (AA) sequences and secondary structure (SS) sequences on the EVA set (section 4.1.1). We quantify the protein AA-AA, AA-SS and SS-SS correlations using mutual information, an information-theoretic measure that quantifies how much knowledge one quantity reveals about another. Our study reveals strong and long-ranged correlations between secondary structural patterns.

8.1.1 Primary Structure is Random

Figure 8.1.1 shows the correlations between residues separated by varying distances along the chain. The AA-AA mutual information $I(R_i; R_{i+j})$ is calculated by:

$$I(R_i, R_{i+j}) = \sum_{R_i, R_{i+j} \in \Sigma_R} p(R_i, R_{i+j}) \log \frac{p(R_i, R_{i+j})}{p(R_i)p(R_{i+j})} \quad (8.1)$$

CHAPTER 8. CASCADED BIDIRECTIONAL RECURRENT NEURAL NETWORKS FOR PROTEIN SECONDARY STRUCTURE PREDICTION

where Σ_R denotes the set of 20 amino acids. The nearest neighbor mutual information, $I(R_i; R_{i+1}) = 0.005849$, is small. Neighboring amino acids are approximately independent, and the correlations between more distantly separated residues are also very weak. Amino acids are almost entirely uncorrelated along the chain. This implies that the primary structure of proteins is essentially random at the level of linear amino acid sequences.

Figure 8.1: AA-AA mutual information $I(R_i; R_{i+j})$ with respect to j .

In spite of the fact that protein sequences have been selected by evolution to fold into compact and functional shapes, little regularities in the sequences can be detected by the means of information theory. Our results support the notion originally expressed by Monod that protein primary sequences are not structured by any rules [90]. Weiss and his colleague's studies on the information content of protein sequences have confirmed the idea that amino

acid sequences can be regarded as slightly edited random strings [143]. However, it does not require non-randomness for proteins to be functional. Ptitsyn & Volkenstein [107] suggested that proteins have the same amino acids in some fixed positions that are also active.

8.1.2 Primary-Secondary Structure Correlation is Weak

With the objective of better understanding secondary structure formation, we are interested in measuring the information carried by an amino acid of type R at position $i+j$ concerning the secondary structure conformation S at position i .

$$I(S_i, R_{i+j}) = \sum_{S_i \in \{C,H,E\}} \sum_{R_{i+j} \in \Sigma_R} p(S_i, R_{i+j}) \log \frac{p(S_i, R_{i+j})}{p(S_i)p(R_{i+j})} \quad (8.2)$$

The AA-SS mutual information $I(S_i; R_{i+j})$ with respect to j is illustrated in Figure 8.2. The primary-to-secondary structure interactions are subtle ($I(S_i; R_{i+j}) < 0.08$), making protein secondary structure determination from the amino acid sequence a difficult task.

8.1.3 Strong Correlations in Secondary Structure

In this subsection, we investigate the protein SS-SS correlations on the EVA set. As formulated below, $I(S_i, S_{i+j})$ is defined to be the mutual information between the secondary structure types of two residues that are separated by j residues along the primary sequence.

$$I(S_i, S_{i+j}) = \sum_{S_i, S_{i+j} \in \{C,H,E\}} p(S_i, S_{i+j}) \log \frac{p(S_i, S_{i+j})}{p(S_i)p(S_{i+j})} \quad (8.3)$$

Figure 8.3 demonstrates the SS-SS mutual information $I(S_i; S_{i+j})$ with respect to j . Neighboring secondary structure elements are strongly correlated along the chain, resulting in a relatively large nearest neighbor mutual information value, $I(S_i; S_{i+1}) = 0.789805$ bits.

CHAPTER 8. CASCADED BIDIRECTIONAL RECURRENT NEURAL NETWORKS FOR PROTEIN SECONDARY STRUCTURE PREDICTION

Figure 8.2: AA-SS mutual information for residues at various separations along the amino acid chain.

Secondary structures are also strongly correlated over extended distances because of the long helices and strands interspersed with long regions. Although the amount of information $I(S_i; S_{i+j})$ is modest when j is larger than 8, we observe it to be important for secondary structure prediction. PHD has successfully used a structure-to-structure network to correlate predictions on adjacent amino acids. The strong correlations between secondary structures explain why PHD is much more efficient than other neural network-based predictors. However, by using a local input window, PHD's structure-to-structure network is unable to capture the long-range dependencies between SS elements along the secondary structure sequence.

CHAPTER 8. CASCADED BIDIRECTIONAL RECURRENT NEURAL NETWORKS FOR PROTEIN SECONDARY STRUCTURE PREDICTION

Figure 8.3: SS-SS mutual information for residues at various separations along the amino acid chain.

8.2 Cascaded Bidirectional Recurrent Neural Network

Based on the analysis in the previous section, we propose a cascaded bidirectional recurrent neural network to capture long-range interactions and strong SS-SS correlations for PSS prediction.

8.2.1 Architecture and Dynamics

A cascaded bidirectional recurrent neural network (BRNN) is essentially two BRNNs in cascade (Figure 8.4). The first BRNN (sequence-to-structure BRNN) predicts the secondary structures based on amino acid sequences. The second BRNN (structure-to-structure BRNN) refines the raw predictions from the first BRNN. The purpose of the second BRNN

CHAPTER 8. CASCADED BIDIRECTIONAL RECURRENT NEURAL NETWORKS FOR PROTEIN SECONDARY STRUCTURE PREDICTION

is to capture the contextual information of the neighboring secondary structures in determining the secondary structure of a particular residue.

Figure 8.4: Cascaded BRNN.

Letting $t(1 \leq t \leq L)$ denote position within an amino acid sequence of length L and vector X_t encode the residue at position t , the overall model outputs for each t a probability vector Z_t representing the posterior probabilities of the residue at position t in each one of the three secondary structure classes (C,H,E).

$$Z_t = \mathfrak{N}_Z(FY_t, Y_t, BY_t) \quad (8.4)$$

The vectors FY_t and BY_t encode the contextual information contained in the upstream

CHAPTER 8. CASCADED BIDIRECTIONAL RECURRENT NEURAL NETWORKS FOR PROTEIN SECONDARY STRUCTURE PREDICTION

(i.e. $Y_1, Y_2, \dots, Y_t$) and downstream (i.e. $Y_t, Y_{t+1}, \dots, Y_L$) portions of the output sequence ($Y_1, Y_2, \dots, Y_L$) from the lower BRNN. They are defined by the following recurrent bidirectional equations:

$$FY_t = \aleph_{FY}(FY_{t-1}, Y_t) \quad (8.5)$$

$$BY_t = \aleph_{BY}(BY_{t+1}, Y_t) \quad (8.6)$$

Vector Y_t is the output generated by the sequence-to-structure BRNN.

$$Y_t = \aleph_Y(FX_t, X_t, BX_t) \quad (8.7)$$

Similarly, the bidirectional dynamics of the first level BRNN are formulated as below:

$$FX_t = \aleph_{FX}(FX_{t-1}, X_t) \quad (8.8)$$

$$BX_t = \aleph_{BX}(BX_{t+1}, X_t) \quad (8.9)$$

The forward chain FX_t stores contextual information contained at the left of position t (i.e. $X_1, X_2, \dots, X_t$). The backward chain BX_t is in charge of storing contextual information contained at the right of position t (i.e. $X_t, X_{t+1}, \dots, X_L$).

The functions $\aleph_Y(\cdot)$ and $\aleph_Z(\cdot)$ are realized by feed-forward sub-networks N_Y and N_Z . $\aleph_{FX}(\cdot)$, $\aleph_{BX}(\cdot)$, $\aleph_{FY}(\cdot)$ and $\aleph_{BY}(\cdot)$ are learnable non-linear state transition functions, implemented by recurrent sub-networks N_{FX} , N_{BX} , N_{FY} and N_{BY} respectively. The boundary conditions of FX , BX , FY and BY are set to 0, i.e. $FX_0 = 0$, $BX_{L+1} = 0$, $FY_0 = 0$ and $BY_{L+1} = 0$ where L is the length of the sequence being examined.

8.2.2 Inference Algorithm

The inference algorithm in cascaded BRNN is straightforward. Figure 8.5 gives a simplified view of the updates for variables involved in cascaded BRNN. Starting from $FX_0 = 0$, all the states $FX_t(t = 1, 2, \dots, L)$ are updated from left to right, following equation 8.8. Similarly, states $BX_t(t = 1, 2, \dots, L)$ are updated from right to left using the boundary condition $BX_{L+1} = 0$ and following equation 8.9. After the forward and backward propagations have taken place in the first BRNN, states $Y_t(t = 1, 2, \dots, L)$ can be computed using equation 8.7. The outputs $Y_t(t = 1, 2, \dots, L)$ from the first level BRNN are fed to the second level BRNN as inputs. States $FY_t(t = 1, 2, \dots, L)$ and $BY_t(t = 1, 2, \dots, L)$ are updated according to equation 8.5 and 8.6 with FY_0 and BY_{L+1} initialized to be 0. Finally, the prediction $Z_t(t = 1, 2, \dots, L)$ is obtained by combining $FY_t(t = 1, 2, \dots, L)$, $Y_t(t = 1, 2, \dots, L)$ and $BY_t(t = 1, 2, \dots, L)$ using equation 8.4. For both sequence-to-structure BRNN and structure-to-structure BRNN, the forward and backward propagations need to be computed from end to end only once per protein sequence.

8.2.3 Learning Algorithm

The cascaded BRNN is trained in the way that minimizes the mean squared error. The error at position t is calculated by:

$$E_t = \sum_{i=1}^{Znum} \frac{1}{2} \times (Z_t^i - S_t^i)^2 \quad (8.10)$$

where S_t^i is the i -th target output and Z_t^i is the i -th actual output at position t . Optimization is based on gradient descent, where gradients are computed by a noncausal version of real time recurrent learning algorithm. The derivative of error E_t with respect to parameter

CHAPTER 8. CASCADED BIDIRECTIONAL RECURRENT NEURAL NETWORKS FOR PROTEIN SECONDARY STRUCTURE PREDICTION

Figure 8.5: Direct dependencies among the variables involved in a cascaded BRNN.

W is calculated for each position t . The total gradients are obtained by summing all the contributions associated to different positions in the sequence:

$$\frac{\partial E}{\partial W} = \sum_{t=1}^L \frac{\partial E_t}{\partial W} \quad (8.11)$$

The weight W is updated by:

$$W = W - \alpha * \frac{\partial E}{\partial W} \quad (8.12)$$

with a constant learning rate α .

The gradients of W_{Z_FY} , W_{Z_X} and W_{Z_BY} can be easily calculated. We derive the learning algorithm for the rest of the weights. For simplicity, we assume that each layer

CHAPTER 8. CASCADED BIDIRECTIONAL RECURRENT NEURAL NETWORKS FOR PROTEIN SECONDARY STRUCTURE PREDICTION

contains only one unit. The synaptic inputs to FY , Y and BY are:

$$FY_t^{net} = W_{FY_FY} * FY_{t-1} + W_{FY_Y} * Y_t \quad (8.13)$$

$$Y_t^{net} = W_{Y_FX} * FX_t + W_{Y_X} * X_t + W_{Y_BX} * BX_t \quad (8.14)$$

$$BY_t^{net} = W_{BY_BY} * BY_{t+1} + W_{BY_Y} * Y_t \quad (8.15)$$

The derivatives of E_t with respect to the weights are calculated by:

$$\frac{\partial E_t}{\partial W_{FY}} = \frac{\partial E_t}{\partial FY_t^{net}} \frac{\partial FY_t^{net}}{\partial W_{FY}} \quad (8.16)$$

$$\frac{\partial E_t}{\partial W_{BY}} = \frac{\partial E_t}{\partial BY_t^{net}} \frac{\partial BY_t^{net}}{\partial W_{BY}} \quad (8.17)$$

$$\frac{\partial E_t}{\partial W_Y} = \frac{\partial E_t}{\partial FY_t^{net}} \frac{\partial FY_t^{net}}{\partial W_Y} + \frac{\partial E_t}{\partial Y_t^{net}} \frac{\partial Y_t^{net}}{\partial W_Y} + \frac{\partial E_t}{\partial BY_t^{net}} \frac{\partial BY_t^{net}}{\partial W_Y} \quad (8.18)$$

where W_{FY} stands for connections leading to layer FY , i.e. W_{FY_FY} and W_{FY_Y} ; W_{BY} denotes W_{BY_BY} and W_{BY_Y} ; W_Y represents all weights below the layer Y , including W_{Y_FX} , W_{Y_X} , W_{Y_BX} , W_{FX_FX} , W_{FX_X} , W_{BX_BX} and W_{BX_X} .

