

# Geometry guided supervised representation learning for classification

Li, Yue

2020

Li, Y. (2020). Geometry guided supervised representation learning for classification.  
Doctoral thesis, Nanyang Technological University, Singapore.

<https://hdl.handle.net/10356/137528>

<https://doi.org/10.32657/10356/137528>

---

This work is licensed under a Creative Commons Attribution-NonCommercial 4.0  
International License (CC BY-NC 4.0).

*Downloaded on 13 Mar 2024 17:28:41 SGT*

---

# GEOMETRY GUIDED SUPERVISED REPRESENTATION LEARNING FOR CLASSIFICATION

---



LI YUE

School of Electrical & Electronic Engineering

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

2020



## Statement of Originality

I hereby certify that the work embodied in this thesis is the result of original research, is free of plagiarised materials, and has not been submitted for a higher degree to any other University or Institution.

27 Feb. 2020

.....

Date



.....

LI YUE





## Supervisor Declaration Statement

I have reviewed the content and presentation style of this thesis and declare it is free of plagiarism and of sufficient grammatical clarity to be examined. To the best of my knowledge, the research and writing are those of the candidate except as acknowledged in the Author Attribution Statement. I confirm that the investigations were conducted in accord with the ethics policies and integrity standards of Nanyang Technological University and that the research data are presented honestly and without prejudice.

27 Feb. 2020

.....

Date



.....

Prof. Huang Guang-Bin



## Authorship Attribution Statement

This thesis contains material from 3 paper(s) published in the following peer-reviewed journal(s) in which I am listed as an author.

Chapter 3 is published as Yue Li, Chamara Kasun Liyanaarachchi Lekamalage, Tianchi Liu, Pinan Chen, Guang-Bin Huang, “Learning representations with local and global geometries preserved for machine fault diagnosis”, *IEEE Transactions on Industrial Electronics*, 67(3):2360-70, 2019.

The contributions of the co-authors are as follows:

- Prof G.B. Huang provided the initial project direction and edited the manuscript drafts.
- I prepared the manuscript drafts. The manuscript was revised by Dr T. Liu and Dr. C. K. Liyanaarachchi Lekamalage.
- I co-designed the study with Prof G.B. Huang and performed all the laboratory work at EEE-Delta Joint Lab on Internet of Things.
- I also designed the algorithms and analyzed experimental results. Dr. C. K. Liyanaarachchi Lekamalage helped to provide the source code.
- Dr P. Chen assisted to provide the dataset.

Chapter 4 is published as Yue Li, Yijie Zeng, Yuanyuan Qing, Guang-Bin Huang, “Learning Local Discriminative Representations via Extreme Learning Machine for Machine Fault Diagnosis”, *Neurocomputing*, minor revision.

The contributions of the co-authors are as follows:

- Prof G.B. Huang provided the initial project direction and edited the manuscript drafts.
- I wrote the drafts of the manuscript. The manuscript was revised together with Mr. Y. Zeng and Ms. Y. Qing.
- I designed the algorithms, performed all the laboratory works, and analyzed experimental results.

Chapter 5 is published as Yue Li, Yijie Zeng, Tianchi Liu, Xiaofan Jia, and Guang-Bin Huang, "Simultaneously learning affinity matrix and data representations for machine fault diagnosis", *Neural Networks*, 122:395-406, 2020.

The contributions of the co-authors are as follows:

- Prof G.B. Huang provided the initial project direction and edited the manuscript drafts.
- I wrote the drafts of the manuscript. The manuscript was revised together with Mr. Y. Zeng, Dr. T. Liu and Ms. X. Jia.
- I designed the algorithms, performed all the laboratory works, and analyzed experimental results.

27 Feb. 2020

.....

Date



.....

LI YUE

# Acknowledgements

I would like to express my greatest gratitude to my supervisor Professor Huang Guang-Bin, for his support and excellent supervision. He is a wise and knowledgeable guider who offered me many inspiring suggestions on my research topic and thesis. Not only the academic support, he also provided personal guidance to help me build the right moral standard.

Great thanks to Dr. Liu Tianchi, who guided me as a senior on my research topic and academic writing. I am also thankful to Dr. Liyanaarachchi Lekamalage Chamara Kasun for the source code and thoughtful discussions. Special thanks to my friends Dr. Cui Dongshun, Mr. Han Wei, Mr. Song Kang, and Mr. Zeng Yijie when I was in Delta-EEE Joint Research Lab.

I would like to thank the technical and financial support from Research Staff Office and Delta-EEE Joint Research Lab in the School of Electrical and Electronic Engineering, Nanyang Technological University.

Last but not least, I would like to thank my parents and my wife Sumei for their love, encouragements, and support during my Ph.D. studies.



# Contents

<b>Acknowledgements</b>	<b>v</b>
<b>Summary</b>	<b>vii</b>
<b>List of Figures</b>	<b>ix</b>
<b>List of Tables</b>	<b>xi</b>
<b>Symbols and Acronyms</b>	<b>xii</b>
<b>1 Introduction</b>	<b>1</b>
1.1 Background and Motivation . . . . .	2
1.2 Objectives and Contributions . . . . .	4
1.3 Structure of Thesis . . . . .	6
<b>2 Literature Review</b>	<b>8</b>
2.1 Representation Learning Algorithms . . . . .	9
2.1.1 Global Geometrically Motivated Algorithms . . . . .	9
2.1.1.1 Principal Component Analysis . . . . .	9
2.1.1.2 Linear Discriminant Analysis . . . . .	10
2.1.2 Local Geometrically Motivated Algorithms . . . . .	11
2.1.2.1 Locally Linear Embedding . . . . .	11
2.1.2.2 Laplacian Eigenmaps . . . . .	12
2.1.2.3 Locality Preserving Projection . . . . .	13
2.1.3 General Graph Embedding . . . . .	14
2.1.4 Auto-encoders . . . . .	16
2.1.4.1 Denoising Auto-Encoder (DAE) . . . . .	17
2.1.4.2 Stacked Auto-encoder (SAE) . . . . .	18
2.2 Extreme Learning Machines . . . . .	20
2.2.1 Regularized Extreme Learning Machine . . . . .	22
2.2.2 Extreme Learning Machine Auto-encoder . . . . .	23
2.2.3 Semi-supervised Extreme Learning Machine . . . . .	25
2.2.4 Unsupervised Extreme Learning Machine . . . . .	26
2.2.5 Generalized Extreme Learning Machine Auto-encoder . . . . .	26



2.2.6	Extreme Learning Machine with Constrained Laplacian Rank	27
2.3	Machine Fault Diagnosis	29
2.4	Summary	34
<b>3</b>	<b>Learning Representations with Local and Global Geometries Preserved</b>	<b>35</b>
3.1	Background and Motivation	36
3.2	Proposed Method	38
3.2.1	Cost Functions	38
3.2.1.1	Reconstruction Cost	38
3.2.1.2	Local Geometry Preserving Cost	39
3.2.1.3	Global Geometry Preserving Cost	39
3.2.1.4	Discrimination Cost	41
3.2.2	Fast Autoencoder with The Local and Global Penalties	42
3.3	Experiments	45
3.3.1	Datasets Descriptions	45
3.3.2	Experimental Setups	48
3.3.3	Evaluation of Local and Global Penalties	50
3.3.4	Evaluation of Computational Efficiency	52
3.3.5	Evaluation of Noise Robustness	57
3.3.6	Comparison with Related Methods	57
3.4	Summary	59
<b>4</b>	<b>Learning Local Discriminative Representations via Extreme Learning Machine</b>	<b>60</b>
4.1	Background and Motivation	61
4.2	Proposed Method	63
4.2.1	Local discriminant preserving extreme learning machine autoencoder	63
4.2.2	Multilayer local discriminant preserving extreme learning machine autoencoder	69
4.3	Experiments	71
4.3.1	Datasets Descriptions and Experiment Setups	71
4.3.2	Comparison with the related methods	73
4.3.3	Application on the machine fault diagnosis	80
	Selection of the number of hidden numbers	81
	Selection of the trade-off parameters	81
	Visualization of the representations	81
	Comparison with state-of-the-art algorithms	84
	Computational complexity	85
4.4	Summary	86
<b>5</b>	<b>Simultaneously Learning Affinity Matrix and Data Representations</b>	<b>87</b>

5.1	Background and Motivation . . . . .	88
5.2	Proposed Method . . . . .	90
5.2.1	Objective function . . . . .	90
	LELMAE-AN . . . . .	90
	Affinity matrix learning . . . . .	92
	Representation learning . . . . .	92
5.2.2	Optimization . . . . .	93
	Update affinity matrix $\mathbf{S}$ . . . . .	93
	Update output matrix $\beta$ . . . . .	96
	Complete training process . . . . .	97
5.3	Experiments . . . . .	99
5.3.1	Datasets Descriptions . . . . .	99
5.3.2	Experimental settings . . . . .	99
5.3.3	The convergence and effect of the affinity matrix . . . . .	101
	Convergence . . . . .	101
	Comparison between the learned and manually constructed affinity matrix . . . . .	101
5.3.4	Experimental results on the UCI and image datasets . . . . .	104
	UCI datasets . . . . .	104
5.3.5	Experimental results on COIL20 dataset . . . . .	106
5.3.6	Experimental results on USPST dataset . . . . .	107
5.3.7	Experimental results on CWRU dataset . . . . .	107
5.3.7.1	Sensitivity analysis of hyper-parameters . . . . .	108
	Number of hidden neurons $l$ . . . . .	108
	Number of neighbors $k$ . . . . .	108
	Trade-off parameters $C$ , $\alpha_1$ , $\alpha_2$ and $\alpha_3$ . . . . .	108
5.3.7.2	Soft and hard discrimination constraint . . . . .	110
5.3.7.3	Data number for training . . . . .	110
5.3.7.4	Comparison with state-of-the-art methods . . . . .	113
5.4	Summary . . . . .	114
<b>6</b>	<b>Conclusions and Future Works</b> . . . . .	<b>115</b>
6.1	Conclusions . . . . .	116
6.2	Future Works . . . . .	118
	<b>List of Author's Publications</b> . . . . .	<b>120</b>
	<b>Bibliography</b> . . . . .	<b>122</b>



# Summary

Machine learning is an essential part of artificial intelligence and a useful tool for data mining. Machine learning algorithms learn a mathematical model from the training dataset and use the model to make predictions on the test dataset without using the explicitly programming. The performances of machine learning algorithms highly depend on the quality of input features, i.e., the redundant information contained in input data may affect the performance and generalization capability of machine learning algorithms. Therefore, it is necessary to remove the unwanted information and retain the relevant information from input data before applying it in machine learning algorithms.

Representation learning algorithms remove the redundant information and extract the useful features from input data automatically. For example, the auto-encoder (AE) retains the relevant information from input data by forcing the embedded representations to reconstruct original input data. Additionally, the extreme learning machine (ELM) was recently extended to representation learning based on the structure of AE. Different from feature extraction algorithms, representation learning algorithms do not require domain knowledge and can reduce human labor. However, as an important property of data representations, the geometry information has not well exploited in existing representation learning algorithms. Therefore, this thesis investigates geometry information discovering and preserving in the representation learning algorithm and applies it to machine fault diagnosis application.

Firstly, I exploits the local and global geometry preserving in data representations. Specifically, the thesis proposes a representation learning algorithm, which is named as the Fast Auto-encoder with the Local and Global Penalties (FAE-LG). The proposed algorithm can efficiently learn discriminative representations with the local and global geometry of input data preserved. FAE-LG uses two cost functions to preserve the local and global geometry of input data, and another cost function to force the learned data representations to reconstruct the original input data. In

the thesis, I practically proves the importance of preserving both local and global geometry in data representations, and theoretically proves that minimizing the difference between random projected data and the representations can preserve the global geometry of input data. Moreover, the proposed algorithm contains a discrimination cost function based on the label information. Hence, it can use a one-step training process and reduces training time significantly. The discrimination cost also reduces the number of neurons required in hidden layers and decreases the test time. Experimental studies on the benchmark dataset demonstrate that FAE-LG is an efficient tool for machine fault diagnosis.