The derivatives of synaptic inputs FY_t^{net} and BY_t^{net} are computed in a recurrent way, starting from the derivatives at position $t = 0$ and $t = L + 1$ initialized to be zero.

$$\frac{\partial FY_t^{net}}{\partial W_{FY_FY}} = FY_{t-1} + \frac{\partial FY_t^{net}}{\partial FY_{t-1}^{net}} \frac{\partial FY_{t-1}^{net}}{\partial W_{FY_FY}} \quad (8.19)$$

$$\frac{\partial FY_t^{net}}{\partial W_{FY_Y}} = Y_t + \frac{\partial FY_t^{net}}{\partial FY_{t-1}^{net}} \frac{\partial FY_{t-1}^{net}}{\partial W_{FY_Y}} \quad (8.20)$$

$$\frac{\partial BY_t^{net}}{\partial W_{BY_BY}} = BY_{t+1} + \frac{\partial BY_t^{net}}{\partial BY_{t+1}^{net}} \frac{\partial BY_{t+1}^{net}}{\partial W_{BY_BY}} \quad (8.21)$$

$$\frac{\partial BY_t^{net}}{\partial W_{BY_Y}} = Y_t + \frac{\partial BY_t^{net}}{\partial BY_{t+1}^{net}} \frac{\partial BY_{t+1}^{net}}{\partial W_{BY_Y}} \quad (8.22)$$

CHAPTER 8. CASCADED BIDIRECTIONAL RECURRENT NEURAL NETWORKS FOR PROTEIN SECONDARY STRUCTURE PREDICTION

$$\frac{\partial FY_t^{net}}{\partial W_Y} = \frac{\partial FY_t^{net}}{\partial Y_t^{net}} \frac{\partial Y_t^{net}}{\partial W_Y} + \frac{\partial FY_t^{net}}{\partial FY_{t-1}^{net}} \frac{\partial FY_{t-1}^{net}}{\partial W_Y} \quad (8.23)$$

$$\frac{\partial BY_t^{net}}{\partial W_Y} = \frac{\partial BY_t^{net}}{\partial Y_t^{net}} \frac{\partial Y_t^{net}}{\partial W_Y} + \frac{\partial BY_t^{net}}{\partial BY_{t+1}^{net}} \frac{\partial BY_{t+1}^{net}}{\partial W_Y} \quad (8.24)$$

The derivatives of the synaptic input Y_t^{net} with respect to W_{FX} and W_{BX} depend on the derivatives of FX_t^{net} and BX_t^{net} :

$$\frac{\partial Y_t^{net}}{\partial W_{FX}} = \frac{\partial Y_t^{net}}{\partial FX_t^{net}} \frac{\partial FX_t^{net}}{\partial W_{FX}} \quad (8.25)$$

$$\frac{\partial Y_t^{net}}{\partial W_{BX}} = \frac{\partial Y_t^{net}}{\partial BX_t^{net}} \frac{\partial BX_t^{net}}{\partial W_{BX}} \quad (8.26)$$

where W_{FX} includes W_{FX_FX} and W_{FX_X} ; W_{BX} includes W_{BX_BX} and W_{BX_X} .

The initial derivatives of FX_0^{net} and BX_{L+1}^{net} are set to be zero. At position t , the derivatives of FX_t^{net} and BX_t^{net} are calculated recurrently as follows:

$$\frac{\partial FX_t^{net}}{\partial W_{FX_FX}} = FX_{t-1} + \frac{\partial FX_t^{net}}{\partial FX_{t-1}^{net}} \frac{\partial FX_{t-1}^{net}}{\partial W_{FX_FX}} \quad (8.27)$$

$$\frac{\partial FX_t^{net}}{\partial W_{FX_X}} = X_t + \frac{\partial FX_t^{net}}{\partial FX_{t-1}^{net}} \frac{\partial FX_{t-1}^{net}}{\partial W_{FX_X}} \quad (8.28)$$

$$\frac{\partial BX_t^{net}}{\partial W_{BX_BX}} = BX_{t+1} + \frac{\partial BX_t^{net}}{\partial BX_{t+1}^{net}} \frac{\partial BX_{t+1}^{net}}{\partial W_{BX_BX}} \quad (8.29)$$

$$\frac{\partial BX_t^{net}}{\partial W_{BX_X}} = X_t + \frac{\partial BX_t^{net}}{\partial BX_{t+1}^{net}} \frac{\partial BX_{t+1}^{net}}{\partial W_{BX_X}} \quad (8.30)$$

8.3 Experiment

8.3.1 Experimental Settings

Training is stopped by a fixed threshold on the reduction in error or when the maximum number (1000) of training epochs is reached. For cascaded BRNN, PBRNN and the original

CHAPTER 8. CASCADED BIDIRECTIONAL RECURRENT NEURAL NETWORKS FOR PROTEIN SECONDARY STRUCTURE PREDICTION

BRNN, we adopted the same learning strategy except for minor adjustments according to their specific architectural details. Specifically, we used the following:

- the weights in the network were updated after each sequence presentation.
- constant learning rate of 0.01.
- no bias weights are added to the neurons.
- all the activation functions are logistic function sigmoid in $[0,1]$.
- every parameter is initialized with small random values in $[-0.1,0.1]$.
- the boundary conditions of all state vectors containing upstream and downstream context are specified to be zero, at both ends of the input sequence.
- no embedded memory is inserted in the forward and backward subnetworks.

8.3.2 Results of Cascaded BRNN

The experimental results of cascaded BRNN on the PSIPRED set are summarized in Table 8.1. In our experiments, different cascaded BRNNs were obtained by varying architectural details such as the number of hidden units. From Table 8.1, we select the cascaded BRNN architecture that yields the highest Q3 accuracy for each Testing set, and provide the values of other performance measures in Table 8.2.

Among the slightly different results of the several cascaded BRNNs on the PSIPRED Combined test set, the highest Q3 is 74.38% and the lowest Q3 is 73.75%; the SOV value varies within the range $[65.446, 66.464]$. As indicated in Table 8.2, the highest Q3 on the

CHAPTER 8. CASCADED BIDIRECTIONAL RECURRENT NEURAL NETWORKS FOR PROTEIN SECONDARY STRUCTURE PREDICTION

Table 8.1: Experimental results of cascaded BRNN on protein secondary structure prediction. The cascaded BRNN has 20 input units (Xnum=20) and 3 output units (Znum=3). FYnum (BYnum)=the number of units in layer FY (BY); Ynum=the number of units in layer Y; FXnum (BXnum)=the number of units in layer FX (BX); FYnum=BYnum and FXnum=BXnum.

FYnum (BYnum)	Ynum	FXnum (BXnum)	Q3(%) (SOV)			
			Test set 1	Test set 2	Test set 3	Combined test set
8	8	8	73.75(63.918)	72.93(65.325)	74.56(67.331)	73.75(65.446)
8	8	10	73.63(63.845)	73.13(66.876)	74.65(67.420)	73.80(65.924)
8	8	15	73.93(64.359)	73.59(66.328)	75.14(68.048)	74.21(66.135)
8	10	8	74.24(65.604)	73.20(66.050)	74.52(66.000)	74.01(65.869)
8	10	10	73.80(63.567)	73.17(66.587)	74.99(68.257)	73.98(65.977)
8	10	15	74.30(64.922)	73.60(66.410)	75.07(68.083)	74.33(66.390)
10	8	10	74.07(64.348)	73.96(65.584)	74.92(66.875)	74.31(65.539)
10	8	15	74.31(64.608)	73.89(66.425)	74.95(67.351)	74.38(66.046)
10	10	10	74.10(64.590)	73.79(67.377)	75.11(66.744)	74.32(66.144)
10	10	15	74.20(64.847)	73.72(66.761)	75.17(68.088)	74.36(66.464)
15	8	15	74.38(65.061)	73.45(65.407)	75.06(67.765)	74.31(66.027)
15	10	15	74.19(64.987)	73.84(65.433)	75.01(67.968)	74.34(66.064)

Table 8.2: The best results of cascaded BRNN on different testing sets. The cascaded BRNN has 20 input units (Xnum = 20) and 3 output units (Znum = 3). FYnum (BYnum) = number of units in layer FY (BY); Ynum = number of units in layer Y; FXnum (BXnum) = number of units in layer FX (BX); FYnum = BYnum and FXnum = BXnum.

Testing set	Test set 1	Test set 2	Test set 3	Combined test set
FYnum (BYnum)	15	10	10	10
Ynum	8	8	10	8
FXnum (BXnum)	15	10	15	15
Q3(%)	74.38	73.96	75.17	74.38
Q_C(%)	82.82	78.34	83.06	82.21
Q_H(%)	77.03	75.78	75.88	75.13
Q_E(%)	53.66	64.21	58.57	58.60
SOV	65.0606	65.5839	68.0878	66.0455
SOV_C	62.1888	63.3783	66.5329	63.5815
SOV_H	74.5217	70.0465	73.7840	72.7643
SOV_E	56.0276	64.0209	61.0819	60.5657
C_C	0.55052	0.53828	0.56196	0.54907
C_H	0.69421	0.68835	0.67615	0.68824
C_E	0.53836	0.57072	0.60126	0.56379

CHAPTER 8. CASCADED BIDIRECTIONAL RECURRENT NEURAL NETWORKS FOR PROTEIN SECONDARY STRUCTURE PREDICTION

entire PSIPRED testing set (Combined test set) is obtained from a cascaded BRNN having 10 units in the layer FY (BY), 8 units in the layer Y and 15 units in the FX (BX) layer (i.e. $FY_{num}=BY_{num}=10$, $Y_{num}=8$, and $FX_{num}=BX_{num}=15$). This cascaded BRNN achieves a high SOV of 66.0455.

8.3.3 Comparison among Cascaded BRNN, PBRNN and BRNN

We summarize the best results of our cascaded BRNN, PBRNN and BRNN in Table 8.3.

In what follows, we are going to carry out a comparison among cascaded BRNN, PBRNN and the original BRNN. We firstly compare our cascaded BRNN with PBRNN and the original BRNN individually.

- Cascaded BRNN vs. the original BRNN:
 - The three-state accuracy Q3 of cascade BRNN is higher than that of the original BRNN on each testing set of Test set 1, Test set 2, Test set 3 and Combined test set;
 - Cascaded BRNN achieves higher correlation coefficients C_C, C_H and C_E on each testing subset and on the entire testing set.
 - Specifically, our cascaded BRNN obtains significant improvement in the SOV accuracy, which means cascaded BRNN gives more realistic secondary structure prediction. For Test set 1, SOV increases from 62.21 to 65.0606; for Test set 2, SOV increases from 61.67 to 65.5839; for Test set 3, SOV increases from 64.35 to 68.0878; and for the Combined test set, SOV increases from 62.96 to 66.0455.

CHAPTER 8. CASCADED BIDIRECTIONAL RECURRENT NEURAL NETWORKS FOR PROTEIN SECONDARY STRUCTURE PREDICTION

Table 8.3: Comparison among PBRNN, BRNN and Cascaded_BRNN.

measure	Test	PBRNN	BRNN	Cascaded_BRNN
Q3(%)	Test set 1	71.95	73.61	74.31
	Test set 2	71.30	72.94	73.89
	Test set 3	73.41	74.47	74.95
	Combined test set	72.21	73.67	74.38
Q_C(%)	Test set 1	80.69	82.45	84.61
	Test set 2	79.74	80.17	80.80
	Test set 3	78.49	80.24	80.70
	Combined test set	79.72	81.06	82.21
Q_H(%)	Test set 1	77.16	74.31	75.34
	Test set 2	71.43	73.48	72.90
	Test set 3	77.09	75.08	76.76
	Combined test set	75.49	74.33	75.13
Q_E(%)	Test set 1	46.62	55.09	52.44
	Test set 2	56.74	59.94	63.37
	Test set 3	57.14	62.17	60.59
	Combined test set	53.21	58.85	58.60
SOV	Test set 1	59.8223	62.2104	64.6082
	Test set 2	60.3440	62.4237	66.4247
	Test set 3	62.5845	64.3487	67.3507
	Combined test set	60.8649	62.9593	66.0455
SOV_C	Test set 1	58.7954	61.5206	62.4261
	Test set 2	61.1245	63.4552	64.0105
	Test set 3	60.1526	63.7589	64.5363
	Combined test set	59.9516	62.8277	63.5815
SOV_H	Test set 1	69.6493	67.7551	73.5481
	Test set 2	62.6777	62.4404	70.7232
	Test set 3	70.0166	66.3065	73.6217
	Combined test set	67.7118	65.7217	72.7643
SOV_E	Test set 1	46.0937	54.3339	54.7563
	Test set 2	55.7286	60.4976	65.2534
	Test set 3	54.6762	61.8782	62.2092
	Combined test set	51.9527	58.6804	60.5657
C_C	Test set 1	0.52384	0.54317	0.55203
	Test set 2	0.51965	0.52916	0.53716
	Test set 3	0.53916	0.55937	0.55721
	Combined test set	0.52753	0.54407	0.54907
C_H	Test set 1	0.63744	0.67606	0.69914
	Test set 2	0.63038	0.66561	0.69143
	Test set 3	0.64808	0.65197	0.67120
	Combined test set	0.63950	0.66564	0.68824
C_E	Test set 1	0.50012	0.52975	0.52910
	Test set 2	0.51407	0.55182	0.56811
	Test set 3	0.55812	0.59369	0.59770
	Combined test set	0.52283	0.55726	0.56379

CHAPTER 8. CASCADED BIDIRECTIONAL RECURRENT NEURAL NETWORKS FOR PROTEIN SECONDARY STRUCTURE PREDICTION

On average, cascaded BRNN improves SOV by approximately four percentage points over the PBRNN.

- Cascaded BRNN vs. PBRNN: obviously, our cascaded BRNN greatly improves performance over PBRNN in every measure of accuracy on each testing subset and the entire testing set.
 - Cascaded BRNN improves Q3 by 2-3% over PBRNN on each testing subset and the entire testing set. Cascaded BRNN's Q_C, Q_H and Q_E are higher than those of PBRNN. Especially, Q_E is increased by more than 5% on the Combined test set.
 - Cascaded BRNN's SOV is approximately 6% higher than that of PBRNN. For Test set 1, SOV increases from 59.8223 to 64.6082; for Test set 2, SOV increases from 60.3440 to 66.4247; for Test set 3, SOV increases from 62.5845 to 67.3507; and for the Combined test set, SOV increases from 60.8649 to 66.0455.
 - SOV_C, SOV_H and SOV_E of cascaded BRNN are also higher. Cascaded BRNN's SOV_E is almost 9 percent point higher than that of PBRNN.
 - Cascaded BRNN gives better correlation coefficients C_C, C_H and C_E than PBRNN on each of the three testing subsets and the entire testing set.

In summary, our cascaded BRNN improves PSS prediction accuracies over the PBRNN and the original BRNN:

- The improvement in Q3 is modest. However, among the several cascaded BRNNs of different architectural details (Table 8.1), each architecture achieves higher Q3

CHAPTER 8. CASCADED BIDIRECTIONAL RECURRENT NEURAL NETWORKS FOR PROTEIN SECONDARY STRUCTURE PREDICTION

than the PBRNNs and the original BRNNs. Even the lowest Q3 of cascaded BRNN (73.75%) is higher than PBRNN's best Q3 (72.21%) and the original BRNN's best Q3 (73.67%).

- The improvement in SOV is significant. Our cascaded BRNN yields 4-6% higher SOV than the other two BRNN architectures. Therefore, cascaded BRNN gives much more realistic secondary structure prediction. As shown in table 8.1, each cascaded BRNN achieves high SOV values ranging from 65.446 to 66.464, which are greatly higher than the best SOVs of PBRNN and the original BRNN, given that the architectural complexity of our cascaded BRNN is not higher than the other two.
- The lowest accuracies of cascaded BRNN is obtained when $Y_{num}=8$, FX_{num} (BX_{num}) = 8 and FY_{num} (BY_{num}) = 8 (Table 8.1). In such a cascaded BRNN, the total number of weights is only 1064, much less than that of PBRNN and the original BRNN. The cascaded BRNN with only 1064 weights achieves higher Q3 and much higher SOV than the PBRNN and original BRNN that have much more weights.

8.3.4 Comparison among BLSTM, BSMRNN, Cascaded BRNN, PBRNN and BRNN

All these five networks are trained with gradient descent. The time complexity grows with the number of weights and the protein length.

- The time complexity of Pollastri's BRNN is $O(TW)$, where W is the number of weights and T the protein length [9].
- LSTM algorithm's update complexity per time step is $O(W)$, where W is the number

CHAPTER 8. CASCADED BIDIRECTIONAL RECURRENT NEURAL NETWORKS FOR PROTEIN SECONDARY STRUCTURE PREDICTION

of weights [62]. BLSTM's time complexity is $O(WT)$, where T is the protein length.

- The original BRNN's time complexity is $O(WT)$, where W is the number of weights and T the protein length.
- Cascaded BRNN's time complexity is $O(WT)$, where W is the number of weights and T the protein length.
- Bidirectional SMRNN's time complexity is $O(WT)$, where W is the number of weights and T the protein length.

Hence we use the number of weights as a rough complexity measure. For each of these five models, we have carried out many experiments to obtain several predictors of different sizes and different architectural details, among which we select the one that yields the highest Q3 accuracy and give its architectural details in Table 8.4. In each architecture, the forward sub-network and backward sub-network are always symmetrical and equal-sized. As observed from Table 8.4, we have used comparable number of weights for each of these five models, except that BLSTM has much more weights than the others.

Table 8.5 summarizes the best predictions of each of the five methods developed and evaluated in this thesis. From Table 8.5, we observe:

- Q3: On each of Test set 1, Test set 2, Test set 3, and Combined test set, cascaded BRNN achieves the highest Q3, and BSMRNN achieves the second highest Q3.
- SOV: Cascaded BRNN achieves the highest SOV on Combined test set, Test set 1, Test set 2 and Test set 3; BSMRNN achieves the second highest SOV on the Combined

CHAPTER 8. CASCADED BIDIRECTIONAL RECURRENT NEURAL NETWORKS FOR PROTEIN SECONDARY STRUCTURE PREDICTION

Table 8.4: Parameters and total weights of cascaded BRNN, BSMRNN, BLSTM, Pollastri’s BRNN, and the original BRNN. Wnum is the number of parameters. m is the number of memory blocks in the forward/backward LSTM and n is the number of memory cells in each memory block.

cascaded BRNN	Xnum 20	Ynum 8	Znum 3	FXnum (BXnum) 15	FYnum (BYnum) 10	Wnum 1894
BSMRNN	Inum 20	Hnum 20	Onum 3	Fnum (Bnum) 10	FSnum (BSnum) 10	Wnum 1860
BLSTM	m 12	n 4				Wnum 13596
PBRNN	Inum 20	Hnum 15	Onum 3	Fnum (Bnum) 10	FHnum (BHnum) 15	Wnum 1605
BRNN	Inum 20	Onum 3		Fnum (Bnum) 20		Wnum 1780

test set and on each testing subset except Test set 3.

- SOV_C, SOV_H and SOV_E: Cascaded BRNN achieves the highest SOV_C, SOV_H and SOV_E on the entire set and on each subset, except that the highest SOV_C on Test set 2 and the highest SOV_E on Test set 1 are achieved by BSMRNN.
- C_C, C_H, and C_E: Cascaded BRNN and BSMRNN achieve the highest and second highest correlation coefficients on the Combined test set.

The performance of these five models could be sorted as: Cascaded BRNN > BSMRNN > BRNN > PBRNN > BLSTM. Amongst these five RNN-based models, our cascaded BRNN and BSMRNN are the best two models. Cascaded BRNN achieves the highest Q3 of 74.38% and the highest SOV of 66.0455, BSMRNN achieves the second highest Q3 of 74.10% and the second highest SOV of 63.4083.

CHAPTER 8. CASCADED BIDIRECTIONAL RECURRENT NEURAL NETWORKS FOR PROTEIN SECONDARY STRUCTURE PREDICTION

Table 8.5: Comparison among several BRNN architectures on the problem of protein secondary structure prediction.

measure	Test	PBRNN	BRNN	BLSTM	BSMRNN	Cascaded_BRNN
Q3(%)	Test set 1	71.95	73.61	71.44	74.06	74.31
	Test set 2	71.30	72.94	70.81	73.49	73.89
	Test set 3	73.41	74.47	72.33	74.73	74.95
	Combined test set	72.21	73.67	71.53	74.10	74.38
Q_C(%)	Test set 1	80.69	82.45	81.35	83.34	84.61
	Test set 2	79.74	80.17	83.74	88.28	80.80
	Test set 3	78.49	80.24	78.04	81.25	80.70
	Combined test set	79.72	81.06	81.08	84.23	82.21
Q_H(%)	Test set 1	77.16	74.31	71.50	75.06	75.34
	Test set 2	71.43	73.48	67.19	67.89	72.90
	Test set 3	77.09	75.08	72.91	77.10	76.76
	Combined test set	75.49	74.33	70.74	73.70	75.13
Q_E(%)	Test set 1	46.62	55.09	51.82	54.22	52.44
	Test set 2	56.74	59.94	53.33	55.34	63.37
	Test set 3	57.14	62.17	60.22	57.92	60.59
	Combined test set	53.21	58.85	54.83	55.70	58.60
SOV	Test set 1	59.8224	62.2104	59.5777	63.1209	64.6082
	Test set 2	60.3440	62.4237	58.2764	63.6414	66.4247
	Test set 3	62.5845	64.3487	61.1379	63.5198	67.3507
	Combined test set	60.8649	62.9593	59.6577	63.4083	66.0455
SOV_C	Test set 1	58.7954	61.5206	60.7727	61.2115	62.4261
	Test set 2	61.1245	63.4552	62.6127	66.3006	64.0105
	Test set 3	60.1526	63.7589	63.0892	61.7533	64.5363
	Combined test set	59.9516	62.8277	62.0553	62.9076	63.5815
SOV_H	Test set 1	69.6493	67.7551	62.9389	69.9816	73.5481
	Test set 2	62.6777	62.4404	56.7479	64.6789	70.7232
	Test set 3	70.0166	66.3065	60.1386	68.3406	73.6217
	Combined test set	67.7118	65.7217	60.1411	67.8956	72.7643
SOV_E	Test set 1	46.0937	54.3339	51.2242	55.8135	54.7563
	Test set 2	55.7286	60.4976	52.6909	57.5977	65.2534
	Test set 3	54.6762	61.8782	59.0713	58.1880	62.2092
	Combined test set	51.9527	58.6804	54.0219	57.1363	60.5657
C_C	Test set 1	0.52384	0.54317	0.51529	0.54588	0.55203
	Test set 2	0.51965	0.52916	0.50867	0.54517	0.53716
	Test set 3	0.53916	0.55937	0.52669	0.55624	0.55721
	Combined test set	0.52753	0.54407	0.51659	0.54821	0.54907
C_H	Test set 1	0.63744	0.67606	0.63195	0.68872	0.69914
	Test set 2	0.63038	0.66561	0.62223	0.69494	0.69143
	Test set 3	0.64808	0.65197	0.61903	0.66774	0.67120
	Combined test set	0.63950	0.66564	0.62566	0.68344	0.68824
C_E	Test set 1	0.50012	0.52975	0.49679	0.53621	0.52910
	Test set 2	0.51407	0.55182	0.51801	0.55039	0.56811
	Test set 3	0.55812	0.59369	0.55820	0.58915	0.59770
	Combined test set	0.52283	0.55726	0.52261	0.55736	0.56379

8.3.5 Statistical Significance Test

In statistics, a result is called significant if it is unlikely to have occurred by chance. "A statistically significant difference" simply means there is statistical evidence that there is a difference; it does not mean the difference is necessarily large, important or significant in the usual sense of the word. There are five steps in testing for statistical significance:

- State the Research Hypothesis H_1 .
- State the Null Hypothesis H_0 .
- Select a probability of error level (α level).
- Select and compute the test for statistical significance.
- Interpret the results.

T Test

In this section, we carry out t test to find out whether the results obtained by different PSS prediction methods are significantly different from each other. T test is a test of the null hypothesis that the means of two normally distributed populations are equal. Given two data sets, each characterized by its mean, standard deviation and number of data points, we can use some kind of t test to determine whether the means are distinct, provided that the underlying distributions can be assumed to be normal. All such tests are usually called Student's t tests, though strictly speaking that name should only be used if the variances of the two populations are also assumed to be equal. A common application of this is to test if a new process or treatment is superior to a current process or treatment.

CHAPTER 8. CASCADED BIDIRECTIONAL RECURRENT NEURAL NETWORKS FOR PROTEIN SECONDARY STRUCTURE PREDICTION

The student's t test for unpaired data (i.e. independent student's t test) is defined as:

$$H_0 : \mu_1 = \mu_2$$

$$H_1 : \mu_1 \neq \mu_2$$

$$\text{TestStatistics} : t = \frac{\overline{X}_1 - \overline{X}_2}{s_{\overline{X}_1 - \overline{X}_2}}$$

$$\text{where } s_{\overline{X}_1 - \overline{X}_2} = \sqrt{\frac{(n_1 - 1)s_1^2 + (n_2 - 1)s_2^2}{n_1 + n_2 - 2} \left(\frac{1}{n_1} + \frac{1}{n_2} \right)}$$

$$\text{Significancelevel} : \alpha$$

CriticalRegion : Reject the null hypothesis that the two means are equal if

$$t < -t_{\alpha/2}(v) \text{ or } t > t_{\alpha/2}(v)$$

$$\text{where } v = \frac{(s_1^2/n_1 + s_2^2/n_2)^2}{(s_1^2/n_1)^2/(n_1 - 1) + (s_2^2/n_2)^2/(n_2 - 1)}$$

if equal variances are assumed, then $v = n_1 + n_2 - 2$

where 1=group one, 2=group two, μ_1 is the mean of group one, μ_2 is the mean of group two, s_1^2 is the sample variance of group one, s_2^2 is the sample variance of group two, $\overline{X}_1$ is the sample mean of group one, $\overline{X}_2$ is the sample mean of group two, n_1 is the sample size of group one, n_2 is the sample size of group two.