Secondly, the thesis proposes an algorithm that improves the training efficiency of representation learning algorithms and studies the local geometry and local discriminant information exploiting of input data. The previous study proved the importance of preserving geometry information in data representations. However, the AE-based representation learning method, FAE-LG, is trained iteratively by using back-propagation (BP) that requires a significant amount of training time. The extreme learning machine auto-encoder (ELM-AE) is an extension of ELM, which is well-known for its fast training speed and strong generalization ability. Based on ELM-AE, a new algorithm named as the Local Discriminant Preserving Extreme Learning Machine Auto-encoder (LDELM-AE) is proposed. LDELM-AE can learn data representations with the local geometry and local discriminant of input data exploited. Specifically, LDELM-AE incorporates a graph-based penalty in ELM-AE to enhance the within-class compactness and between-class separability of data representations. In the representation space, the local geometry of input data is preserved by minimizing the within-class compactness, which is achieved by mapping the closed data points from the same class to similar representations. Also, the local discriminant information is extracted by maximizing the distances between the margin points and their neighbors in different classes, where the margin points are the data points located at the border of each class. The experimental results demonstrate that LDELM-AE outperforms other related algorithms on several benchmark datasets, and the empirical study also shows it is an efficient tool on machine fault diagnosis.

Finally, the thesis studies the adaptable affinity matrix in representations learning algorithms. The previously proposed algorithms, i.e., FAE-LG and LDELM-AE, proved that preserving geometry information of input data in data representations

can improve the performance of classification tasks. However, these algorithms require to predefine the affinity matrix, which is used to preserve geometry information in data representations. One of the limitations of existing algorithms is the affinity matrix may not able to determine the real relationships between data points precisely, since it is learned under the assumption of a fixed and assumed prior knowledge. Also, learning affinity matrix and data representations in two separated steps may not be optimal and universal for data classification tasks. To overcome the limitations, a novel method, which is named as the Locality-preserving Extreme Learning Machine Auto-encoder with Adaptive Neighbors (LELMAE-AN), is proposed in this thesis to learn the data representations and the affinity matrix simultaneously. Instead of predefining and fixing the affinity matrix, the proposed algorithm adjusts the similarities by taking into account the capability of capturing the geometry information in both original data space and non-linearly mapped representation space. Meanwhile, the geometry information of original data can be preserved in the embedded representations with the help of the affinity matrix. Experimental results on several benchmark datasets demonstrate the effectiveness of the proposed algorithm, and the empirical study also shows it is an efficient tool on machine fault diagnosis.



# List of Figures

2.1	Architecture of AE. . . . .	17
2.2	Architecture of SAE. . . . .	18
2.3	Architecture of ELM. . . . .	20
2.4	The architecture of ELM-AE. . . . .	23
2.5	The structure of rolling bearing. . . . .	29
2.6	The structure of induction motor. . . . .	30
2.7	Vibration signal collected by accelerometer with 12kHz rate. . . . .	33
3.1	The test rig of CWRU bearing dataset. . . . .	46
3.2	Run-to-failure vibration signal with outer race failure in IMS bearing dataset. . . . .	47
3.3	Scatter plots of the learned data representations of the CWRU bearing dataset by using t-SNE. (a) Original data. (b) SFAE. . . . .	51
3.4	Scatter plots of the learned data representations of the CWRU bearing dataset by using t-SNE. (a) SFAE-L. (b) SFAE-G. . . . .	53
3.5	Scatter plots of the learned data representations of the CWRU bearing dataset by using t-SNE. (a) and (b) show representations learned by SFAE-LG in different angles. . . . .	54
3.6	The classification accuracies of SAE, SDA and SFAE-LG with different SNRs (in dB). . . . .	56
4.1	The black data points belong to class 1 and white data points belong to class 2. The left part shows the within-class compactness that is measured by using the data points and their neighbors from the same class. The right part shows the between-class separability that is measure by using the margin data points and their neighbors from the different class. . . . .	64
4.2	Architecture of ML-LDELM. . . . .	65
4.3	Average accuracy of different number of hidden layers for ELM-based algorithms on: (a) IRIS dataset and (b) WINE dataset. . . . .	76
4.4	Average accuracy of different number of hidden layers for ELM-based algorithms on: (a) LIVER dataset and (b) SATIMAGE dataset. . . . .	77
4.5	Average accuracy of different number of hidden layers for ELM-based algorithms on: (a) COIL20 dataset and (b) USPST dataset. . . . .	78
4.6	Machine fault diagnosis performances using different number of hidden neurons. . . . .	80



4.7	Machine fault diagnosis performances using different combinations of trade-off parameters. (a) shows the diagnosis accuracies with different combinations of $\lambda$ and $\gamma$ . (b) shows the diagnosis accuracies with different combinations of $\lambda$ and $C$ . . . . .	82
4.8	Scatter plots of the original data points and learned data representations of the CWRU bearing dataset. . . . .	83
5.1	The value changes in $\ \mathbf{S}\ _2^2$ with iterations. . . . .	102
5.2	Comparison between the learned and manually constructed affinity matrix. . . . .	103
5.3	Sample images from the image datasets. . . . .	104
5.4	Vibration signal collected in 10 seconds. . . . .	106
5.5	CWRU dataset motor diagnosis results using various numbers of hidden neurons. . . . .	107
5.6	CWRU dataset motor diagnosis results using various values of hyperparameters $k$ and $C$ . . . . .	109
5.7	CWRU dataset motor diagnosis results using various values of hyperparameters $\alpha_1$ , $\alpha_2$ and $\alpha_3$ . . . . .	111
5.8	CWRU dataset motor diagnosis results using the soft and hard discrimination constraint in LELMAE-AN. . . . .	112
5.9	CWRU dataset motor diagnosis results using various percentages of samples as the training data. . . . .	112

# List of Tables

2.1	Localized faults of bearing. . . . .	31
2.2	Measurement methods for bearing fault diagnosis. . . . .	33
3.1	Descriptions of bearing conditions in CWRU dataset. . . . .	46
3.2	Number of samples in each class in IMS bearing dataset. . . . .	47
3.3	Hyper-parameters selection range for cross-validation. . . . .	47
3.4	Hyper-parameters of SFAE-LG model. . . . .	49
3.5	Evaluation of local and global geometries for SFAE-LG with four hidden layers. . . . .	50
3.6	The comparison of hidden neurons used by AE-based methods. . . . .	55
3.7	The comparison of training time on CWRU bearing dataset. . . . .	56
3.8	The comparison of classification accuracy ( $Acc$ ), precision ( $Pre$ ), recall ( $Rec$ ) and f-score ( $f$ ) on CWRU bearing dataset. . . . .	57
3.9	The comparison of classification accuracy ( $Acc$ ), precision ( $Pre$ ), recall ( $Rec$ ) and f-score ( $f$ ) on IMS bearing dataset. . . . .	57
3.10	The comparison of classification accuracy among published perfor- mances. . . . .	58
4.1	Descriptions of benchmark datasets. . . . .	71
4.2	The comparison of classification accuracies for each algorithm with one hidden layer. . . . .	74
4.3	The comparison of classification accuracies for each algorithm with two hidden layers. . . . .	75
4.4	The comparison of classification accuracies for each algorithm with three hidden layers. . . . .	79
4.5	The comparison of classification accuracies on CWRU bearing dataset. . . . .	84
4.6	Comparison of training and testing time on CWRU dataset. . . . .	85
5.1	Hyper-parameters selection range for cross-validation. . . . .	101
5.2	The comparison of classification accuracies on UCI and image datasets. . . . .	105
5.3	The comparison of classification accuracies on CWRU bearing dataset. . . . .	113



# Symbols and Acronyms

## Symbols

$d$	Dimension of each sample in original space
$l$	Dimension of each sample in embedded space
$m$	Number of classes
$k$	Number of neighbors
$n$	Number of samples
$\mathcal{N}_i$	The index set of the neighbors of agent $i$
$\mathbf{x}_i \in \mathbb{R}^{d \times 1}$	The $i$ -th sample
$\mathbf{x}'_i \in \mathbb{R}^{d \times 1}$	The corrupted input data $\mathbf{x}_i$
$\hat{\mathbf{x}}_i \in \mathbb{R}^{d \times 1}$	The reconstructed data of $i$ -th sample
$\mathbf{y}_i \in \mathbb{R}^{m \times 1}$	The corresponding one-hot label of the $i$ -th sample
$\mathbf{h}_i \in \mathbb{R}^{l \times 1}$	The output of the hidden layer with respect to the $i$ -th sample
$\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_N] \in \mathbb{R}^{d \times N}$	Data matrix
$\mathbf{X}_{proj} \in \mathbb{R}^{l \times N}$	The embedded data representations
$\mathbf{Y} = [\mathbf{y}_1, \mathbf{y}_2, \dots, \mathbf{y}_N] \in \mathbb{R}^{m \times N}$	Label matrix
$\mathbf{H} = [\mathbf{h}_1, \mathbf{h}_2, \dots, \mathbf{h}_N] \in \mathbb{R}^{l \times N}$	Output matrix of the hidden layer
$\mathbf{W} \in \mathbb{R}^{d \times l}$	The projection matrix
$\boldsymbol{\beta} \in \mathbb{R}^{d \times l}$	Output weights between the hidden layer and output layer
$\mathbf{A} \in \mathbb{R}^{l \times d}$	Input weights between the input layer and the hidden layer
$\mathbf{S} \in \mathbb{R}^{N \times N}$	The affinity matrix
$\mathbf{L} \in \mathbb{R}^{N \times N}$	The graph Laplacian
$\mathbf{I}$	The identity matrix

# Acronyms

ACC	Accuracy
AE	Auto-Encoder
BP	Back-Propagation
CAN	Clustering with Adaptive Neighbors
CNN	Convolutional Neural Network
CWRU	Case Western Reserve University
DAE	Denoising Auto-Encoder
DBN	Deep Belief Networks
DL	Deep Learning
DIR	Degraded Inner Raceway
DOR	Degraded Outer Raceway
DR	Degraded Roller
DWT	Discrete Wavelet Transform
EEMD	Ensemble Empirical Mode Decomposition
ELM	Extreme Learning Machine
ELM-AE	Extreme Learning Machine Auto-Encoder
ELM-CLR	Extreme Learning Machine with Constrained Laplacian Rank
ELMNet	Extreme Learning Machine Network
EPRI	Electric Power Research Institute
FAE-LG	Fast Auto-encoder with the Local and Global penalties
GDELM-AE	Graph Embedded Denoising Extreme Learning Machine
GELM-AE	Generalized Extreme Learning Machine
IRF	Inner Raceway Fault
LDA	Linear Discriminant Analysis
LDELM-AE	Local Discriminant Preserving Extreme Learning Machine Auto-Encoder
LE	Laplacian Eigenmaps
LELMAE-AN	Locality-preserving Extreme Learning Machine Auto-Encoder with Adaptive Neighbors
LFDA	Local Fisher Discrimination Analysis
LLE	Local Linear Embedding
LPP	Locality Preserving Projection

---

MFA	Marginal Fisher Analysis
ML-ELM	Multilayer Extreme Learning Machine
ML-GELM	Multilayer Generalized Extreme Learning Machine
ML-LDELM	Multilayer Local Discriminant Preserving Extreme Learning Machine Auto-Encoder
N	Normal condition
ORF	Outer Raceway Fault
PCA	Principal Component Analysis
PCAN	Projected Clustering with Adaptive Neighbors
PdM	Predictive Maintenance
RBM	Restricted Boltzmann Machine
RF	Roller Fault
SAE	Stacked Auto-Encoder
SFAE-LG	Stacked Fast Auto-Encoder with the Local and Global penalties
SGD-ELM	Stacked Graph Embedded Denoising Extreme Learning Machine
SLFN	Single Layer Feed-forward Neural-network
SNR	Signal-to-Noise Ratio
SS-ELM	Semi-Supervised Extreme Learning Machine
SVD	Singular Value Decomposition
SVM	Support Vector Machine
t-SNE	t-Distributed Stochastic Neighbor Embedding
US-ELM	Unsupervised Extreme Learning Machine



# Chapter 1

## Introduction

*Chapter 1 introduces the background and motivations of studying representation learning and its application to machine fault diagnosis. This chapter also defines research objectives and highlights contributions of this thesis. The structure of this thesis is presented at the end of this chapter.*



## 1.1 Background and Motivation

Machine learning is an essential part of artificial intelligence and a useful tool for data mining, it aims to identify patterns and make decisions with minimum human intervention [1]. The performances of machine learning tasks, such as the accuracies of classification, are affected by noises and redundant information contained in raw data, as noises may cause overlapping classes [2]. Traditionally, features are manually extracted from raw data to remove redundant information and retain the relevant information. For example, in machine fault diagnosis, B. Samanta et al. [3] extracted statistical features, e.g., root mean square, kurtosis, and skewness, from time-domain sensory data. The machine health conditions are then monitored by using the extracted features and an artificial neural network-based classifier. Moreover, F. Al-Badour et al. [4] used the wavelet transform to extract features from the time-frequency domain of original data, and J.-H. Zhong et al. [5] used intrinsic mode functions, which are extracted by using empirical mode decomposition, as the features for further machine health conditions monitoring. However, manual feature extraction is time-consuming, labor-intensive, and requires expert domain knowledge [6]. Hence, the growing interest of machine learning focuses on how to extract features with lesser human labor and time cost.

Representation learning algorithms are developed to overcome the weakness of manual feature extraction since they are designed to learn data representations automatically from raw data [7]. For example, Convolutional Neural Network (CNN) [8] applies multiple learnable filters on the data points to learn data representations. Restricted Boltzmann Machine (RBM) [9] learns representations by approximating the probability distribution of the original inputs. Hierarchical representations can be obtained by stacking RBMs to form a deep structure, which is known as Deep Belief Networks (DBN) [10]. Auto-encoder (AE) [11] learns data representations by reconstructing the original data points from its embedded representation. Similar to DBN, the deep structure, Stacked AE (SAE), can be formed by stacking multiple AEs to obtain hierarchical representations [12]. Furthermore, to improve the training efficiency, Extreme Learning Machine Auto-encoder (ELM-AE) [13, 14] is proposed to learn data representations without iterative training process. Representation learning algorithms are successfully applied in many applications such as natural language processing [15], computer vision [16] and speech recognition [17]. Recently, representation learning algorithms are

employed in machine fault diagnosis tasks to reduce human labor and time cost on features extraction [18–21].

As for the aspect of applications, machine fault diagnosis, which is a fundamental step of predictive maintenance (PdM) [22], aims to monitor machine health conditions from collected sensor data, e.g., using vibration data collected by accelerometers to determine the health condition of motors [23]. As reported by McKinsey global institute, predictive maintenance (PdM) techniques can help factories reduce up to 40% maintenance cost, which is about \$630 billion per year [24]. The machine condition monitoring system is designed to collect real-time data and inspect the health conditions of machines. Generally, the system acquires a large amount of data, and the data is collected much faster than it is analyzed manually by diagnosticians. Therefore, a fault diagnosis system with the ability to adapt various applications and operations conditions with less expert knowledge requirement would be highly desired. This thesis investigates how representation learning algorithms can be applied to machine fault diagnosis. In particular, it considers representation learning algorithms that can learn data representations efficiently with data geometry preserved.

Recently, preserving geometry information of original data points is proved to be an important property of data representations, since the geometry structure needs to be unified in both original and representation space. For example, local geometry represents the structure among partial data points. While learning representations of data points from Swiss roll space, it is necessary to preserve the local geometry from the original space to representation space. Otherwise, the data points from different classes will be put into the same class in the representation space. This learning process is well known as manifold learning. In another hand, global geometry represents the relationship among the whole dataset. The global geometry discovers discriminative information from original data space to representation space. While preserving local geometry minimizes the intraclass compactness, preserving global geometry help to keep the interclass separation.

Local geometry is preserved by retaining the same relationship between a data point and its neighbors before and after representation learning. For example, local linear embedding (LLE) [25] first reconstructs each data point from a linear combination of its neighbors and then learns representations that have the same

linear relationship among data points. Laplacian eigenmaps (LE) [26] is a graph-based method that preserves the distances between data points and their neighbors while mapping them from the input data space to the representation space. Global geometry is preserved by retaining the same relationship among all data points before and after the representation learning. For example, linear discriminant analysis (LDA) [27] rotates the axes to maximize the between-class variance and minimize the within-class variance. Isomap [28] retains the geodesic distances among all data points during the learning process.

To improve the efficiency of representation learning algorithms, extreme learning machine (ELM) based algorithms have been proposed. ELM introduced by Huang et. al [29–38] is a single layer feed-forward neural-network (SLFN) with randomly generated and fixed hidden neurons. ELM is well known for its efficient learning procedure and excellent generalization capability. As an extension of ELM, ELM-AE efficiently learns data representations from input data by utilizing the advantages of random hidden neurons of ELM. Inspired by deep learning structure, multiple ELM-AEs can be stacked together to learn hierarchical representations. D. Cui et al. [39] stacked multiple ELM-AEs to build an extreme learning machine network (ELMNet) and achieved promising performance on the handwritten dataset.

## 1.2 Objectives and Contributions

The objectives of this thesis are defined as follows:

- To efficiently learn data representations that can improve the performance of supervised classification with an application for bearing fault diagnosis.
- To exploit and preserve geometry information of input data while learning data representations.

The following part then describes the contributions of this thesis to achieve the above objectives.

The first study proposes a representation learning algorithm to learn discriminative representations with the local and global geometry of input data preserved. The

proposed algorithm is named as the Fast Auto-encoder with the Local and Global Penalties (FAE-LG). Moreover, the study proposes a stacked FAE-LGs that used to learn hierarchical representations. The representations learned by SFAE-LG can be used by a classifier to detect machine faults. This study theoretically proves the global geometry of input data can be preserved by minimizing the difference between the representations and the random projected input data points. Also, the experiment results show that the non-linear activation can approximately preserve the global geometry of input data. The study also experimentally demonstrates that it is important to preserve both local and global geometry for representation learning in machine fault diagnosis. The proposed algorithm requires fewer hidden neurons in each layer and thus has less computational complexity compared with SAEs. Furthermore, SFAE-LG can be trained more efficiently because it does not require the additional fine-tuning step as existing SAEs. This work has been published in a journal, and the details are as follows:

Yue Li, Chamara Kasun Liyanaarachchi Lekamalage, Tianchi Liu, Pinan Chen, Guang-Bin Huang, “Learning representations with local and global geometries preserved for machine fault diagnosis”, *IEEE Transactions on Industrial Electronics*, 67(3):2360-70, 2019.

The second study proposes an ELM-based representation learning algorithm, which is named as the Local Discriminant Preserving Extreme Learning Machine Auto-encoder (LDELMAE). By utilizing the advantages of ELM, the proposed algorithm learns representations with an excellent generalization capability and faster training speed than other algorithms. Moreover, the learned representations preserve the local geometry and exploit the local discrimination information from input data. The study experimentally demonstrates that LDELMAE learns data representations with within-class compactness and between-class separability maximized. This work has been submitted in a journal, and the details are as follows:

Yue Li, Yijie Zeng, Yuanyuan Qing, Guang-Bin Huang, “Learning Local Discriminative Representations via Extreme Learning Machine for Machine Fault Diagnosis”, *Neurocomputing*, minor revision.

The third study proposes a representation learning algorithm, which is named as the Locality-preserving Extreme Learning Machine Auto-encoder with Adaptive

Neighbors (LELMAE-AN). According to the first and second studies, the geometry information is exploited and preserved based on a predefined affinity matrix. This study proposes a unified objective function that can concurrently learn data representations and the affinity matrix. The affinity matrix is learned by utilizing the geometry information of both original data points and their representations. The assumption is that the pairwise data points with a smaller distance in both original data space and embedded representation space should have a higher similarity; thus, the geometry structure of original data can be preserved in representations. Moreover, this study proposes a soft discriminate constraint, which is optimized together with the objective function. The constraint utilizes the label information to obtain the discriminate affinity matrix and representations, and the soft constraint prevents over-fitting compared to the hard constraint used in other affinity matrix learning methods. Additionally, the label information forces the data points within the same class have a higher similarity than in the different class. The proposed algorithm learns non-linear representations in fast learning speed and excellent generalization capability. By using ELM-AE, the algorithm can approximate non-linear functions explicitly and obtains non-linear data representations together with the capability to learn the affinity matrix. This work has been submitted in a journal, and the details are as follows:

Yue Li, Yijie Zeng, Tianchi Liu, Xiaofan Jia, and Guang-Bin Huang, "Simultaneously learning affinity matrix and data representations for machine fault diagnosis", *Neural Networks*, 122:395-406, 2020.

### 1.3 Structure of Thesis

The remaining chapters of the thesis are organized as follows.

Chapter 2 reviews the two main categories of representation learning algorithms, which are the geometrically motivated algorithms and reconstruction-based algorithms. It also reviews the related works on ELM and its variants. Lastly, this chapter reviews the applications on machine fault diagnosis.

In Chapter 3, the study investigates the importance of preserving both local and global geometry of input data in deep learning-based representation learning algorithms. It first describes the limitations of the existing algorithms and introduces FAE-LG algorithm and its deep structure SFAE-LG to address those limitations.

In Chapter 4, the study improves the training efficiency of representation learning algorithms that retain the capability of preserving geometry information of input data. It first introduces the merits of using ELM-based algorithm to learn data representations and proposes LDELM-AE algorithm to learn data representations with local geometry preserved and local discrimination exploited.

In Chapter 5, the proposed algorithm learns the affinity matrix and data representations simultaneously. The limitations of using the predefined and fixed affinity matrix to preserve geometry information in representation learning are first studied, and LELMAE-AN algorithm is proposed to learn the affinity matrix that can exploit and preserve the geometry information in both original data space and representation space.

Chapter 6 summarizes the contributions and discoveries of this thesis and suggests future research directions.



# Chapter 2

## Literature Review

*Chapter 2 begins with introductions of representation learning algorithms in Section 2.1, and followed by introductions of extreme learning machine and its variants in Section 2.2. Lastly, a real-world application, bearing fault diagnosis, is briefly introduced in Section 2.3.*



## 2.1 Representation Learning Algorithms

Machine learning methods are affected by redundant information contained in sensor data. Hence, researchers extract features from sensor data based on domain knowledge to remove the redundant information and retain the relevant information. Manual feature extraction can be labor-intensive and time-consuming. The objective of representation learning is to learn useful features that relative to the machine learning tasks, e.g., classification; meanwhile, it removes the redundant information contained in input data. Bengio et al. [7] categorizes representation learning to the probabilistic models, the geometrically motivated models, and the reconstruction-based models. This thesis focus on researches of the geometrically motivated and reconstruction-based representation learning algorithms. Therefore, this section reviews: 1) global geometrically motivated algorithms, i.e., principal component analysis (PCA) [40–42] and linear discriminant analysis (LDA) [27, 43, 44]; 2) local geometrically motivated algorithms, i.e., Laplacian eigenmaps (LE) [26, 45, 46], locally linear embedding (LLE) [25, 47, 48], and locality preserving projection (LPP) [49–51]; 3) reconstruction-based algorithms related to auto-encoders [52–61].