F test

Student's t test requires the assumption that the variances of two groups are equal. Therefore, before students' t test, we use F test to validate the equality of variances. An F test is used to test if the standard deviations of two populations are equal. This test can be a two-tailed test or a one-tailed test. The two-tailed version tests against the alternative that the standard deviations are not equal. The one-tailed version only tests in one direction,

CHAPTER 8. CASCADED BIDIRECTIONAL RECURRENT NEURAL NETWORKS FOR PROTEIN SECONDARY STRUCTURE PREDICTION

that is the standard deviation from the first population is either greater than or less than (but not both) the second population standard deviation. The null and research hypothesis of one-tailed and two-tailed F test is given as:

$$H_0 : \sigma_1^2 = \sigma_2^2$$

$$H_1 : \sigma_1^2 < \sigma_2^2 \text{ for a lower one tailed test}$$

$$\sigma_1^2 > \sigma_2^2 \text{ for an upper one tailed test}$$

$$\sigma_1 \neq \sigma_2 \text{ for a two tailed test}$$

In our problem, we choose two-tailed F test, which is defined as:

$$\text{Test Statistic} : F = \frac{n_{big}s_{big}^2/(n_{big} - 1)}{n_{small}s_{small}^2/(n_{small} - 1)}$$

Significance level : α

Critical region : The hypothesis that the two variances are equal is rejected if

$$F < F_{1-\alpha/2}(n_{big} - 1, n_{small} - 1) \text{ or } F > F_{\alpha/2}(n_{big} - 1, n_{small} - 1)$$

where

$$n_{big} = n_1, s_{big} = s_1, n_{small} = n_2, s_{small} = s_2 \quad \text{if} \quad n_1 s_1^2 / (n_1 - 1) > n_2 s_2^2 / (n_2 - 1)$$

$$n_{big} = n_2, s_{big} = s_2, n_{small} = n_1, s_{small} = s_1 \quad \text{if} \quad n_1 s_1^2 / (n_1 - 1) \leq n_2 s_2^2 / (n_2 - 1)$$

Significance Test on Q3 Accuracies of Our experiments

In our experiments, all the results we obtained are the average of the results of ten experiments. Table 8.6 summarizes the sample mean and sample variance of Q3 accuracies on the combined test set, i.e. the union of test set 1, test set 2 and test set 3.

CHAPTER 8. CASCADED BIDIRECTIONAL RECURRENT NEURAL NETWORKS FOR PROTEIN SECONDARY STRUCTURE PREDICTION

Table 8.6: Q3 distribution of BRNN, PBRNN, BSMRNN and Cascaded BRNN on the combined test set.

	PBRNN	BRNN	BSMRNN	cascaded BRNN
sample mean(%)	72.2093	73.6727	74.0953	74.3834
sample variance	0.1983	0.2036	0.2517	0.2224

we use the following symbols to notate the sample mean, sample variance, mean, variance and sample size:

- $\bar{X}_{PBRNN}, \bar{X}_{BRNN}, \bar{X}_{BSMRNN}, \bar{X}_{CBRNN}$ are sample mean of Q3 of PBRNN, BRNN, BSMRNN, cascaded BRNN.
- $s_{PBRNN}^2, s_{BRNN}^2, s_{BSMRNN}^2, s_{CBRNN}^2$ are sample variance of Q3 of PBRNN, BRNN, BSMRNN, cascaded BRNN.
- $\mu_{PBRNN}, \mu_{BRNN}, \mu_{BSMRNN}, \mu_{CBRNN}$ are mean of Q3 of PBRNN, BRNN, BSMRNN, cascaded BRNN.
- $\sigma_{PBRNN}^2, \sigma_{BRNN}^2, \sigma_{BSMRNN}^2, \sigma_{CBRNN}^2$ are variance of Q3 of PBRNN, BRNN, BSMRNN, cascaded BRNN.
- $n_{PBRNN}, n_{BRNN}, n_{BSMRNN}, n_{CBRNN}$ are sample size of PBRNN, BRNN, BSMRNN, cascaded BRNN.
- $\alpha = 0.01$ is the significance level.

In our problem, sample mean and sample variance of Q3 are known, while mean and variance of Q3 are unknown. From Table 8.6, we have

$$\bar{X}_{PBRNN} = 72.2093, \bar{X}_{BRNN} = 73.6727, \bar{X}_{BSMRNN} = 74.0953, \bar{X}_{CBRNN} = 74.3834,$$

CHAPTER 8. CASCADED BIDIRECTIONAL RECURRENT NEURAL NETWORKS FOR PROTEIN SECONDARY STRUCTURE PREDICTION

$$s_{PBRNN}^2 = 0.1983, s_{BRNN}^2 = 0.2036, s_{BSMRNN}^2 = 0.2517, s_{CBRNN}^2 = 0.2224,$$

$$n_{PBRNN} = 10, n_{BRNN} = 10, n_{BSMRNN} = 10, n_{CBRNN} = 10$$

Cascaded BRNN vs. BRNN

(1) F test:

$$H_0 : \sigma_{CBRNN}^2 = \sigma_{BRNN}^2, H_1 : \sigma_{CBRNN}^2 \neq \sigma_{BRNN}^2$$

$s_{CBRNN}^2 > s_{BRNN}^2$, $n_{CBRNN} = n_{BRNN}$, hence

$$F = \frac{s_{CBRNN}^2}{s_{BRNN}^2} = \frac{0.2224}{0.2036} = 1.092338 \quad (8.31)$$

Using MatLAB function `finv()`, we calculate the upper and lower critical values of F:

$$F_{\alpha/2}(n_{CBRNN} - 1, n_{BRNN} - 1) = F_{0.005}(9, 9) = \text{finv}(0.995, 9, 9) = 6.5411 \quad (8.32)$$

$$F_{1-\alpha/2}(n_{CBRNN} - 1, n_{BRNN} - 1) = F_{0.995}(9, 9) = \text{finv}(0.005, 9, 9) = 0.1529 \quad (8.33)$$

$F_{0.995}(9, 9) = 0.1529 < F = 1.092338 < 6.5411 = F_{0.005}(9, 9)$, hence we accept the null hypothesis, i.e. $\sigma_{CBRNN}^2 = \sigma_{BRNN}^2$.

(2) T test:

$$H_0 : \mu_{CBRNN} = \mu_{BRNN}, H_1 : \mu_{CBRNN} \neq \mu_{BRNN}$$

The value of t is calculated by:

$$t = \frac{74.3834 - 73.6727}{\sqrt{\frac{0.2224^2 + 0.2036^2}{10}}} = 7.4537 \quad (8.34)$$

The critical value of t is calculated by MatLAB function `tinu()`:

$$t_{0.01}(10 + 10 - 2) = t_{0.01}(18) = \text{tinu}(0.99, 18) = 2.5524 \quad (8.35)$$

CHAPTER 8. CASCADED BIDIRECTIONAL RECURRENT NEURAL NETWORKS FOR PROTEIN SECONDARY STRUCTURE PREDICTION

Because $7.4537 > 2.5524$, we reject the null hypothesis $\mu_{CBRNN} = \mu_{BRNN}$. There is statistically significant difference between the Q3 of cascaded BRNN and that of BRNN.

Cascaded BRNN vs. PBRNN

(1) F test:

$$H_0 : \sigma_{CBRNN}^2 = \sigma_{PBRNN}^2, H_1 : \sigma_{CBRNN}^2 \neq \sigma_{PBRNN}^2$$

$s_{CBRNN}^2 > s_{PBRNN}^2$, $n_{CBRNN} = n_{PBRNN}$, hence

$$F = \frac{s_{CBRNN}^2}{s_{PBRNN}^2} = \frac{0.2224}{0.1983} = 1.1215 \quad (8.36)$$

Using MatLAB function `finv( )`, we calculate the upper and lower critical values of F:

$$F_{\alpha/2}(n_{CBRNN} - 1, n_{PBRNN} - 1) = F_{0.005}(9, 9) = \text{finv}(0.995, 9, 9) = 6.5411 \quad (8.37)$$

$$F_{1-\alpha/2}(n_{CBRNN} - 1, n_{PBRNN} - 1) = F_{0.995}(9, 9) = \text{finv}(0.005, 9, 9) = 0.1529 \quad (8.38)$$

$F_{0.995}(9, 9) = 0.1529 < F = 1.1215 < 6.5411 = F_{0.005}(9, 9)$, hence we accept the null hypothesis, i.e. $\sigma_{CBRNN}^2 = \sigma_{PBRNN}^2$.

(2) T test:

$$H_0 : \mu_{CBRNN} = \mu_{PBRNN}, H_1 : \mu_{CBRNN} \neq \mu_{PBRNN}$$

The value of t is calculated by:

$$t = \frac{74.3834 - 72.2093}{\sqrt{\frac{0.2224^2 + 0.1983^2}{10}}} = 23.0733 \quad (8.39)$$

The critical value of t is calculated by MatLAB function `tin( )`:

$$t_{0.01}(10 + 10 - 2) = t_{0.01}(18) = \text{tin}(0.99, 18) = 2.5524 \quad (8.40)$$

CHAPTER 8. CASCADED BIDIRECTIONAL RECURRENT NEURAL NETWORKS FOR PROTEIN SECONDARY STRUCTURE PREDICTION

Because $23.0733 > 2.5524$, we reject the null hypothesis $\mu_{CBRNN} = \mu_{PBRNN}$. There is statistically significant difference between the Q3 of cascaded BRNN and that of PBRNN.

BSMRNN vs. BRNN

(1) F test:

$$H_0 : \sigma_{BSMRNN}^2 = \sigma_{BRNN}^2, H_1 : \sigma_{BSMRNN}^2 \neq \sigma_{BRNN}^2$$

$s_{BSMRNN}^2 > s_{BRNN}^2$, $n_{BSMRNN} = n_{BRNN}$, hence

$$F = \frac{s_{BSMRNN}^2}{s_{BRNN}^2} = \frac{0.2517}{0.2036} = 1.2362 \quad (8.41)$$

Using MatLAB function `finv( )`, we calculate the upper and lower critical values of F:

$$F_{\alpha/2}(n_{BSMRNN} - 1, n_{BRNN} - 1) = F_{0.005}(9, 9) = \text{finv}(0.995, 9, 9) = 6.5411 \quad (8.42)$$

$$F_{1-\alpha/2}(n_{BSMRNN} - 1, n_{BRNN} - 1) = F_{0.995}(9, 9) = \text{finv}(0.005, 9, 9) = 0.1529 \quad (8.43)$$

$F_{0.995}(9, 9) = 0.1529 < F = 1.2362 < 6.5411 = F_{0.005}(9, 9)$, hence we accept the null hypothesis, i.e. $\sigma_{BSMRNN}^2 = \sigma_{BRNN}^2$.

(2) T test:

$$H_0 : \mu_{BSMRNN} = \mu_{BRNN}, H_1 : \mu_{BSMRNN} \neq \mu_{BRNN}$$

The value of t is calculated by:

$$t = \frac{74.0953 - 73.6727}{\sqrt{\frac{0.2517^2 + 0.2036^2}{10}}} = 4.1280 \quad (8.44)$$

The critical value of t is calculated by MatLAB function `tinv( )`:

$$t_{0.01}(10 + 10 - 2) = t_{0.01}(18) = \text{tinv}(0.99, 18) = 2.5524 \quad (8.45)$$

CHAPTER 8. CASCADED BIDIRECTIONAL RECURRENT NEURAL NETWORKS FOR PROTEIN SECONDARY STRUCTURE PREDICTION

Since $4.1280 > 2.5524$, we reject the null hypothesis $\mu_{BSMRNN} = \mu_{BRNN}$. There is statistically significant difference between the Q3 of BSMRNN and that of BRNN.

BSMRNN vs. PBRNN

(1) F test:

$$H_0 : \sigma_{BSMRNN}^2 = \sigma_{PBRNN}^2, H_1 : \sigma_{BSMRNN}^2 \neq \sigma_{PBRNN}^2$$

$s_{BSMRNN}^2 > s_{PBRNN}^2$, $n_{BSMRNN} = n_{PBRNN}$, hence

$$F = \frac{s_{BSMRNN}^2}{s_{PBRNN}^2} = \frac{0.2517}{0.1983} = 1.2693 \quad (8.46)$$

Using MatLAB function `finv( )`, we calculate the upper and lower critical values of F:

$$F_{\alpha/2}(n_{BSMRNN} - 1, n_{PBRNN} - 1) = F_{0.005}(9, 9) = \text{finv}(0.995, 9, 9) = 6.5411 \quad (8.47)$$

$$F_{1-\alpha/2}(n_{BSMRNN} - 1, n_{PBRNN} - 1) = F_{0.995}(9, 9) = \text{finv}(0.005, 9, 9) = 0.1529 \quad (8.48)$$

$F_{0.995}(9, 9) = 0.1529 < F = 1.2693 < 6.5411 = F_{0.005}(9, 9)$, hence we accept the null hypothesis, i.e. $\sigma_{BSMRNN}^2 = \sigma_{PBRNN}^2$.