### 2.1.1 Global Geometrically Motivated Algorithms

Global geometry is preserved by retaining the same relationship among all samples before and after embedding data samples from the original data space to the representation space.

#### 2.1.1.1 Principal Component Analysis

PCA [40–42] embeds the dataset to a subspace with the projection directions with minimal variance removed. The embedding is achieved by using a linear projection matrix that seeks a direction with a maximal variance of the dataset. The projection matrix  $\mathbf{W}$  is composed of eigenvectors of the covariance matrix, which are linearly independent vectors. Each eigenvector corresponds with an eigenvalue, and the value of each eigenvalue reflects the amount of variance. The higher eigenvalue has a more significant amount of variance, and its corresponding eigenvector is

the principal component contains more information (variance). By ranking eigenvectors according to their corresponding eigenvalues, i.e., highest to lowest, the principal components are ranked in order of significance. Therefore, PCA embeds the dataset by removing eigenvectors with low eigenvalue in the projection matrix. The objective function of PCA can be formulated as:

$$\begin{aligned} \mathbf{W}^* &= \arg \min_{\mathbf{W}} \mathbf{W}^\top \mathbf{C} \mathbf{W} \\ \text{s.t. } \mathbf{W}^\top \mathbf{W} &= \mathbf{I} \end{aligned} \quad (2.1)$$

where  $\mathbf{C} \in \mathbb{R}^{d \times d} = \frac{1}{n} \sum_i (\mathbf{x}_i - \bar{\mathbf{x}})(\mathbf{x}_i - \bar{\mathbf{x}})^\top$  is the covariance matrix and  $\bar{\mathbf{x}}$  is the mean of all samples. The input data  $\mathbf{X}$  can be embedded to representation space by  $\mathbf{X}_{proj} = \mathbf{W}^\top \mathbf{X}$ .

#### 2.1.1.2 Linear Discriminant Analysis

LDA [27, 43, 44] is a linear, supervised representation learning method that has globality preserving properties. Compared to PCA, LDA uses label information and learn representations that especially for supervised classification tasks. LDA forces the data points to be closed while they belong to the same class and to be far away while they belong to different classes in the feature space. The between-class variance  $\mathbf{S}_B$  of LDA is calculated by:

$$\mathbf{S}_B = \sum_{k=1}^m n_k \mathbf{S}_{Bk} \quad (2.2)$$

where  $m$  is the number of classes, and  $n_i$  is the number of samples in each class. The between class variance of each class  $\mathbf{S}_{Bk}$  stands for the distance between the mean of samples in the  $k$ -th class,  $\bar{\mathbf{x}}_k$ , and the mean of all samples,  $\bar{\mathbf{x}}$ :

$$\mathbf{S}_{Bk} = (\bar{\mathbf{x}}_k - \bar{\mathbf{x}})(\bar{\mathbf{x}}_k - \bar{\mathbf{x}})^\top \quad (2.3)$$

The distance between different classes after projection can be calculated as:

$$\|\mathbf{W}^\top \bar{\mathbf{x}}_k - \mathbf{W}^\top \bar{\mathbf{x}}\|_2^2 = \mathbf{W}^\top \mathbf{S}_{Bk} \mathbf{W} \quad (2.4)$$

where  $\mathbf{W}^\top \bar{\mathbf{x}}$  is the projection of the mean of all samples, and  $\mathbf{W}^\top \bar{\mathbf{x}}_k$  is the projection of the mean of samples in the  $k$ -th class.  $\mathbf{W}$  is the projection matrix of LDA. Furthermore, the within-class variance  $\mathbf{S}_W$  of LDA is calculated by:

$$\mathbf{S}_W = \sum_{k=1}^m \mathbf{S}_{Wk} \quad (2.5)$$

The within-class variance  $\mathbf{S}_{Wk}$  stands for the difference between the mean and the samples in the  $k$ -th class, which can be calculated by:

$$\mathbf{S}_{Wk} = \sum_{j=1}^{n_k} (x_{kj} - \bar{\mathbf{x}}_k)(x_{kj} - \bar{\mathbf{x}}_k)^\top \quad (2.6)$$

where  $x_{kj}$  represents the  $j$ -th sample in the  $k$ -th class. The difference between projected mean and the projected samples of each class is defined as the following:

$$|\mathbf{W}^\top \mathbf{x}_k - \mathbf{W}^\top \bar{\mathbf{x}}_k|^2 = \mathbf{W}^\top \mathbf{S}_{Wk} \mathbf{W} \quad (2.7)$$

Hence, LDA finds a projection matrix  $\mathbf{W}$  to maximise the between-class variance and minimise the within-class variance:

$$\mathbf{W}^* = \arg \min_{\mathbf{W}} \frac{\mathbf{W}^\top \mathbf{S}_B \mathbf{W}}{\mathbf{W}^\top \mathbf{S}_W \mathbf{W}} \quad (2.8)$$

Similar to PCA, the input data  $\mathbf{X}$  can be embedded to representation space by  $\mathbf{X}_{proj} = \mathbf{W}^\top \mathbf{X}$ .

## 2.1.2 Local Geometrically Motivated Algorithms

Local geometry is preserved by retaining the same relationship between a sample and its neighbors mapping data samples from the original data space to the data representation space.

### 2.1.2.1 Locally Linear Embedding

As both PCA and LDA exploit global geometry information of input data, LLE [25, 47, 48] learns data representation and discovers local geometry information by assuming that if each data point and its neighbors lie on or close to a locally

linear patch of the manifold, the high dimensional data can be considered as locally linear. LLE first finds the neighbors of each data point  $\mathbf{x}_i \quad \forall i = 1, \dots, n$ , which often achieved by using k-nearest neighbors. It then determines linear combination weights  $\mathbf{W} = \{w_{ij}\}$ , where  $i, j = 1, \dots, n$ , to each pair of neighboring points. The weights can be founded by:

$$\begin{aligned} \mathbf{W}^* &= \arg \min_{\mathbf{W}} \sum_{i=1}^n \left\| \mathbf{x}_i - \sum_{j=1}^n w_{ij} \mathbf{x}_j \right\|^2 \\ \text{s.t.} \quad &\sum_{i=1}^n w_{ij} = 1 \\ &w_{ij} = 0, \quad \text{if } \mathbf{x}_i \text{ and } \mathbf{x}_j \text{ are not neighbors} \end{aligned} \quad (2.9)$$

The constraint enforces translation invariance of  $\mathbf{x}_i$  and its neighbors. To preserve the local linear structure from data space, the weights  $\mathbf{W}$  reconstructs data  $\mathbf{x}_i$  from its neighbors in data space should also reconstructs its projected manifold coordinates in the representation space. Hence, the embedded coordinates  $\hat{\mathbf{x}}_i$  can be found by the following cost function:

$$\begin{aligned} \mathbf{X}_{proj}^* &= \arg \min_{\mathbf{X}_{proj}} \sum_{i=1}^n \left\| \hat{\mathbf{x}}_i - \sum_{j=1}^n w_{ij} \hat{\mathbf{x}}_j \right\|^2 \\ \text{s.t.} \quad &\frac{1}{n} \sum_{i=1}^n \hat{\mathbf{x}}_i \hat{\mathbf{x}}_i^\top = \mathbf{I} \\ &\sum_{i=1}^n \hat{\mathbf{x}}_i = \mathbf{0} \end{aligned} \quad (2.10)$$

where  $\mathbf{I}$  is an identity matrix and  $\mathbf{0}$  is a zero-vector.  $\mathbf{X}_{proj} = [\hat{\mathbf{x}}_1, \dots, \hat{\mathbf{x}}_n]$  is the matrix of embedded data representations. The first constraint removes the rotational degree of freedom and fixes the scale, and the second constraint removes the translation degree of freedom. Therefore, it can obtain the unique solution.

### 2.1.2.2 Laplacian Eigenmaps

LE [26, 45, 46] is another computationally efficient algorithm for non-linear representation learning that has locality preserving properties. Similar to LLE, LE firstly constructs a graph by connecting samples in neighborhoods. The following two methods can define the neighbors:

- (a).  $\epsilon$ -neighborhoods. The  $i$ -th and  $j$ -th samples are connected by an edge if  $\|\mathbf{x}_i - \mathbf{x}_j\|^2 < \epsilon$ , where  $\epsilon$  is the hyper-parameter defined by users.
- (b).  $k$ -nearest neighbors. The  $i$ -th and  $j$ -th samples are connected by an edge if sample  $\mathbf{x}_j$  is within  $k$  nearest neighbors of sample  $\mathbf{x}_i$ , where  $k$  is the hyper-parameter defined by users.

Next, the weights of edges  $s_{ij}$ , which is also known as similarities, between two neighbors nodes  $\mathbf{x}_i$  and  $\mathbf{x}_j$  can be determined by two methods:

- (a). Heat kernel. If samples  $\mathbf{x}_i$  and  $\mathbf{x}_j$  are neighbors:

$$s_{ij} = e^{-\frac{\|\mathbf{x}_i - \mathbf{x}_j\|^2}{\sigma}} \quad (2.11)$$

where  $\sigma$  is a hyper-parameter.

- (b). Simple method. If samples  $\mathbf{x}_i$  and  $\mathbf{x}_j$  are neighbors,  $s_{ij} = 1$ . Otherwise,  $s_{ij} = 0$ .

After the affinity matrix  $\mathbf{S}$ , which is also known as similarity matrix, is formed. The embedded data representation  $\mathbf{X}_{proj} = \{\hat{\mathbf{x}}_i\}_{i=1}^n$  is then determined by:

$$\mathbf{X}_{proj}^* = \arg \min_{\mathbf{X}_{proj}} \sum_{i=1, j=1}^t (\hat{\mathbf{x}}_i - \hat{\mathbf{x}}_j)^2 s_{ij} \quad (2.12)$$

where  $t$  is the number of connected nodes. It can be solved by computing eigenvalues and eigenvectors of the generalized eigenvalue problem:

$$\mathbf{L}\hat{\mathbf{x}} = \lambda\mathbf{D}\hat{\mathbf{x}} \quad (2.13)$$

where  $\mathbf{D}$  is diagonal weight matrix with  $D_{ii} = \sum_j w_{ij}$ , and  $\mathbf{L} = \mathbf{D} - \mathbf{W}$  is the Laplacian matrix.

### 2.1.2.3 Locality Preserving Projection

Although LLE and LE exploit local geometry information of input data, they can only process the existing training data samples. In another words, it is hard to

apply them on the new test samples. LPP [49–51] learns a linear projection model to preserve local geometry in data representations. Hence, compared to LLE and LE, LPP naturally can process the new test data samples.

LPP uses the same affinity matrix as LE in Section 2.1.2.2, and determines the linear projection matrix  $\mathbf{W}$  as:

$$\begin{aligned}
\mathbf{W}^* &= \arg \min_{\mathbf{W}} \sum_{ij} (\mathbf{W}^\top \mathbf{x}_i - \mathbf{W}^\top \mathbf{x}_j)^2 s_{ij} \\
&= \arg \min_{\mathbf{W}} \sum_i \mathbf{W}^\top \mathbf{x}_i D_{ii} \mathbf{x}_i^\top \mathbf{W} - \sum_{ij} \mathbf{W}^\top \mathbf{x}_i s_{ij} \mathbf{x}_j^\top \mathbf{W} \\
&= \arg \min_{\mathbf{W}} \mathbf{W}^\top \mathbf{X} \mathbf{L} \mathbf{X}^\top \mathbf{W} \\
\text{s.t. } &\mathbf{W}^\top \mathbf{X} \mathbf{D} \mathbf{X}^\top \mathbf{W} = \mathbf{I}_l
\end{aligned} \tag{2.14}$$

where  $\mathbf{D}$  is a diagonal matrix with  $D_{ii} = \sum_j s_{ij}$ , and  $\mathbf{L} = \mathbf{D} - \mathbf{S}$  is the graph Laplacian matrix [62].  $\mathbf{I}_l \in \mathbb{R}^{l \times l}$  is an identity matrix. It can be noticed that LPP is the linear approximation of LE.

### 2.1.3 General Graph Embedding

From the above discussions, the geometrically motivated representation learning algorithms, i.e., PCA, LDA, LLE, LE, and LPP, can be summarized to a general graph embedding formulation [63]. Preserving geometry in data representations can be achieved by treating data sample as vertices of an undirected weighted graph  $\mathbf{G} = (\mathbf{X}, \mathbf{S})$ . Each column of  $\mathbf{X}$  is a vertex of the graph, and each element of  $\mathbf{S}$  represents the weight between two vertices.

The graph embedding is defined under the assumption of smoothness, which assumes the data samples that are closed in the original space should also be closed in the embedded space. Yan et al. [63] summarized the graph embedding into a generalized graph preserving criterion:

$$\begin{aligned}
\mathbf{X}_{proj}^* &= \arg \min_{\hat{\mathbf{x}}_i \in \mathbf{X}_{proj}} \sum_{i \neq j} \|\hat{\mathbf{x}}_i - \hat{\mathbf{x}}_j\|^2 s_{ij} \\
&= \arg \min_{\hat{\mathbf{x}}_i \in \mathbf{X}_{proj}} \mathbf{X}_{proj}^\top \mathbf{L} \mathbf{X}_{proj} \\
\text{s.t. } &\mathbf{X}_{proj}^\top \mathbf{B} \mathbf{X}_{proj} = \mathbf{C}
\end{aligned} \tag{2.15}$$

where  $\mathbf{X}_{proj} = \{\hat{\mathbf{x}}_i\}_{i=1}^n$  is the matrix of embedded samples.  $c$  is a constant and  $\mathbf{L} = \mathbf{D} - \mathbf{S}$  is the graph Laplacian matrix.  $\mathbf{B}$  is the constraint matrix.

In order to process new test data samples, the Equation (2.15) is extended with a linear projection:

$$\begin{aligned} \mathbf{W}^* &= \arg \min_{\mathbf{W}} \mathbf{W}^\top \mathbf{X} \mathbf{L} \mathbf{X}^\top \mathbf{W} \\ \text{s.t. } &\mathbf{W}^\top \mathbf{X} \mathbf{B} \mathbf{X}^\top \mathbf{W} = \mathbf{C} \end{aligned} \quad (2.16)$$

Therefore, the previous geometrically motivated representation learning algorithms, i.e., PCA, LDA, LLE, LE, and LPP, can be written as special cases of Equation (2.15) and Equation (2.16):

- PCA can be written as a special case of Equation (2.16) with  $\mathbf{B} = \mathbf{I}$ , and  $s_{ij} = \frac{1}{n}, \forall i \neq j$ .
- LDA can be written as a special case of Equation (2.16) with  $\mathbf{B} = \mathbf{I} - \frac{1}{n} \mathbf{e} \mathbf{e}^\top$ , and  $s_{ij} = \frac{\delta_{c_i, c_j}}{n_{c_i}}$ .
- LLE can be written as a special case of Equation (2.15) with  $\mathbf{B} = \mathbf{I}$ , and  $\mathbf{S} = \mathbf{W} + \mathbf{W}^\top - \mathbf{W}^\top \mathbf{W}$ , where  $\mathbf{W}$  is determined by Equation (2.9).
- LE can be written as a special case of Equation (2.15) with  $\mathbf{B} = \mathbf{D}$ , and  $s_{ij} = e^{-\frac{\|\mathbf{x}_i - \mathbf{x}_j\|^2}{\sigma}}$  as shown in Equation (2.11).
- LPP can be written as a special case of Equation (2.16) with  $\mathbf{B} = \mathbf{D}$ , and  $s_{ij} = e^{-\frac{\|\mathbf{x}_i - \mathbf{x}_j\|^2}{\sigma}}$  as shown in Equation (2.11).

Based on the above discussions, there are two weaknesses of the existing geometrically motivated representation learning algorithms. Firstly, the existing algorithms either have no models for the incoming test data or can only embed the test data linearly. Although the kernel trick can achieve the non-linearity, it is hard to choose the correct kernel function for various machine learning tasks. Secondly, the affinity matrix used in existing algorithms is predefined and fixed. However, the assumed prior knowledge might not precisely represent the real geometry relationships between data points.

### 2.1.4 Auto-encoders

The deep-learning-based algorithms have been developed to improve the capability of learning non-linearly data representations.

AE [54] is a reconstruction-based representation learning algorithm that maps the input data sample to itself, and it consists of two parts, which are the encoder and decoder. The structure of AE is shown in Figure 2.1. Firstly, the encoder maps the input data sample  $\mathbf{x}_i$  to a non-linear hidden representation  $\mathbf{h}_i = [h_i^1, \dots, h_i^l]$ :

$$\mathbf{h}_i = g(\mathbf{x}_i \mathbf{W} + \mathbf{b}) \quad (2.17)$$

where  $\mathbf{b} = [b^1, \dots, b^l]$  are the bias of hidden nodes, and  $\mathbf{W} \in \mathbb{R}^{d \times l}$  is the projection matrix for the encoding process. Then, the decoder reconstructs the input data  $\mathbf{x}_i$  from the hidden representation  $\mathbf{h}_i$ :

$$\tilde{\mathbf{x}}_i = g(\mathbf{h}_i \mathbf{W}' + \mathbf{c}) \quad (2.18)$$

where  $\tilde{\mathbf{x}}_i$  is the reconstructed data of  $i$ -th sample, and  $\mathbf{W}' \in \mathbb{R}^{l \times d}$  is the projection matrix for the decoding process.  $\mathbf{c} = [c^1, \dots, c^d]$  are biases of output nodes.  $g(\cdot)$  is the activation function, which normally using sigmoid function in Equation (2.19):

$$g(z) = \frac{1}{1 + \exp(-z)} \quad (2.19)$$

For a convenient training process, the tied weights AE constrains the output weight  $\mathbf{W}'$  to be the transpose of the input weight:  $\mathbf{W}' = \mathbf{W}^\top$ . The tied weights of AE can also be viewed as one kind of regularization to prevent the over-fitting issue. The objective function of AE is to minimize the difference between original data  $\mathbf{x}$  and reconstructed data  $\tilde{\mathbf{x}}$  from Equation (2.18):

$$J(\mathbf{W}, \mathbf{b}, \mathbf{c}) = \min_{\mathbf{W}, \mathbf{b}, \mathbf{c}} \frac{1}{n} \sum_{i=1}^n (\mathbf{x}_i - \tilde{\mathbf{x}}_i)^2 \quad (2.20)$$

Objective function Equation (2.20) can be minimized by using a gradient descent-based optimization tool. The output of the hidden layer is the data representation of the input data.



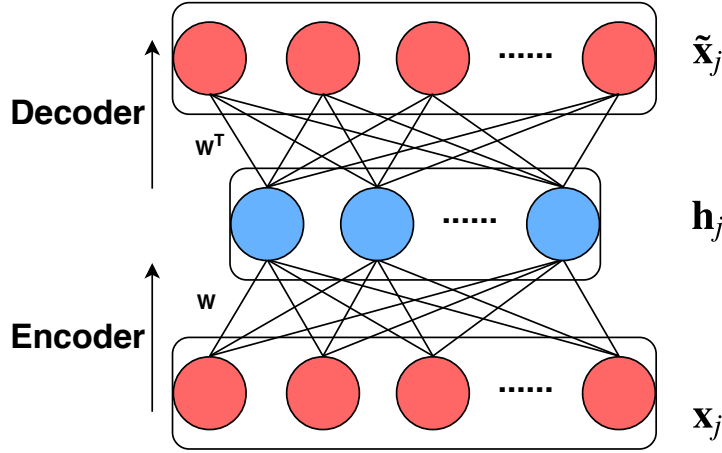


FIGURE 2.1: Architecture of AE.

#### 2.1.4.1 Denoising Auto-Encoder (DAE)

The denoising auto-encoder (DAE) randomly corrupts the input to zero, and learns noise robust data representations to recover the original data [58]. In contrast with AE, DAE attempts to reconstruct the inputs from the corrupted input data. The encoder of DAE is shown in Equation (2.21).

$$\mathbf{h}_i = f(\mathbf{x}'_i \mathbf{W} + \mathbf{b}) \quad (2.21)$$

where  $f$  is the activation function.  $\mathbf{b} = [b^1, \dots, b^L]$  are the bias of hidden nodes,  $\mathbf{W} \in \mathbb{R}^{d \times L}$  is the projection matrix for the encoding process, and  $\mathbf{x}'_i$  is the corrupted version of input data  $\mathbf{x}_i$ . To reconstruct the input  $\mathbf{x}_i$ , the decoder of DAE is shown in Equation (2.22).

$$\tilde{\mathbf{x}}_i = f(\mathbf{h}_i \mathbf{W}' + \mathbf{c}) \quad (2.22)$$

where  $\tilde{\mathbf{x}}_i$  is the reconstructed data of the  $i$ -th sample.  $\mathbf{W}'$  is the projection matrix for the decoding process, and  $\mathbf{c} = [c^1, \dots, c^d]$  are biases of output nodes. Similar to AE, the encoder maps the corrupted input data  $\mathbf{x}'_i$  to a hidden representation  $\mathbf{h}_i$ . And the decoder attempts to reconstruct the input data  $\mathbf{x}_i$  from its hidden representation  $\mathbf{h}_i$ . By reconstructing original inputs from corrupted input data, the DAE not only learns hidden representations, but also capturing the statistical dependencies between the input data samples.

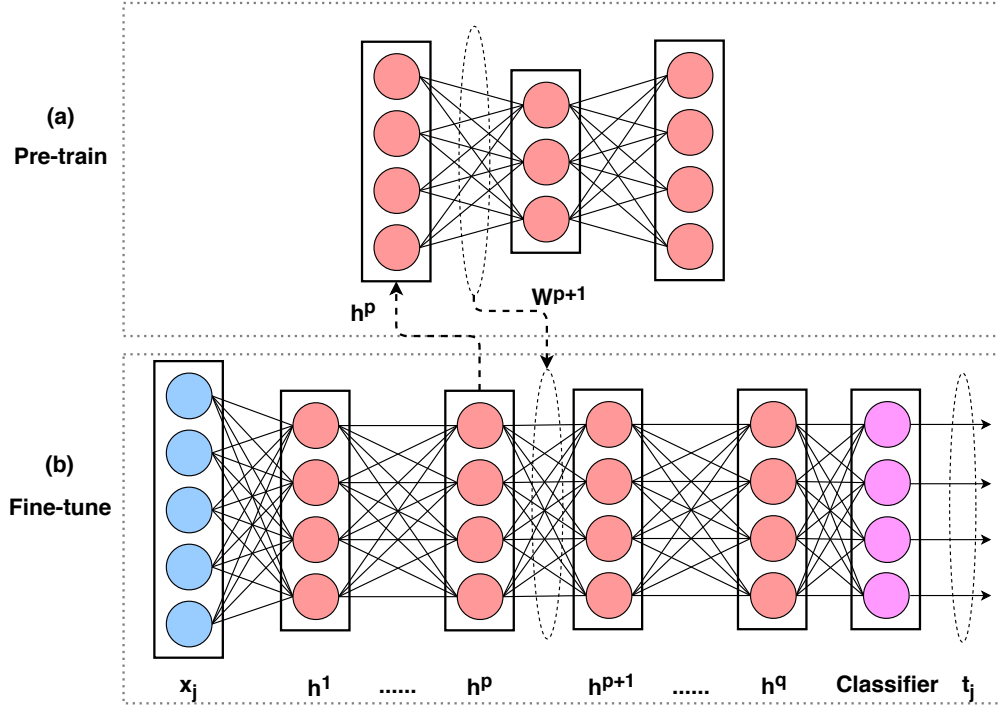


FIGURE 2.2: Architecture of SAE.