(2) T test:

$$H_0 : \mu_{BSMRNN} = \mu_{PBRNN}, H_1 : \mu_{BSMRNN} \neq \mu_{PBRNN}$$

The value of t is calculated by:

$$t = \frac{74.0953 - 72.2093}{\sqrt{\frac{0.2517^2 + 0.1983^2}{10}}} = 18.6126 \quad (8.49)$$

The critical value of t is calculated by MatLAB function `tinvt( )`:

$$t_{0.01}(10 + 10 - 2) = t_{0.01}(18) = \text{tinvt}(0.99, 18) = 2.5524 \quad (8.50)$$

CHAPTER 8. CASCADED BIDIRECTIONAL RECURRENT NEURAL NETWORKS FOR PROTEIN SECONDARY STRUCTURE PREDICTION

Since $18.6126 > 2.5524$, we reject the null hypothesis $\mu_{BSMRNN} = \mu_{PBRNN}$. There is statistically significant difference between the Q3 of BSMRNN and that of PBRNN.

8.4 Discussion

We have addressed the strong, long-range structure-structure correlations in protein sequences and developed a novel architecture called cascaded bidirectional recurrent neural network. The structure-to-structure BRNN in cascaded BRNN has been proved to successfully capture correlations between secondary structure elements. Compared with the PBRNN and the original BRNN, the secondary structure predictor based on cascaded BRNN is able to improve the Q3 accuracy and the correlation coefficients for each class of secondary structure, and to increase the value of SOV significantly.

Chapter 9

Summary of This Research

9.1 Summary

Prediction of protein secondary structure is one of the classical problems in bioinformatics, and is still far from being satisfactorily solved. Most of existing predictors use a sliding fixed-sized window centered on the predicted residue to take into account only local interactions between amino acids. A major drawback of local window approaches is that they cannot exploit relevant information contained in distant portion of the protein sequence. By using the concept of mutual information, we formulate the inter-residue interactions, especially long-range interactions, i.e. those interactions between amino acids far apart along the protein chain. Our investigations show that long-range interactions commonly exist in protein folding, and play an important role in the formation of protein secondary structures (PSS). Moreover, we reveal that long-range interactions have negative effect on PSS prediction accuracy of local window approaches. Prediction accuracy falls as the sequence separation between contacting residues increases. We suggest that neglecting long-range interactions is the main reason for the insufficient prediction accuracy of methods based on local information only. Hence, in order to improve prediction accuracy, we should make

CHAPTER 9. SUMMARY OF THIS RESEARCH

use of the whole protein sequence instead of fixed-length segments, and exploit both local information and distant information for PSS prediction. There are two typical models for processing strings of variable length, namely stochastic grammar and recurrent neural network (RNN). So far, neither stochastic grammar nor RNN has been satisfactorily applied to predict protein secondary structure. In consideration of complexity and feasibility, we choose RNNs as our candidate model for PSS prediction at this stage.

The conventional and widely used RNN architectures are Elman's network and its variations. Learning algorithms for Elman-style networks are usually based on gradient descent. There are theoretical and experimental evidence showing that Elman-style networks have difficulties in learning long-term dependencies with gradient descent [13]. The training of Elman-style networks becomes more and more difficult as the duration of the dependencies to be captured increases. Hence, the Elman-style networks have difficulties in capturing long-range interactions in proteins for PSS prediction. In order to deal with this problem, we propose a novel architecture called segmented-memory recurrent neural network (SMRNN), which tries to memorize long sequences in segments. The SMRNN uses separate hidden layers of states to store contextual information respectively at symbol-level and at segment-level. The input sequence is broken into segments and presented to the SMRNN one symbol per cycle. Context at symbol-level is updated for each symbol presented for input, while context at segment-level is updated only at the end of each segment. In other words, the SMRNN memorizes each segment first and then cascades them to form the final sequence. Besides the SMRNN, we also adopt an RNN architecture and algorithm called Long Short-Term Memory (LSTM) that can learn to bridge long time lags.

CHAPTER 9. SUMMARY OF THIS RESEARCH

Another limitation of RNN is its causality. A dynamic system is causal if the output only depends on the current and previous input values and not on any future input values. Obviously, PSS prediction is a non-causal problem in which the secondary structure type of an amino acid depends on both upstream and downstream information in the protein sequence. Both the LSTM and SMRNN are causal systems, which makes them unsuited for PSS prediction. To remove this limitation, we propose bidirectional LSTM (BLSTM) and bidirectional SMRNN (BSMRNN). The BLSTM/BSMRNN uses a forward LSTM/SMRNN and a backward LSTM/SMRNN to capture upstream and downstream context respectively. Then, a feed-forward sub-network placed on top of the forward and backward recurrent sub-networks combines the upstream context, the residue of interest and the downstream context to provide an output. The final output at sequence position i is a vector of posterior probabilities of secondary structure classes for the i th residue given the protein's sequence of amino acids.

Prior research on protein secondary structure suggests that the secondary structure at a particular location also depends on the secondary structures in the neighborhood. Our statistical analysis on the EVA protein set discovers strong and non-local correlations between secondary structure elements. In order to incorporate the correlations among the elements of the secondary structure sequence, we propose a cascaded BRNN consisting of one BRNN on top of the other BRNN. The high-level BRNN is used for capturing the context of neighboring secondary structures, and the low-level BRNN captures the context of amino acids. Unlike PHD's sequential training of the first network and then the second one, the high-level and low-level BRNNs in cascade BRNN are trained simultaneously to

converge to a common solution.

In summary, we have proposed and developed three novel recurrent neural network architectures for protein secondary structure prediction. They are bidirectional long short-term memory (BLSTM), bidirectional segmented-memory recurrent neural network (BSMRNN) and cascaded bidirectional recurrent neural network (Cascaded BRNN). For the purpose of comparison and evaluation, we also constructed two earlier RNN-based prediction models, namely Pollastri's bidirectional recurrent neural network (PBRNN) and the original bidirectional recurrent neural network (BRNN) that were proposed by other researchers. These five models are all trained and tested on the PSIPRED training/testing data sets.

9.2 Achievements

The five models evaluated in this thesis, including our BLSTM, BSMRNN, cascaded BRNN, Pollastri's BRNN and the original BRNN, are all RNN-based models. All of them make use of the whole protein sequence to consider both local and long-range interactions for PSS prediction. They are promising alternatives to local window-based approaches. The advantage of using bidirectional RNN architectures rather than local fixed-sized windows has been proved by the earlier work of Pollastri and his colleagues [7, 105, 102, 104]. Our work has illustrated the advantage of our BSMRNN and cascaded BRNN over Pollastri's BRNN and the original BRNN.

Segmented-memory recurrent neural network proposed here is a novel RNN architecture that is capable of learning long-term dependencies. In order to assess the perfor-

CHAPTER 9. SUMMARY OF THIS RESEARCH

mance of SMRNN, we have conducted two experiments, namely information latching and “2-sequence” problems, and carried out a comparison between SMRNN and conventional RNNs. The results obtained indicate that SMRNN performs better on long-term dependency problems. Our experiments on the PSIPRED protein data set indicate that bidirectional SMRNN achieves higher PSS prediction accuracies than the BLSTM, Pollastri’s BRNN and the original BRNN.

Pollastri’s BRNN is the most recent existing model that has been reported to exploit non-local information for PSS prediction. PHD is the first one to incorporate the correlations between adjacent secondary structure (SS) patterns. To the best of our knowledge, no prior methods are able to consider both long-range interactions and SS-SS correlations in one single model. Our cascaded BRNN for the first time successfully captures both long-range interactions and SS-SS correlations for PSS prediction. It outperforms Pollastri’s BRNN and the original BRNN in both Q3 and SOV. Specifically, it greatly improves the SOV score.

Our BSMRNN and cascaded BRNN slightly outperform Pollastri’s BRNN and the original BRNN in Q3. Protein secondary structure prediction is a classical difficult problem. It is not easy to achieve even small improvements given the large amount of work that has been reported in the literature. During the last few years, not much progress was made on this topic. Moreover, given the large number of protein sequences available through genome and other sequencing projects, even small percentage improvement in secondary structure prediction can be significant. The SOV is greatly improved by cascaded BRNN. It is proved that cascaded BRNN successfully incorporates the correlations between secondary

CHAPTER 9. SUMMARY OF THIS RESEARCH

structure patterns, and gives more realistic secondary structure prediction. Finally, the prediction performance of our models is comparable to that of the best existing methods for PSS prediction.

All of our three methods including BLSTM, BSMRNN and cascaded BRNN are variants of bidirectional recurrent neural networks. They make use of the whole protein sequences instead of local segments, and take into account long-range interactions. BLSTM/BSMRNN and cascaded BRNN try to improve protein secondary structure prediction by taking care of different aspects of the problem. BLSTM and BSMRNN are more capable of learning long-term dependencies between amino acids that are involved in long-range interactions. Cascaded BRNN is able to incorporate the correlations between secondary structural elements in prediction. The ideas of BLSTM and BSMRNN can be combined with that of cascaded BRNN, which means a cascaded BLSTM or cascaded BSMRNN will achieve even higher accuracies. Furthermore, as stated in the literature, mixture of estimators has proved to be efficient to increase the generalization performance of individual model. We expect benefits from combining the predictions of BLSTM, BSMRNN and cascaded BRNN by computing an arithmetic average of the output values of these three models.

Out of our three methods, cascaded BRNN performs the best on the protein secondary structure prediction problem. However, we do not suggest using cascaded BRNN in all situations. Each of them owns its specific advantages that are desirable for different applications. BLSTM and BSMRNN are efficient in solving those sequential processing problems where long-term dependencies are playing important role. Besides protein structure prediction, BLSTM has also found some applications in the area of speech recognition. BSMRNN

CHAPTER 9. SUMMARY OF THIS RESEARCH

is preferable for tasks on sequential data in which there are segmental patterns and long-term dependencies. In our present work, cascaded BRNN is especially proposed for protein secondary structure prediction given that both long-range interactions and correlations between secondary structures have effect on secondary structure prediction. As is known, several regularities are observed in the secondary structure sequences of proteins. And our statistical study also indicates that secondary structural elements are strongly correlated along the chain. Hence, cascaded BRNN performs well by using a second-level BRNN to refine the raw prediction obtained from the first-level BRNN. Therefore, in cases where there are strong correlations between the elements of the predicted sequences, cascaded BRNN will be greatly helpful.

9.3 Future Work

9.3.1 Protein Grammar

The related work in the field of “biological linguistics”, especially SCFGs for RNA modeling, suggests that a linguistic approach to PSS prediction is an appealing alternative to probabilistic methods. As defined in chapter 1, protein sequences are strings over the amino acid alphabet. Collections of protein sequences can be viewed as protein language generated by protein grammar. The syntactic structure or the corresponding parse tree of a protein sequence provides a candidate for optimal folding that can be used to predict secondary structure. Thus, one goal of our future work is to construct a PSS predictor based on protein grammar.

Appropriate Grammar with Efficient Learning and Parsing Algorithm

Searls [129, 130, 131] has suggested that the languages of biology also fall into the Chomsky hierarchy. In Chomsky hierarchy, for classes of grammars having higher representational capability, the parsing and learning algorithms are more computationally complex. Thus one needs to construct an efficiently parsable and learnable grammar of protein language, which has sufficient but not overmuch expressive power.

Stochastic context-free grammar has been successfully applied to RNA secondary structure prediction. Our task is similar to this but much more complicated. Firstly, the size of amino acid alphabet is 20, much larger than that of RNA sequences. Secondly, the interactions between amino acids are much more complicated. In RNA, hydrogen bonds form mainly between complementary bases, C-G and A-U, which are called Watson-Crick base pairs. While in protein, H-bond can be formed between any pair of amino acids. Finally, the dependencies exhibited by elements of RNA secondary structure (except pseudoknots) are local and non-crossed so they can be easily captured by context-free grammar. In the case of protein, the non-local and crossed correlation among amino acids is far beyond context-free, therefore more powerful classes of grammars should be considered.

Context-free grammars do not have enough expressive power for protein language while context-sensitive grammars (CSGs), are unattractive because for them parsing is an NP-complete problem. There lies a class of grammars between CFGs and CSFs, namely mildly context-sensitive grammars (MCSGs), which are more expressively powerful than CFG while still can be parsed and learned in polynomial time. To our best knowledge, MCSGs have not led to successful protein secondary structure prediction due to the high computa-

CHAPTER 9. SUMMARY OF THIS RESEARCH

tional requirement of the parsing algorithm and the training algorithm. Thus one aim of our future research will be to construct an appropriate MCSG for protein language and to design efficient parsing and learning algorithms for MCSGs.