#### 2.1.4.2 Stacked Auto-encoder (SAE)

Hierarchical representations are achieved by stacking multiple AEs, denoted as stacked auto-encoder (SAE), which is shown in Figure 2.2. Each hidden layer of SAE is expected to improve the representation based on the previous layer. The output of the last hidden layer can be used for other machine learning tasks, e.g., classification. The weights in the SAE network are learned in two steps. In the first step, each hidden layer is initialized with the weights trained by an AE based on the output of the previously hidden layer. For example, the input weights of the  $(p + 1)$ -th hidden layer is initialized with the trained weights by the  $(p + 1)$ -th AE, whose input is the output of the  $p$ -th SAE hidden layer. Secondly, a softmax classifier is stacked after the last hidden layer of SAE using the label information as the target output. The prediction error given by the softmax classifier is propagated back through the whole SAE network and used to tune all weights. The first step is usually referred to as pre-training and the second step as fine-tuning. Because the fine-tuning step makes use of the label information, the final representation given by SAE has the discriminative capability, and it is suitable for classification.

However, it is worth noting that although reconstruction-based representation learning algorithms can embed the input data to non-linear data representations,

their training procedures are time-consuming. Moreover, the existing reconstruction-based algorithms do not consider any relationship among data points. An important property of data representations is to preserve the geometry of data points, i.e., the local and global geometries. The first work of this thesis investigates the geometry preserving in reconstruction-based algorithms.

## 2.2 Extreme Learning Machines

This thesis has reviewed the reconstruction-based representation learning algorithms, i.e., AEs, which use back-propagation (BP) to optimize their objective functions. BP optimizes objective function iteratively based on gradient information. Hence, the training process is time-consuming and might converge to a local minimum. Huang et al. [29–38] developed the extreme learning machine (ELM) to avoid using BP as the optimization algorithm. ELM is a single layer feed-forward neural network (SLFN) with randomly generated and fixed hidden neurons. Huang theoretically proved that the hidden neuron of SLFN can be randomly generated without tuning. Therefore, only output weights need to be learned in ELM. The output weights connect the hidden layer and the output layer. ELM has proved to solve various machine learning tasks, such as regression [35], classification [35], clustering [64], and representation learning [13]. Moreover, ELM has been applied on many real-world applications [65–71]. Figure 2.3 shows the architecture of ELM, which is originally proposed as the generalized SLFN. Given a input data  $\{\mathbf{x}_i \in \mathbb{R}^d, \mathbf{y}_i \in \mathbb{R}^m\}$ , ELM first randomly embeds it to ELM feature space by:

$$h_i^j = g(\mathbf{a}_j, b_j, \mathbf{x}_i^j) \quad \forall j = 1, \dots, l \quad (2.23)$$

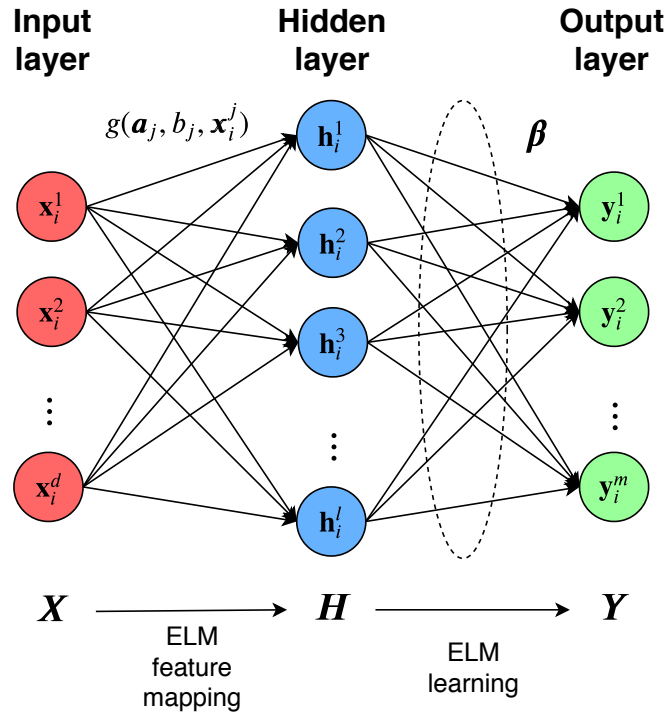


FIGURE 2.3: Architecture of ELM.

where  $\mathbf{h}_i^j$  is the embedded ELM feature, also known as the output of hidden neuron, of the corresponding input data  $\mathbf{x}_i$ .  $g(\mathbf{a}_j, b_j, \mathbf{x}_i^j)$  is a piecewise non-linear activation function, and  $(\mathbf{a}_j \in \mathbb{R}^d, b_j \in \mathbb{R})$  are the parameters of the activation function in the  $j$ -th hidden neuron. For example, the non-linear activation functions can be as follows [35]:

- Sigmoid function.

$$g(\mathbf{a}, b, \mathbf{x}_i) = \frac{1}{1 + \exp(-(\mathbf{a} \cdot \mathbf{x}_i + b))} \quad (2.24)$$

- Hard-limit function.

$$g(\mathbf{a}, b, \mathbf{x}_i) = \begin{cases} 1, & \text{if } \mathbf{a} \cdot \mathbf{x}_i - b \geq 0 \\ 0, & \text{otherwise.} \end{cases} \quad (2.25)$$

- Gaussian function.

$$g(\mathbf{a}, b, \mathbf{x}_i) = \exp(-b\|\mathbf{x}_i - \mathbf{a}\|^2) \quad (2.26)$$

- Multiquadric function.

$$g(\mathbf{a}, b, \mathbf{x}_i) = \sqrt{(\|\mathbf{x}_i - \mathbf{a}\|^2 + b^2)} \quad (2.27)$$

Based on the embedded ELM features  $\mathbf{h}_i = [h_i^1, h_i^2, \dots, h_i^l]^\top$ , the ELM output can be determined by:

$$f_L(\mathbf{x}_i) = \boldsymbol{\beta} \mathbf{h}_i \quad (2.28)$$

where  $\boldsymbol{\beta} \in \mathbb{R}^{m \times l}$  is the output weight matrix connects the hidden layer and the output layer.

The original ELM is proposed for regression and classification tasks. Given a dataset  $\{\mathbf{x}_i, \mathbf{y}_i\}_{i=1}^n$ , the output weight matrix  $\boldsymbol{\beta}$  can be found by solving the following mathematical formulation:

$$\arg \min_{\boldsymbol{\beta}} \|\boldsymbol{\beta} \mathbf{H} - \mathbf{Y}\|^2 \quad (2.29)$$

where  $\mathbf{H} \in \mathbb{R}^{l \times n}$  is the hidden-layer output matrix

$$\begin{aligned} \mathbf{H} &= [\mathbf{h}_1, \mathbf{h}_2, \dots, \mathbf{h}_n] \\ &= \begin{bmatrix} h_1^1 & h_2^1 & \cdots & h_n^1 \\ h_1^2 & h_2^2 & \cdots & h_n^2 \\ \vdots & \vdots & \ddots & \vdots \\ h_1^l & h_2^l & \cdots & h_n^l \end{bmatrix} \end{aligned} \quad (2.30)$$

and  $\mathbf{Y} = [\mathbf{y}_1, \mathbf{y}_2, \dots, \mathbf{y}_n]$ , where  $\mathbf{y}_i$  is a  $m$ -dimensional one-hot column vector that only one element, which corresponds to the class of  $i$ -th sample, equals to 1 and the other elements equal to 0.

Therefore,

$$\boldsymbol{\beta} = \mathbf{Y} \mathbf{H}^\dagger \quad (2.31)$$

where  $\mathbf{H}^\dagger$  is the Moore–Penrose generalized inverse of matrix  $\mathbf{H}$ .

ELM is well known for its efficient training procedure, which is an analytical solution, as shown in Equation (2.31). Furthermore, ELM has the universal approximation capability [33], which means that ELM can approximate any target functions with its randomly generated neurons. Hence, many variants of ELM has been introduced by utilizing its advantages. The following section reviews some variants of ELM, which including the regularized extreme learning machine [35, 72, 73] for classification and regression, the Extreme Learning Machine Auto-encoder (ELM-AE) [13, 14] for representation learning, the semi-supervised extreme learning machine (SS-ELM) [64] that exploits geometry information, the Unsupervised Extreme Learning Machine (US-ELM) [64] that uses geometry information for clustering, the Generalized Extreme Learning Machine Auto-encoder (GELM-AE) [74] that preserves geometry information in representation learning, and the Extreme Learning Machine with Constrained Laplacian Rank (ELM-CLR) [75] that investigates using adaptive neighbors for clustering.

### 2.2.1 Regularized Extreme Learning Machine

Based on the original ELM, a regularization term is proposed to prevent overfitting [35, 72, 73]. Also, Bartlett [76] shows that minimizing the norm of the output weights improves the generalization capability. The mathematical formulation of

the regularized ELM can be written as:

$$\arg \min_{\beta} \frac{C}{2} \|\beta \mathbf{H} - \mathbf{Y}\|^2 + \frac{1}{2} \|\beta\|^2 \quad (2.32)$$

where  $C$  is a hyper-parameter determined by users.

While the number of samples larger than the number of hidden neurons, i.e.,  $n > l$ , the problem can be solved by letting  $\nabla_{\beta} = 0$ , where

$$\begin{aligned} \nabla_{\beta} &= C(\beta \mathbf{H} - \mathbf{Y}) \mathbf{H}^{\top} + \beta \\ &= \beta (C \mathbf{H} \mathbf{H}^{\top} + \mathbf{I}) - \mathbf{Y} \mathbf{H}^{\top} \end{aligned} \quad (2.33)$$

While the number of samples smaller than the number of hidden neurons, i.e.,  $n < l$ ,  $\mathbf{H}$  will have more rows than columns, which is the under-determined problem of least-square methods. To solve this problem,  $\beta$  can be restricted to a linear combination of the columns of  $\mathbf{H}$ , i.e.,  $\beta = \alpha \mathbf{H}^{\top}$ , where  $\alpha \in \mathbb{R}^{d \times n}$ .

Therefore, the solution of  $\beta$  is:

$$\beta^* = \begin{cases} \mathbf{Y} \mathbf{H}^{\top} (C \mathbf{H} \mathbf{H}^{\top} + \mathbf{I})^{-1}, & \text{if } N > l \\ \mathbf{Y} (C \mathbf{H}^{\top} \mathbf{H} + \mathbf{I})^{-1} \mathbf{H}^{\top}, & \text{otherwise.} \end{cases} \quad (2.34)$$

### 2.2.2 Extreme Learning Machine Auto-encoder

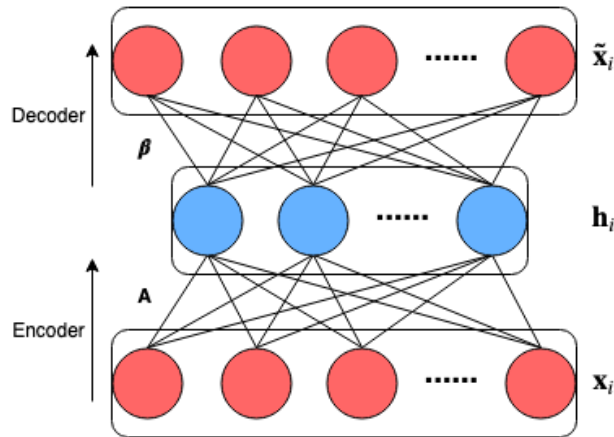


FIGURE 2.4: The architecture of ELM-AE.

Although ELM can randomly map input data to ELM features, the data representation learning capability of it is still lacking. Kasun et al. [13, 14] investigated the

data representation learning capability of ELM and proposed ELM-AE, which is a representation learning algorithm.

ELM-AE is an unsupervised representation learning algorithm, which uses randomly generated and fixed hidden neurons to embed input data to another space. It learns the output weights to reconstruct the input data from the outputs of the hidden layer.

As shown in Figure 2.4, given a sample  $\mathbf{x}_i$ , the output of the hidden layer  $\mathbf{h}_i$  can be encoded by a randomly and non-linearly mapping function, e.g., the sigmoid function.

$$\mathbf{h}_i = \frac{1}{1 + \exp(\mathbf{A}\mathbf{x}_i)} \quad (2.35)$$

$\mathbf{A} \in \mathbb{R}^{l \times d}$  is a randomly generated weight matrix. The objective of ELM-AE is to minimize the error between the original data and the reconstructed data:

$$\min_{\boldsymbol{\beta}} \frac{1}{2} \|\boldsymbol{\beta}\|_F^2 + \frac{C}{2} \|\mathbf{X} - \boldsymbol{\beta}\mathbf{H}\|_F^2 \quad (2.36)$$

where  $\mathbf{H} = [\mathbf{h}_1, \mathbf{h}_2, \dots, \mathbf{h}_N] \in \mathbb{R}^{l \times N}$  is the output matrix of hidden layer, and  $\boldsymbol{\beta}$  is the output weight matrix connects the hidden layer and the output layer.  $C > 0$  is a trade-off parameter. The first term is a regularization term to prevent over-fitting, and the second term is the reconstruction error between the reconstructed samples and the original samples. The output weight matrix  $\boldsymbol{\beta}$  can be analytically solved:

$$\boldsymbol{\beta}^* = \begin{cases} \mathbf{X}\mathbf{H}^\top \left( \mathbf{H}\mathbf{H}^\top + \frac{\mathbf{I}_l}{C} \right)^{-1} & \text{if } N \geq l \\ \mathbf{X} \left( \mathbf{H}^\top \mathbf{H} + \frac{\mathbf{I}_N}{C} \right)^{-1} \mathbf{H}^\top & \text{if } N < l \end{cases} \quad (2.37)$$

where  $\mathbf{I}_l$  and  $\mathbf{I}_N$  is the identity matrix with dimension of  $l$  and  $N$ , respectively. The representations of the input data  $\mathbf{X}$  can be obtained by  $\mathbf{X}_{proj} = \boldsymbol{\beta}^\top \mathbf{X}$ . They are then used for the further classification tasks.



### 2.2.3 Semi-supervised Extreme Learning Machine

This thesis also reviews the ELM variants that preserves the geometry information of input data. Huang et al. [64] proposed SS-ELM by using the manifold regularization in the regularized ELM.

**Manifold Regularization.** The manifold regularization [77] is similar to LE in Section 2.1.2.2. Under the smooth assumption of machine learning, which assumes if samples  $\mathbf{x}_i$  and  $\mathbf{x}_j$  are close to each other in the input data space, the conditional probabilities  $p(\mathbf{y}|\mathbf{x}_i)$  and  $p(\mathbf{y}|\mathbf{x}_j)$  should be similar as well, the loss function of the manifold regularization can be formed as:

$$L = \frac{1}{2} \sum_{i,j} \|p(\mathbf{y}|\mathbf{x}_i) - p(\mathbf{y}|\mathbf{x}_j)\|^2 s_{ij} \quad (2.38)$$

The affinity matrix  $\mathbf{S} = \{s_{ij}\}_{i,j=1}^n$  can be determined by using the heat kernel method or simple method in Section 2.1.2.2. Since it is difficult to compute the conditional probability, the manifold regularization can be simplified to its approximation expression:

$$\hat{L} = \frac{1}{2} \sum_{i,j} \|\hat{\mathbf{y}}_i - \hat{\mathbf{y}}_j\|^2 s_{ij} \quad (2.39)$$

where  $\hat{\mathbf{y}}_i$  and  $\hat{\mathbf{y}}_j$  are the predictions with respect to  $\mathbf{x}_i$  and  $\mathbf{x}_j$ , respectively. The loss function can then be vectorized as:

$$\hat{L} = \text{Tr}(\hat{\mathbf{Y}} \mathbf{L} \hat{\mathbf{Y}}^\top) \quad (2.40)$$

where  $\mathbf{L}$  is the graph Laplacian matrix as shown in Section 2.1.2.2.

**SS-ELM.** Given a dataset  $\{\mathbf{X}_l, \mathbf{Y}_l\} = \{\mathbf{x}_i, \mathbf{y}_i\}_{i=1}^{n_l}$ , and unlabelled data  $\mathbf{X}_u = \{\mathbf{x}_i\}_{i=1}^{n_u}$ , where  $n_l$  and  $n_u$  are the number of labelled and unlabelled data, respectively. Based on the regularized ELM and manifold regularization, the cost function of SS-ELM can be formulated as:

$$\begin{aligned} \arg \min_{\boldsymbol{\beta}} & \frac{1}{2} \|\boldsymbol{\beta}\|^2 + \frac{1}{2} \sum_{i=1}^{n_l} C_i \|\boldsymbol{\beta} \mathbf{h}_i - \mathbf{y}_i\|^2 + \frac{\lambda}{2} \text{Tr}(\mathbf{F} \mathbf{L} \mathbf{F}^\top) \\ \text{s.t. } & \mathbf{f}_i = \boldsymbol{\beta} \mathbf{h}_i, \quad \forall i = 1, \dots, n_l + n_u \end{aligned} \quad (2.41)$$

The cost function can be vectorized as:

$$\begin{aligned} \arg \min_{\boldsymbol{\beta}} \frac{1}{2} \|\boldsymbol{\beta}\|^2 + \frac{1}{2} \|\mathbf{C}^{\frac{1}{2}}(\boldsymbol{\beta}\mathbf{H} - \tilde{\mathbf{Y}})\|^2 \\ + \frac{\lambda}{2} \text{Tr}(\boldsymbol{\beta}\mathbf{H}\mathbf{L}\mathbf{H}^\top \boldsymbol{\beta}^\top) \end{aligned} \quad (2.42)$$

where  $\tilde{\mathbf{Y}}$  is the augmented training target with its first  $n_l$  columns equal to  $\mathbf{Y}_l$  and the rest equal to 0.  $\mathbf{C} \in \mathbb{R}^{(n_l+n_u) \times (n_l+n_u)}$  is a diagonal matrix with its first  $n_l$  diagonal elements  $C_{ii} = C_i$  and the rest equal to 0. The cost function can be minimized analytically as similar as the regularized ELM in Section 2.2.1.

### 2.2.4 Unsupervised Extreme Learning Machine

Huang et al. [64] also proposed US-ELM to investigate the clustering capability of ELM while the label information is lacking. Given a unlabelled dataset  $\mathbf{X} = \{\mathbf{x}_i\}_{i=1}^n$ , the task is to discover the underline clusters of all data samples. The mathematical formulation of US-ELM is:

$$\begin{aligned} \arg \min_{\boldsymbol{\beta}} \frac{1}{2} \|\boldsymbol{\beta}\|^2 + \lambda \text{Tr}(\boldsymbol{\beta}\mathbf{H}\mathbf{L}\mathbf{H}^\top \boldsymbol{\beta}^\top) \\ \text{s.t. } \boldsymbol{\beta}\mathbf{H}\mathbf{H}^\top \boldsymbol{\beta}^\top = \mathbf{I} \end{aligned} \quad (2.43)$$

The cost function of US-ELM can be minimized by solving the generalized eigenvalue problem:

$$(\mathbf{I} + \lambda \mathbf{H}\mathbf{L}\mathbf{H}^\top) \mathbf{v} = \gamma \mathbf{H}\mathbf{H}^\top \mathbf{v} \quad (2.44)$$

The rows of  $\boldsymbol{\beta}^*$  are the eigenvectors corresponding to the first  $l$  smallest eigenvalues.

### 2.2.5 Generalized Extreme Learning Machine Auto-encoder

Sun et al. [74] proposed GELM-AE to investigate the geometry preserving capability in the ELM-based representation learning algorithm. GELM-AE constraints ELM-AE by using the manifold regularization, which describes in Section 2.2.3. Hence, the cost function of GELM-AE can be formed as:

$$\min_{\boldsymbol{\beta}} \frac{1}{2} \|\boldsymbol{\beta}\|_F^2 + \frac{C}{2} \|\mathbf{X} - \boldsymbol{\beta}\mathbf{H}\|_F^2 + \frac{\lambda}{2} \text{Tr}(\boldsymbol{\beta}\mathbf{H}\mathbf{L}\mathbf{H}^\top \boldsymbol{\beta}^\top) \quad (2.45)$$

The manifold regularization forces the data representations  $\mathbf{h}(\mathbf{x}_i)$  and  $\mathbf{h}(\mathbf{x}_j)$  are similar if samples  $\mathbf{x}_i$  and  $\mathbf{x}_j$  are close to each other in the original data space. According to Section 2.2.1, the output weights  $\boldsymbol{\beta}$  can be analytically solved by:

$$\boldsymbol{\beta}^* = \begin{cases} \mathbf{X}\mathbf{H}^\top(\mathbf{I} + \mathbf{C}\mathbf{H}\mathbf{H}^\top + \lambda\mathbf{H}\mathbf{L}\mathbf{H}^\top)^{-1}, & \text{if } N \geq l \\ \mathbf{X}(\mathbf{I} + \mathbf{C}\mathbf{H}^\top\mathbf{H} + \lambda\mathbf{H}^\top\mathbf{H}\mathbf{L})^{-1}\mathbf{H}^\top, & \text{if } N < l \end{cases} \quad (2.46)$$

Sun et al. [74] experimentally proved that learning data representations with local geometry preserved performs better than the original ELM-AE in various standard benchmark datasets. Motivated by SS-ELM and GELM-AE, the second work of this thesis investigates the ELM-based representation learning algorithm that exploits both local geometry information and local discriminant information of input data.