9.3.2 Integration of Protein Grammar and Connectionist Approaches

For processing sequential data, there are two typical methods, traditional grammatical inference (symbolic approaches) and recurrent neural networks (connectionist approaches). Symbolic and connectionist approaches have been shown to have different strengths and weaknesses. The symbolic method deals with high-level information formulated via logical formulas. Data processing is human-understandable, hence it is easy to combine prior knowledge, or to retrain the system on additional data. However, grammar learning is complex, inefficient, and sensitive to noise. To the best of our knowledge, very few successful applications of grammar theory to PSS prediction have been reported. In contrast, RNN deals with low-level information. Its behavior is not human-understandable, and it is not easy to incorporate prior knowledge, or additional data requires complete retraining. Nevertheless, training is efficient, noise tolerant, and robust. Recently, much research attention has been paid to the integration of connectionist and symbolic approaches. Such integration allows combining the bottom-up (learning from data) and top-down (deductive reasoning) kinds of inference. Two relevant aspects of the integration are insertion of prior rules into the initial recurrent networks and extraction of rules from the trained recurrent networks. They allow the exchange of information between symbolic and connectionist knowledge representations. By integrating RNN-based prediction models and protein grammar, we

will ultimately obtain a robust and efficient linguistic model for PSS prediction.

9.4 Publications

Journal Paper Published

J. Chen and N. S. Chaudhari, Bidirectional Segmented-Memory Recurrent Neural Network for Protein Secondary Structure Prediction, *Soft Computing Journal*, 10:315-324, 2005.

J. Chen and N. S. Chaudhari, Cascaded Bidirectional Recurrent Neural Networks for Protein Secondary Structure Prediction, *IEEE Transaction on Computational Biology and Bioinformatics*, accepted, December 2006.

Conference Papers Published in Proceedings

J. Chen and N. S. Chaudhari, Statistical Analysis of Long-Range Interactions in Proteins, In *Proceedings of the 2006 International Conference on Bioinformatics & Computational Biology*, pages 296-302, Las vegas, USA, June 26-29, 2006.

J. Chen and N. S. Chaudhari, Protein Secondary Structure Prediction with a Hybrid RNN/HMM System, In *Proceedings of the International Joint Conference on Neural Networks (IJCNN 2005)*, volume 1, pages 538-541, Montréal, Canada, July 31-August 4, 2005.

J. Chen and N. S. Chaudhari, Learning Long-Term Dependencies in Segmented Memory Recurrent Neural Networks, In *Proceedings of the International Symposium on Neural Networks (ISNN 2004)*, Dalian, China, August 19-21, 2004. Also published in, *Advances in Neural Networks: Part I Lecture Notes in Computer Science*, Volume 3173, pages 362-369, Springer Verlag, Berlin, 2004.

J. Chen and N. S. Chaudhari, Capturing Long-Term Dependencies for Protein Secondary

CHAPTER 9. SUMMARY OF THIS RESEARCH

Structure Prediction, In *Proceedings of the International Symposium on Neural Networks (ISNN 2004)*, Dalian, China, August 19-21, 2004. Also published in *Advances in Neural Networks: Part II Lecture Notes in Computer Science*, Volume 3174, pages 494-500, Springer Verlag, Berlin, 2004.

J. Chen and N. S. Chaudhari, Improvement of Bidirectional Recurrent Neural Network for Learning Long-Term Dependencies, In *Proceedings of the 17th International Conference on Pattern Recognition (ICPR)*, volume 4, pages 593-596, Cambridge, U.K., August 23-26, 2004.

J. Chen and N. S. Chaudhari, Protein Family Classification Using Second-Order Recurrent Neural Networks, In *Proceedings of the 14th International Conference on Genome Informatics (GIW 2003)* organized by Japanese Society for Bioinformatics (JSBI <http://www.jsbi.org/>), Also in *Genome Informatics*, Volume 14, pages 520-521, Tokyo, Japan, December 13-15, 2003.

J. Chen and N. S. Chaudhari, Improvement of the inside-outside algorithm using prediction and application to RNA modeling, In *Proceedings of the Second International Conference on Machine Learning and Cybernetics (IC-MLC 2003)*, (IEEE Cat. No.03EX693), Volume 4, pages 2255-2260, Xi-an, China, November 02-05, 2003.

Appendix

Changed PDB IDs of EVA Proteins

The PDB IDs of some proteins in the EVA set are replaced by new ones (Table 9.1). Refer to <http://www.rcsb.org/pdb/>, and Search the pdb database with old ids.

Table 9.1: Changed PDB ids of the EVA proteins.

original id	new id	original id	new id	original id	new id	original id	new id
1O09	1P9K	1GDG	1KW8	1NNG	1YLI	1KC0	1N2S
1P9V	1SDJ	1H40	1UZC	1TGQ	2B95	1KOM	1T23
1NWJ	1Q53	1IKZ	1M3G	1Z61	1ZAE	1KYG	1YEP
1NVH	1QXF	1J5R	1O2D	1e53	1z60	1PLZ	1X82
1NJ7	1T50	1K8E	1O17	1dhx	1p2z	1QA8	1GFW
1LW8	1Q4V	1LNR	1NKW	1ddc	1h21	1UM3	1VF5
1LQR	1TYK	1TSG	1O7B	1c0d	1vrx	2OSP	1FJ1
1L7U	1M40	1UOX	1R51	1byt	1no3	1aa8	1ve9
1L2R	1NTH	2STV	2BUK	1b91	1n72	1abm	1n0j
1L1A	1MUR	1E01	1E0G	1FH6	1OHG	1az9	1wl9
1KYG	1YEP	1EKT	1Z0R	1FJF	1J5E	1KA3	1TKV
1D5K	1GH8	1EUL	1SU4	1FVZ	1T27	1JYG	1RYK
1DW7	1GH9	1GR8	1OCK	1H77	1HK8	1JTR	1MWA
1HLR	1VLB	1IB3	1M8Z	1IJ7	1M9U	1JSB	1RYJ
1IYA	1J3G	1JFY	1JR7	1JGF	1LS5	1JMR	1MKM
1JJY	1SU8						

Theoretical Model Structures in EVA set

The pdb files may be determined experimentally by x-ray diffraction or by NMR, or may be generated as a theoretical model structure. The PDB separated theoretical model coordinate files from the main archive beginning July 1, 2002. Since that date, the main archive has consisted of structures determined using experimental methods only. All theoretical models (current and obsolete) are kept in a separate location in the FTP archive (<ftp://pub.pdb/data/structures/models/obsolete>). The theoretical model structures in the EVA list are given below (Table 9.2).

Table 9.2: Theoretical model structures in EVA Set.

1LXB	1LLL	1LJB	1LGO	1LFN	1LA7	1K14	1LLY	1GTY	1LGZ
1LX9	1LLK	1LJA	1LGJ	1LFE	1L9I	1K15	1GOP	1FL4	1LJC
1LX2	1LLJ	1LJ6	1LGI	1LEU	1L8M	1KOV	1H8Q	1LS7	1LU7
1LX1	1LLG	1LHX	1LG3	1LER	1L6Q	1KPN	1I2J	1HN5	1H8R
1LWP	1LLE	1LHQ	1LG0	1LDV	1DL9	1KPW	1K13	1fzn	1J5G
1LVI	1LKW	1LHA	1LFX	1LD0	1DNQ	1KPX	1JX5	1JX3	1dt8
1LUO	1LKG	1LH9	1LFR	1LCH	1DNR	1L6Q	1K11	1IC3	1ICZ
1LUB	1I3B	1FH4	1LH8	1LK8					

Redundant Entries in EVA set

There are 10 pairs of entries, for which different PDB IDs refer to the same sequence (Table 9.3).

Prediction Methods Evaluated

Table 9.4 gives a detailed list of the six protein secondary structure prediction methods examined in Chapter 4.

Table 9.3: Redundant Entries in EVA set

1fjf_B - 1j5e_B	1fjf_L - 1j5e_L
1fjf_C - 1j5e_C	1fjf_M - 1j5e_M
1fjf_I - 1j5e_I	1fjf_N - 1j5e_N
1fjf_J - 1j5e_J	1fjf_T - 1j5e_T
1fjf_K - 1j5e_K	1e01_A - 1e0g_A

Table 9.4: Protein secondary structure prediction servers participating.

Server	APSSP [110]
URL	http://imtech.res.in/raghava/apssp/
Description	Advanced Protein Secondary Structure Prediction
Authors	Raghava, G. P. S.
Quote	Protein secondary structure prediction using nearest neighbor and neural network approach
Server	PHDpsi [106]
URL	http://cubic.bioc.columbia.edu/predictprotein
Description	PHD using PSI-BLAST
Authors	Przybylski, D. and Rost, B.
Quote	Alignments grow, secondary structure prediction improves.
Server	JPred [37]
URL	http://www.compbio.dundee.ac.uk/~www-jpred/
Description	A consensus method for protein secondary structure prediction
Authors	Cuff, J. A. and Barton, G. J.
Quote	Jpred: A Consensus Secondary Structure Prediction Server.
Server	PHD [113]
URL	http://cubic.bioc.columbia.edu/predictprotein
Description	Profile network from HeiDelberg
Authors	Rost, B.
Quote	PHD: predicting one-dimensional protein structure by profile based neural networks.
Server	PSIpred [67]
URL	http://bioinf.cs.ucl.ac.uk/psipred/psiform.html
Description	Structure prediction based on position-specific scoring matrices
Authors	Jones, D. T.
Quote	Protein secondary structure prediction based on position-specific scoring matrices.

References

- [1] N. Abe and H. Mamitsuka. Predicting protein secondary structure using stochastic tree grammars. *Machine Learning*, 29:275–301, 1997.
- [2] S. F. Altschul, W. Gish, W. Miller, E. W. Myers, and D. J. Lipman. A basic local alignment search tool. *Journal of Molecular Biology*, 215(3):403–410, 1990.
- [3] S. F. Altschul, T. L. Madden, A. A. Schaffer, J. Zhang, Z. Zhang, W. Miller, and D. J. Lipman. Gapped blast and psi-blast: A new generation of protein database search programs. *Nucleic Acids Research*, 25(17):3389–3402, 1997.
- [4] C. B. Anfinsen. Principles that govern the folding of protein chains. *Science*, 181:223–238, 1973.
- [5] K. Asai, S. Hayamizu, and K. I. Handa. Prediction of protein secondary structure by the hidden markov model. *Computer Applications in the Biosciences*, 9(2):141–146, 1993.
- [6] P. Baldi, S. Brunak, Y. Chauvin, C. A. F. Andersen, and H. Henrik Nielsen. Assessing the sccuracy of prediction algorithms for classification: An overview. *Bioinformatics*, 16(5):412–424, 2000.

REFERENCES

- [7] P. Baldi, S. Brunak, P. Frasconi, G. Pollastri, and G. Soda. *Bidirectional IOHMMs and Recurrent Neural Networks for Protein Secondary Structure Prediction*. CLUEB, Bologna, Italy, 2000.
- [8] P. Baldi, S. Brunak, P. Frasconi, G. Pollastri, and G. Soda. Bidirectional dynamics for protein secondary structure prediction. *Lecture Notes In Computer Science, Sequence Learning: Paradigms, Algorithms, and Applications*, 1828:80–104, 2001.
- [9] P. Baldi, S. Brunak, P. Frasconi, G. Soda, and G. Pollastri. Exploiting the past and the future in protein secondary structure prediction. *Bioinformatics*, 15(11):937–946, 1999.
- [10] P. Baldi and G. Pollastri. Machine learning structural and functional proteomics. *IEEE Intelligent Systems, Special Issue on Intelligent Systems in Biology*, 17(2):28–35, 2002.
- [11] P. Baldi and G. Pollastri. The principled design of large-scale recursive neural network architectures – dag-rnns and the protein structure prediction problem. *Journal of Machine Learning Research*, 4:575–602, 2003.
- [12] P. Baldi, G. Pollastri, P. Frasconi, and A. Vullo. New machine learning methods for the prediction of protein topologies. *Artificial Intelligence and Heuristic Methods in Bioinformatics*, 183:51–74, 2003.
- [13] Y. Bengio, P. Simard, and P. Frasconi. Learning long-term dependencies with gradient descent is difficult. *IEEE Transactions on Neural Networks*, 5(2):157–166, 1994.

REFERENCES

- [14] H. M. Berman, J. Westbrook, Z. Feng, G. Gilliland, T. N. Bhat, H. Weissig, I. N. Shindyalov, and P. E. Bourne. The protein data bank. *Nucleic Acids Research*, 28(1):235–242, 2000.
- [15] V. Biou, J. F. Gibrat, J. M. Levin, B. Robson, and J. Garnier. Secondary structure prediction: Combination of three different methods. *Protein Engineering*, 2(3):185–191, 1988.
- [16] E. Blout, C. De Loz, S. Bloom, and G. Fasman. Dependence of the conformation of synthetic polypeptides on amino acid composition. *Journal of the American Chemical Society*, 82:3787–3789, 1960.
- [17] E. R. Blout. The dependence of the conformation of polypeptides and proteins upon amino acid composition. In M. Stahman, editor, *Polyamino Acids, Polypeptides, and Proteins*, pages 275–279. University of Wisconsin Press, Madison, 1962.
- [18] R. Boné, M. Crucianu, and Jean-Pierre Asselin de Beauville. Learning long-term dependencies by the selective addition of time-delayed connections to recurrent neural networks. *NeuroComputing*, 48(1-4):251–266, 2002.
- [19] R. Bonneau and D. Baker. Ab initio protein structure prediction: Progress and prospects. *Annual Review of Biophysics and Biomolecular Structure*, 30:173–189, 2001.

REFERENCES

- [20] R. Bonneau, J. Tsai, I. Ruczinski, D. Chivian, C. Rohl, C. E. M. Strauss, and D. Baker. Rosetta in casp4: Progress in ab initio protein structure prediction. *Proteins: Structure, Function, and Genetics*, 45(S5):119–126, 2001.
- [21] J. U. Bowie, R. Luthy, and D. Eisenberg. A method to identify protein sequences that fold into a known three-dimensional structure. *Science*, 253:164–170, 1991.
- [22] C. Branden and J. Tooze. *Introduction to Protein Structure*. Garland, 2nd edition, 1999.
- [23] B. Šter. Latched recurrent neural network. *Electrotechnical Review*, 70(1-2):46–51, 2003.
- [24] C. J. C. BURGESS. A tutorial on support vector machines for pattern recognition. *Data Mining and Knowledge Discovery*, 2:121–167, 1998.
- [25] C. Bystroff and D. Baker. Blind predictions of local protein structure in casp2 targets using the i-sites library. *Proteins: Structure, Function, and Genetics, Suppl.*, 1:167–171, 1997.
- [26] C. Bystroff and Y. Shao. Fully automated ab initio protein structure prediction using i-sites, hmmstr and rosetta. *Bioinformatics*, 18(Supplement 1):S54–S61, 2002.
- [27] C. Bystroff, V. Thorsson, and D. Baker. Hmmstr: a hidden markov model for local sequence-structure correlations in proteins. *Journal of Molecular Biology*, 301(1):173–190, 2000.

REFERENCES

- [28] Lemer C., M. J. Rومان, and S. J. Wodak. Protein structure prediction by threading methods: Evaluation of current techniques. *Proteins: Structure, Function and Genetics*, 23:337–355, 1996.
- [29] J. Chen and N. S. Chaudhari. Statistical analysis of long-range interactions in proteins. In *The 2006 International Conference on Bioinformatics & Computational Biology*, pages 296–302, Las Vegas, Nevada, USA, June 26-29 2006.
- [30] S. Y. Cho, Z. Chi, W. C. Siu, and A. C. Tsoi. An improved algorithm for learning long-term dependency problems in adaptive processing of data structures. *IEEE Transactions on Neural Networks*, 14(4):781–793, 2003.
- [31] P. Y. Chou and G. D. Fasman. Prediction of protein conformation. *Biochemistry*, 13(2):222–245, 1974.
- [32] P. Y. Chou and G. D. Fasman. Empirical predictions of protein conformation. *Annual Review of Biochemistry*, 47:251–276, 1978.
- [33] P. Y. Chou and G. D. Fasman. Prediction of the secondary structure of proteins from their amino acid sequence. *Advance in Enzymology and Related Areas of Molecular Biology*, 47:45–48, 1978.
- [34] W. Chu, Z. Ghahramani, and D. L. Wild. A graphical model for protein secondary structure prediction. In *Proceedings of the International Conference on Machine Learning, ICML'04*, pages 161–168, Canada, 2004.

REFERENCES

- [35] G. E. Crooks and S. E. Brenner. Protein secondary structure: Entropy, correlations and prediction. *Bioinformatics*, 20(10):1603–1611, 2004.
- [36] J. A. Cuff and G. J. Barton. Evaluation and improvement of multiple sequence methods for protein secondary structure prediction. *Proteins: Structure, Function and Genetics*, 34:508–519, 1999.
- [37] J. A. Cuff, M. E. Clamp, A. S. Siddiqui, M. Finlay, and G. J. Barton. Jpred: A consensus secondary structure prediction server. *Bioinformatics*, 14:892–893, 1998.
- [38] A. Daniel. Prediction of protein secondary structure using long short-term memory. Master’s thesis, School of Information Science, Computer and Electrical Engineering, Halmstad University, 2003.
- [39] R. O. Day, J. B. Zydallis, G. B. Lamont, and R. Pachter. Solving the protein structure prediction problem through a multiobjective genetic algorithm. In *Nanotech: Technical Proceedings of the 2002 International Conference on Computational Nanoscience and Nanotechnology*, pages 32–35, Cambridge, USA, 2002.
- [40] J.L. Elman. Distributed representations, simple recurrent networks, and grammatical structure. *Machine Learning*, 7:195–226, 1991.
- [41] A. Fiser, Z. Dosztányi, and I. Simon. The role of long-range interactions in defining the secondary structure of proteins is overestimated. *Computer Applications in the Biosciences*, 13(3):297–301, 1997.

REFERENCES

- [42] M. L. Forcada and R. C. Carrasco. Learning the initial state of a second-order recurrent neural network during regular-language inference. *Neural Computation*, 7(5):923–930, 1995.
- [43] V. D. Francesco, J. Garnier, and P. J. Munson. Improving protein secondary structure prediction with aligned homologous sequences. *Protein Science*, 5:106–113, 1996.
- [44] R. W. Frick. Explanations of grouping effects in immediate ordered recall. *Cognition*, 17:551–562, 1989.
- [45] D. Frishman and P. Argos. Knowledge-based protein secondary structure assignment. *Proteins*, 23(4):566–579, 1995.
- [46] D. Frishman and P. Argos. Incorporation of non-local interactions in protein secondary structure prediction from the amino acid sequence. *Protein Engineering*, 9(2):133–142, 1996.
- [47] D. Frishman and P. Argos. Seventy-five percent accuracy in protein secondary structure prediction. *Proteins*, 27(3):329–335, 1997.
- [48] J. Garnier, J. F. Gibrat, and B. Robson. Gor method for predicting protein secondary structure from amino acid sequence. *Methods in Enzymology*, 266:540–553, 1996.
- [49] J. Garnier, D. J. Osguthorpe, and B. Robson. Analysis of the accuracy and implications of simple methods for predicting the secondary structure of globular proteins. *Journal of Molecular Biology*, 120(1):97–120, 1978.

REFERENCES

- [50] J. Garnier and B. Robson. *Prediction of Protein Structure and the Principles of Protein Conformation*, chapter 10, page 417465. New York Plenum Press, 1989.
- [51] F. A. Gers. *Long Short-Term Memory in Recurrent Neural Networks*. PhD thesis, Department of Computer Science, Swiss Federal Institute of Technology, Lausanne, EPEL, Switzerland, 2001.
- [52] J. F. Gibrat, J. Garnier, and B. Robson. Further developments of protein secondary structure prediction using information theory. *Journal of Molecular Biology*, 198(3):425–443, 1987.
- [53] E. M. Gold. Language identification in the limit. *Information and Control*, 10(5):447–474, 1967.
- [54] M. M. Gromiha and S. Selvaraj. Importance of long-range interactions in protein folding. *Biophysical Chemistry*, 77:49–68, 1999.
- [55] M. M. Gromiha and S. Selvaraj. Role of medium- and long-range interactions in discriminating globular and membrane proteins. *International Journal of Biological Macromolecules*, 29:25–34, 2001.
- [56] Y. Guermeur, C. Geourjon, P. Gallinari, and G. Deléage. Improved performance in protein secondary structure prediction by inhomogeneous score combination. *Bioinformatics*, 15(5):413–421, 1999.

REFERENCES

- [57] Y. Guermeur, G. Pollastri, A. Eliseeff, D. Zelus, H. Paugam-Moisy, and P. Baldi. Combining protein secondary structure prediction models with ensemble methods of optimal complexity. *Neurocomputing*, 56:305–327, 2004.
- [58] N. Guex and M. C. Peitsch. Swiss-model and the swiss-pdbviewer: An environment for comparative protein modelling. *Electrophoresis*, 18(15):2714–2723, 1997.
- [59] L. K. Hansen and P. Salamon. Neural network ensembles. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 12(10):993–1001, 1990.
- [60] S. E. Hihi and Y. Bengio. Hierarchical recurrent neural networks for long-term dependencies. In M. Mozer, D. D. Touretzky, and M. Perrone, editors, *Advances in Neural Information Processing Systems*, pages 493–499. MIT Press, 1996.
- [61] G. J. Hitch, N. Burgess, J. N. Towse, and V. Culpin. Temporal grouping effects in immediate recall: A working memory analysis. *Quarterly Journal of Experimental Psychology*, 49A:116–139, 1996.
- [62] S. Hochreiter and J. Schmidhuber. Long short-term memory. *Neural Computation*, 9(8):1735–1780, 1997.
- [63] S. Hua and Z. Sun. A novel method of protein secondary structure prediction with high segment overlap measure: Support vector machine approach. *Journal of Molecular Biology*, 308:397–407, 2001.
- [64] E. G. Hutchinson and J. M. Thornton. A revised set of potentials for beta-turn formation in proteins. *Protein Science*, 3:2207–2216, 1994.

REFERENCES

- [65] K. Ikeda and J. Higo. Free-energy landscape of a chameleon sequence in explicit water and its inherent alpha/beta bifacial property. *Protein Science*, 12(11):2542–2548, 2003.
- [66] I. Jacoboni, P. L. Martelli, P. Fariselli, M. Compiani, and R. Casadio. Predictions of protein segments with the same amino acid sequence and different secondary structure: a benchmark for predictive methods. *Proteins*, 41(4):535–544, 2000.
- [67] D. T. Jones. Protein secondary structure prediction based on position-specific scoring matrices. *Journal of Molecular Biology*, 292(2):195–202, 1999.
- [68] D. T. Jones, W. R. Taylor, and J. M. Thornton. A new approach to protein fold recognition. *Nature*, 358:86–89, 1992.
- [69] W. Kabsch and C. Sander. Dictionary of protein secondary structure: Pattern recognition of hydrogen-bonded and geometrical features. *Biopolymers*, 22:2577–2637, 1983.
- [70] K. Karplus, C. Barrett, and R. Hughey. Hidden markov models for detecting remote protein homologies. *Bioinformatics*, 14(10):846–856, 1998.
- [71] D. Kihara. The effect of long-range interactions on the secondary structure formation of proteins. *Protein Science*, 14:1955–1963, 2005.
- [72] A. R. Kinjo and K. Nishikawa. Recoverable one-dimensional encoding of three-dimensional protein structures. *Bioinformatics*, 21(10):2167–2170, 2005.

REFERENCES

- [73] J. L. Klepeis and C. A. Floudas. Ab initio tertiary structure prediction of proteins. *Journal of Global Optimization*, 25(1):113–140, 2003.
- [74] S. Kobayashi and T. Yokomori. Modeling rna secondary structures using tree grammars. In *Proceedings of the 5th Genome Informatics Workshop*, pages 29–38, 1994.
- [75] K. W. C. Ku, M. W. Mak, and W. C. Siu. A cellular genetic algorithm for training recurrent neural networks. In *Proceedings of International Conference Neural Networks and Signal Processing*, pages 140–143, 1995.
- [76] T. S. Kumarevel, M. M. Gromiha, S. Selvaraj, K. Gayatri, and P. K. R. Kumar. Influence of medium- and long-range interactions in different folding types of globular proteins. *Biophysical Chemistry*, 99(2):189–198, 2002.
- [77] R. H. Lathrop. The protein threading problem with sequence amino acid interaction preference is np-complete. *Protein Engineering*, 7(9):1059–1068, 1994.
- [78] C. Levinthal. How to fold graciously. In P. DeBrunner, J. Tsibris, and E. Munck, editors, *Mossbauer Spectroscopy in Biological Systems*, pages 22–24. University of Illinois Press, Urbana, USA, 1969.
- [79] T. Lin, B. Horne, P. Tino, and L. Giles. Learning long-term dependencies is not as difficult with narx recurrent neural networks. Umiacs-tr-95-78, International Computer Science Institute, University of Maryland, 1995.

REFERENCES

- [80] T. Lin, B. G. Horne, and C. L. Giles. How embedded memory in recurrent neural network architectures helps learning long-term temporal dependencies. *Neural Networks*, 11:861–868, 1998.
- [81] T. Lin, B. G. Horne, P. Tino, and C. L. Giles. Learning long-term dependencies in narx recurrent neural networks. *IEEE Transactions on Neural Networks*, 7(6):1329–1337, 1996.
- [82] K. U. Linderstrom-Lang. *The Lane Medical Lectures-Proteins and Enzymes*. Stanford University Press, Stanford, California, 1952.
- [83] K. U. Linderstrom-Lang and J. A. Shellman. Protein structure and enzyme activity. In P. D. Boyer, editor, *The Enzymes*, volume 1, pages 443–510. Academic Press, New York, 2 edition, 1959.
- [84] S. Ma and C. Ji. Fast training of recurrent networks based on the em algorithms. *IEEE Transactions on Neural Networks*, 9(1):11–26, 1998.
- [85] A. D. MacKerell, D. Bashford, M. Bellott, R. L. Dunbrack, J. D. Evanseck, M. J. Field, S. Fischer, J. Gao, H. Guo, S. Ha, D. Joseph-McCarthy, K. L. Kuchnir, Kuczera, F. T. K. Lau, C. Mattos, S. Michnick, T. Ngo, D. T. Nguyen, B. Prodhom, W. E. Reiher, B. Roux, M. Schlenkrich, J. C. Smith, R. Stote, J. Straub, M. Watanabe, J. Wiórkiewicz-Kuczera, D. Yin, and M. Karplus. All-atom empirical potential for molecular modeling and dynamics studies of proteins. *The Journal of Physical Chemistry B*, 102(18):3586–3616, 1998.

REFERENCES

- [86] M. A. Marti-Renom, A. Stuart, A. Fiser, R. Sánchez, F. Melo, and A. Sali. Comparative protein structure modelling of genes and genomes. *Annual Review of Biophysics and Biomolecular Structure*, 29:291–325, 2000.
- [87] B. Matthews. Comparison of the predicted and observed secondary structure of t4 phage lysozyme. *Biochimica et Biophysica Acta*, 405(2):442–451, 1975.
- [88] K. Müller, S. Mika, G. Ratsch, K. Tsuda, and B. Scholkopf. An introduction to kernel-based learning algorithms. *IEEE Transactions on Neural Networks*, 12(2):181–201, 2001.
- [89] D. L. Minor and P. S. Kim. Context-dependent secondary structure formation of a designed protein sequence. *Nature*, 380:730–734, 1996.
- [90] J. MONOD. On symmetry and function in biological systems. In A. Engström and B. Strandberg, editors, *Symmetry and Function of Biological Systems at the Macromolecular Level*, pages 15–27. John Wiley & Sons Inc, 1969.
- [91] J. Moult, T. Hubbard, S. H. Bryant, K. Fidelis, and J. T. Pedersen. Critical assessment of methods of protein structure prediction(casp): Round ii. *Proteins: Structure, Function and Genetics*, 29:2–6, 1997.
- [92] V. Munoz, P. Cronet, E. Lopez-Hernandez, and L. Serrano. Analysis of the effect of local interactions on protein stability. *Folding and Design*, 1:167–178, 1996.
- [93] K. Nagano and K. Hasegawa. Logical analysis of the mechanism of protein folding. *Journal of Molecular Biology*, 94:257–281, 1975.

REFERENCES

- [94] S. B. Needleman and C. D. Wunsch. A general method applicable to the search for similarities in the amino acid sequence of two proteins. *Journal of Molecular Biology*, 48(3):443–453, 1970.
- [95] C. A. Orengo, J. E. Bray, T. Hubbard, L. LoConte, and I. I. Sillitoe. Analysis and assessment of ab initio three-dimensional prediction, secondary structure, and contacts prediction. *Proteins: Structure, Function and Genetics*, 37:149–170, 1999.
- [96] X. M. Pan, W. D. Niu, and Z. X. Wang. What is the minimum number of residues to determine the secondary structural state. *Journal of Protein Chemistry*, 18(5):579–584, 1999.
- [97] L. Pauling and R. B. Corey. Configurations of polypeptide chains with favoured orientations around single bonds: Two new pleated sheets. *Proceedings of the National Academy of Sciences of the United States of America*, 37:729–740, 1951.
- [98] L. Pauling, R. B. Corey, and H. R. Branson. The structure of proteins: Two hydrogen-bonded helical configurations of the polypeptide chain. *Proceedings of the National Academy of Sciences of the United States of America*, 37:205–234, 1951.
- [99] W. R. Pearson. Rapid and sensitive comparison with fasta and fastp. *Methods in Enzymology*, 183:63–98, 1990.
- [100] W. R. Pearson. Flexible sequence similarity searching with the fasta3 program package. *Methods in Molecular Biology*, 132:185–219, 2000.
- [101] S. Pinker. Formal models of language learning. *Cognition*, 7(3):217–283, 1979.

REFERENCES

- [102] G. Pollastri. *A Machine Learning Approach to Protein Structure Prediction*. PhD thesis, Department of Information and Computer Science, University of California, Irvine, 2003.
- [103] G. Pollastri, P. Baldi, A. Vullo, and P. Frasconi. Prediction of protein topologies using ghoHmms and grnns. In S. Becker, S. Thrun, and K. Obermayer, editors, *Advance in Neural Information Processing Systems 15 (NIPS'02)*, pages 1449–1456, Cambridge, MA, 2002. MIT Press.
- [104] G. Pollastri and A. McLysaght. Porter: A new, accurate server for protein secondary structure prediction. *Bioinformatics*, 21(8):1719–1720, 2005.
- [105] G. Pollastri, D. Przybylski, B. Rost, and P. Baldi. Improving the prediction of protein secondary structure in three and eight classes using recurrent neural networks and profiles. *Proteins*, 47(2):228–235, 2002.
- [106] D. Przybylski and B. Rost. Alignments grow, secondary structure prediction improves. *Proteins*, 46:197–205, 2002.
- [107] O. B. Ptitsyn and M. V. Volkenstein. Protein structures and neutral theory of evolution. *Journal of Biomolecular Structure and Dynamics*, 4:137–156, 1986.
- [108] N. Qian and T. J. Sejnowski. Predicting the secondary structure of globular proteins using neural network models. *Journal of Molecular Biology*, 202(4):865–884, 1988.
- [109] L. R. Rabiner. A tutorial on hidden markov models and selected application in speech recognition. In *Proceedings of the IEEE*, number 2, pages 257–286, 1989.

REFERENCES

- [110] G. P. S. Raghava. Protein secondary structure prediction using nearest neighbor and neural network approach. *CASP4*, pages 75–76, 2000.
- [111] F. M. Richards and C. E. Kundrot. Identification of structural motifs from protein coordinate data: Secondary structure and first-level supersecondary structure. *PROTEINS: Structure, Function, and Genetics*, 3(2):71–84, 1988.
- [112] S. K. Riis and A. Krogh. Improving prediction of protein secondary structure using structural neural networks and multiple sequence alignment. *Journal of Computational Biology*, 3(1):163–183, 1996.
- [113] B. Rost. Phd: Predicting one-dimensional protein structure by profile based neural networks. *Methods in Enzymology*, 266:525–539, 1996.
- [114] B. Rost and C. Sander. Prediction of protein secondary structure at better than 70% accuracy. *Journal of Molecular Biology*, 232(2):584–599, 1993.
- [115] B. Rost and C. Sander. Combining evolutionary information and neural networks to predict protein secondary structure. *Proteins: Structure, Function and Genetics*, 19(1):55–72, 1994.
- [116] B. Rost, C. Sander, and R. Schneider. Redefining the goals of protein secondary structure prediction. *Journal of Molecular Biology*, 235:13–26, 1994.
- [117] Sander C. Rost B. Improved prediction of protein secondary structure by use of sequence profiles and neural networks. *Proceedings of the National Academy of Sciences of the United States of America*, 90(16):7558–7562, 1993.

REFERENCES

- [118] J. Ryan. Temporal grouping, rehearsal and short-term memory. *Quarterly Journal of Experimental Psychology*, 21:148–155, 1969.
- [119] J. A. Sánchez and J. M. Benedí. Learning of stochastic context-free grammars by means of estimation algorithms. In *In European Conference on Speech Communication and Technology*, pages 1799–1802, Budapest, Hungary, 1999.
- [120] R. Sánchez and A. Šali. Advances in comparative protein-structure modelling. *Current Opinion in Structural Biology*, 7, 1997.
- [121] Y. Sakakibara, M. Brown, R. Hughey, I. S. Mian, K. Sjolander, R. C. Underwood, and D. Haussler. Stochastic context-free grammars for trna modeling. *Nucleic Acids Research*, 22:5112–5120, 1994.
- [122] A. A. Salamov and V. V. Solovyev. Prediction of protein secondary structure by combining nearest-neighbor algorithms and multiple sequence alignments. *Journal of Molecular Biology*, 247(1):11–15, 1995.
- [123] A. A. Salamov and V. V. Solovyev. Protein secondary structure prediction using local alignments. *Journal of Molecular Biology*, 268(1):31–36, 1997.
- [124] J. D. Schaffer, D. Whitley, and L. J. Eshelman. Combinations of genetic algorithms and neural networks: A survey of the state of the art. In *Proceedings of the International Workshop on Combinations of Genetic Algorithms and Neural Networks*, pages 1–37, 1992.

REFERENCES

- [125] J. Schmidhuber. Learning complex, extended sequences using the principle of history compression. *Neural Computation*, 4(2):234–242, 1992.
- [126] S. C. Schmidler. *Statistical Models and Monte Carlo Methods for Protein Structure Prediction*. PhD thesis, Stanford University, Department of Medicine, Statistics, & Biochemistry, 2002.
- [127] S. C. Schmidler, J. S. Liu, and D. L. Brutlag. Bayesian segmentation of protein secondary structure. *Journal of Computational Biology*, 7(1-2):233–248, 2000.
- [128] M. Schuster and K. K. Paliwal. Bidirectional recurrent neural networks. *IEEE Transactions on Signal Processing*, 45(11):2673–2681, 1997.
- [129] D. B. Searls. *The Computational Linguistics of Biological Sequences in Artificial Intelligence in Molecular Biology*, chapter 2, pages 47–120. AAAI press, California, USA, 1993.
- [130] D. B. Searls. Linguistic approaches to biological sequences. *Computer Applications in the Biosciences*, 13(4):333–344, 1997.
- [131] D. B. Searls. Formal language theory and biological macromolecules. *Series in Discrete Mathematics and Theoretical Computer Science*, 47:117–140, 1999.
- [132] S. Selvaraj and M. M. Gromiha. Importance of long-range interactions in $(\alpha/\beta)_8$ barrel fold. *Journal of protein chemistry*, 17(7):691–697, 1998.
- [133] T. Z. Sen, R. L. Jernigan, J. Garnier, and A. Kloczkowski. GOr v server for protein secondary structure prediction. *Bioinformatics*, 21(11):2787–2788, 2005.

REFERENCES

- [134] F. T. Severin and M. K. Rigby. Influence of digit grouping on memory for telephone numbers. *Journal of Applied Psychology*, 47:117–119, 1963.
- [135] J. Skolnick and A. Kolinski. Simulations of the folding of a globular protein. *Science*, 250:1121–1125, 1990.
- [136] T. F. Smith and M. S. Waterman. Identification of common molecular subsequences. *Journal of Molecular Biology*, 147(1):195–197, 1981.
- [137] P. Stolorz, A. Lapedes, and Y. Xia. Predicting protein secondary structure using neural net and statistical methods. *Journal of Molecular Biology*, 225(2):363–377, 1992.
- [138] C. M. Stultz, J. V. White, and T. F. Smith. Structural analysis based on state-space modeling. *Protein Science*, 2(3):305–314, 1993.
- [139] A. G. Szent-Gyrgyi and C. Cohen. Role of proline in polypeptide chain configuration of proteins. *Science*, 126(3276):697–698, 1957.
- [140] W. R. Taylor. Pattern matching methods in protein sequence comparison and structure prediction. *Protein Engineering*, 2:77–86, 1988.
- [141] Y. Uemura, A. Hasegawa, S. Kobayashi, and T. Yokomori. Tree adjoining grammars for rna structure prediction. *Theoretical Computer Science*, 210:277–303, 1999.
- [142] J. J. Ward, L. J. McGuffin, B. F. Buxton, and D. T. Jones. Secondary structure prediction with support vector machines. *Bioinformatics*, 19(13):1650–1655, 2003.

REFERENCES

- [143] O. Weiss, M. A. Jimenez-Montano, and H. Herzel. Information content of protein sequences. *Journal of theoretical biology*, 206(3):379–386, 2000.
- [144] P. J. Werbos. Backpropagation through time: What it does and how to do it. In *Proceedings of the IEEE*, volume 78, pages 1550–1560, 1990.
- [145] W. A. Wickelgren. Rehearsal grouping and hierarchical organization of serial position cues in short-term memory. *Quarterly Journal of Experimental Psychology*, 19:97–102, 1967.
- [146] R. J. Williams and Zipser. Gradient based learning algorithms for recurrent connectionist networks. Technical report, nu-ccs-90-9, Northeastern University, College of Computer Science, 1990.
- [147] Y. Xu and D. Xu. Protein threading using prospect: Design and evaluation. *Proteins: Structure, Function, and Genetics*, 40(3):343–354, 2000.
- [148] T. M. Yi and E. S. Lander. Protein secondary structure prediction using nearest-neighbor methods. *Journal of Molecular Biology*, 232:1117–1129, 1993.
- [149] A. Zemla, C. Venclovas, K. Fidelis, and B. Rost. A modified definition of sov, a segment-based measure for protein secondary structure prediction assessment. *Proteins: Structure, Function, and Genetics*, 34:220–223, 1999.
- [150] X. Zhang, J. P. Mesirov, and D. L. Waltz. Hybrid system for protein secondary structure prediction. *Journal of Molecular Biology*, 225(4):1049–1063, 1992.

REFERENCES

- [151] X. Zhou, F. Alber, G. Folkers, G. H. Gonnet, and G. Chelvanayagam. An analysis of the helix-to-strand transition between peptides with identical sequence. *Proteins*, 41(2):248–256, 2000.