### 2.2.6 Extreme Learning Machine with Constrained Laplacian Rank

The above SS-ELM, US-ELM, and GELM-AE investigate the local geometry preserving capability of ELM and ELM-AE. However, they preserve local geometry by using a predefined and fixed affinity matrix. Liu et al. [75] investigates to learn the affinity matrix in the ELM-based clustering algorithm, which is named as ELM-CLR. The cost function of ELM-CLR can be formulated as:

$$\begin{aligned} \arg \min_S \sum_{i,j=1}^n (\bar{d}_{ij}\bar{p}_{ij}s_{ij} + \gamma s_{ij}^2) \\ \text{s.t. } \forall i, j, s_{ij} \geq 0, \mathbf{s}_i^\top \mathbf{1} = 1 \\ \text{rank}(\mathbf{L}) = n - c \end{aligned} \quad (2.47)$$

where

$$\begin{aligned} \bar{d}_{ij} &= \sqrt{\frac{d_{ij}}{\|\mathbf{d}_i\|_2} \cdot \frac{d_{ij}}{\|\mathbf{d}_j\|_2}} \\ \bar{p}_{ij} &= \sqrt{\frac{p_{ij}}{\|\mathbf{p}_i\|_2} \cdot \frac{p_{ij}}{\|\mathbf{p}_j\|_2}} \end{aligned} \quad (2.48)$$

and

$$\begin{aligned} d_{ij} &= \|\mathbf{x}_i - \mathbf{x}_j\|_2^2 \\ p_{ij} &= \|\mathbf{h}(\mathbf{x}_i)\boldsymbol{\beta} - \mathbf{h}(\mathbf{x}_j)\boldsymbol{\beta}\|_2^2 \end{aligned} \tag{2.49}$$

ELM-CLR forces the similarity  $s_{ij}$  to be inversely proportional to the product of the normalized pairwise distances in the original data space and the ELM embedded space. The low-rank constraint forces the affinity matrix  $\mathbf{S}$  to be a block diagonal matrix; therefore, it naturally has the clustering property. According to Mohar's theory [78] and Fan's theory [79], the cost function of ELM-CLR can be simplified as:

$$\begin{aligned} \arg \min_S \sum_{i,j=1}^n (\bar{d}_{ij}\bar{p}_{ij}s_{ij} + \gamma s_{ij}^2) + 2\lambda \text{Tr}(\mathbf{F}^\top \mathbf{L} \mathbf{F}) \\ \text{s.t. } \forall i, j, s_{ij} \geq 0, \mathbf{s}_i^\top \mathbf{1} = 1, \mathbf{F}^\top \mathbf{F} = \mathbf{I} \end{aligned} \tag{2.50}$$

## 2.3 Machine Fault Diagnosis

This thesis studies the application of representation learning on machine fault diagnosis tasks. There is trend to provide every machine with intelligence. The intelligent machine can monitor its operating conditions in real-time. As one of the most common components in industrial machines, the rolling bearing is used as the example to monitor the conditions of industrial machines. Rolling bearings, as shown in Figure 2.5, is designed to reduce the friction between move parts. They are widely used in industrial applications, which include power plants, turbines, manufacturing, etc. Furthermore, rolling bearings are critical components in machines, their failure can cause machine shutdown that increases safety risk and causes economic loss. As reported in a survey conducted by electric power research institute (EPRI), bearing failures account for 41% of all faults in industrial applications, [80]. Therefore, it is necessary to develop a reliable system that can monitor the operating conditions in real-time.

Machine fault diagnosis aims to detect and classify failures arising in machines, which are the fundamental step of predictive maintenance (PdM) [22]. As reported by McKinsey global institute, PdM techniques can help factories reduce up to 40% maintenance cost, which is about \$630 billion per year [24]. Fault diagnosis for the rolling bearing is difficult because the behavior of bearings is affected by many variables. Typically, extracting useful features from raw signal heavily depends on expert knowledge. The designed fault diagnosis system is specifically for specific bearing applications and the operating environments. Therefore, a fault diagnosis system with the ability to adapt various bearing applications and operations conditions with less expert knowledge requirement would be highly desired.

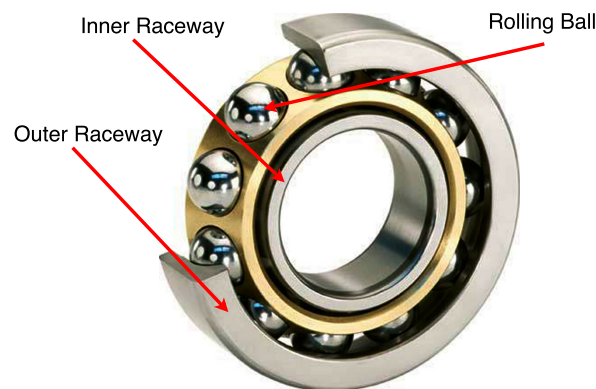


FIGURE 2.5: The structure of rolling bearing.

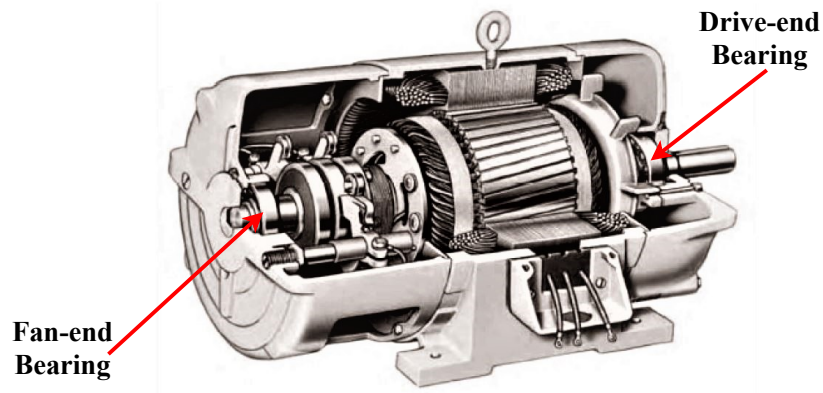




FIGURE 2.6: The structure of induction motor.

For example, the induction motor is an industrial application that is used in this thesis. Each induction motor contains two bearings at drive-end and fan-end separately, which shows in Figure 2.6. The two bearings are located in the motor shell, and it is non-visible if we do not open the motor shell. In this case, the only way to monitor the operating condition of bearings is shutting down the motor and opening the motor shell to observe bearings. However, the economy lost is vast while devices are shutting down, and it is labor-consuming to examine the bearings physically. In another situation, there are intelligent sensors mount on the shell of the induction motor. The sensor can collect the signal in real-time, and monitor the operating conditions of the bearings automatically. Furthermore, the sensor can detect an early sign to predict bearing failures. The monitoring and prediction results will send to the central system wirelessly. Based on the detected and predicted results from the sensor, we can schedule and plan the maintenance to reduce maintenance cost and machine downtime.

**Bearing Faults.** The faults of bearings can be categorized to distributed defects and localized defects [83]. Distributed defects include bearing failures from misaligned races, surface roughness, and waviness, which are usually caused by installation or manufacturing error. Localized defects include cracks, pits, and spalls on the surface of the inner race, outer race, etc. Localized defects are from the fatigue of the rolling parts. This thesis focus on detecting and isolating the faults occur on in-service machines, which assumes the manufacturing and the installation of the machine is correct. Therefore, we consider only localized defects caused by fatigue of the rolling parts. Although localized defects may happen on

TABLE 2.1: Localized faults of bearing.

Fault Type	Details	References
Inner race fault		[81]
Outer race fault		[81]
Rolling ball fault		[82]

any rolling elements, it majority occurs on inner and outer race, and whilst on the rolling ball [84]. The details of localized defects are shown in Table 2.1.

**Signal Measurements.** After a comprehensive review, it is found there are several useful diagnostic measures for fault diagnosis, which include: 1) stator current; 2) temperature; 3) sound; 4) acoustic emissions; 5) vibration measurements. The details of the signals for bearing fault diagnosis are shown in Table 2.2. The stator current may not accurately reflect the bearing fault since it is not directly relative to operating conditions of bearings. Also, the signal-to-noise ratio (SNR) of stator current measurement is low, which is hard to extract useful information. Temperature measurement requires thermal camera which is inconvenient and costly in industrial applications. Instead, collecting vibration and acoustic emission measurements are low cost, and the obtained signals are robust and directly related to bearing conditions. Hence, the vibration and acoustic emission measurements are the most commonly used signal to diagnose bearing faults. In this study, I investigate the bearing fault diagnosis by using vibration signals. An example of the vibration signal is shown in Figure 2.7.

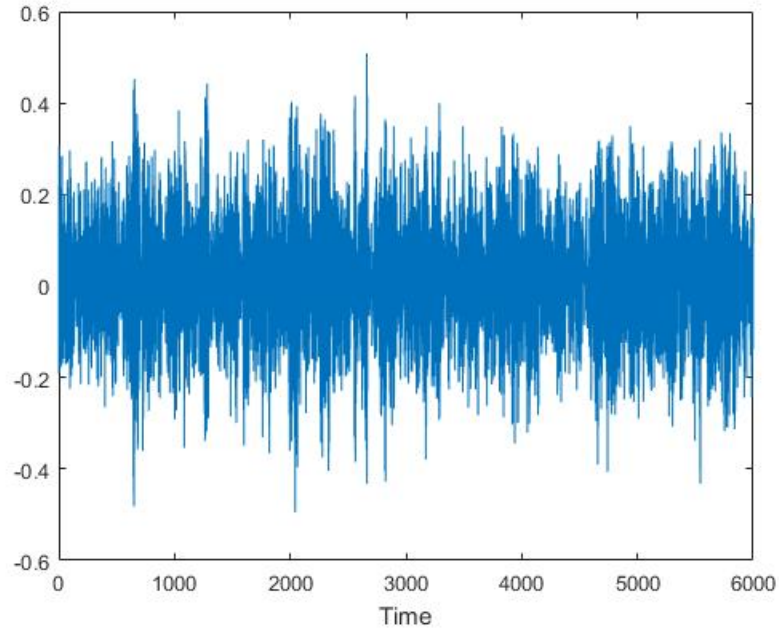


FIGURE 2.7: Vibration signal collected by accelerometer with 12kHz rate.

TABLE 2.2: Measurement methods for bearing fault diagnosis.

Signal Type	Details	References
Vibration	The most common method for fault diagnosis of rolling bearings. The vibration comes from the friction between rolling parts, while faults occur, the vibration signature usually change.	[85–91]
Acoustic emission	The acoustic emission signal measures acoustic wave emissions from the machine when a crack growth in its material.	[92–94]
Sound	The sound signal usually collected by microphones, which record the frequencies under different health conditions. The sound signal is similar to vibration signal but sensitive to background noise.	[95–98]
Temperature	The temperature of bearings changes while friction rate and rolling speed are changing. Hence, temperature measurements can be used to detect bearing faults.	[99]
Stator current	The stator current can be collected by non-intrusive method without sensors.	[100–102]



## 2.4 Summary

This chapter first introduced several geometrically motivated representation learning algorithms, including PCA, LDA, LLE, LE, and LPP. These algorithms are then summarized into a general graph embedding formulation. From the general graph embedding formulation, the common weaknesses of the existing geometrically motivated representation learning algorithms can be found. Firstly, the existing algorithms either have no models for the incoming test data or can only embed the test data linearly. Secondly, the affinity matrix used in existing algorithms is pre-defined and fixed.

This chapter also introduced a reconstruction-based representation learning algorithm, AE. AE learns a model to embed the incoming test data linearly or non-linearly. Furthermore, multiple AEs can be stacked together to form a deep structure, SAE, to learn hierarchical representations. But the existing reconstruction-based algorithms do not consider any geometry relationship among data points. Also, the training procedures of AE and SAE are time-consuming.

After that, this chapter introduced ELM, which is a neural network with an efficient training procedure. Inspired by AE, ELM-AE was introduced to investigate the data representation learning capability of ELM. Similar to conventional ELM, ELM-AE randomly generated and fixed hidden neurons, and learn the output weights analytically. Therefore, the training procedure of ELM-AE is much faster than AE, since it avoids back-propagation. Although ELM-AE reduces the Euclidean distance between data points belonging to the same cluster in the representation space, it doesn't design to preserve any geometry information from input data.

Lastly, this chapter introduced the bearing fault diagnosis as a real-world application. The performance of proposed algorithms will be tested by using common machine learning datasets and bearing fault diagnosis datasets in this thesis.

## Chapter 3

# Learning Representations with Local and Global Geometries Preserved

*Chapter 3 introduces a representation learning method with multiple cost functions to investigate the importance of preserving both local and global geometry of input data in data representations and their application for machine fault diagnosis. Section 3.1 describes the existing representation learning algorithms and their limitations. Section 3.2 describes the detail of the proposed algorithm, and Section 3.3 evaluates the effectiveness of the proposed algorithm.*

### 3.1 Background and Motivation

Machine learning methods are affected by redundant information contained in sensor data. Hence, researchers extract features from sensor data based on domain knowledge to remove the redundant information and retain the relevant information. The commonly used manual feature extraction methods in machine fault diagnosis include: 1) time domain statistical methods such as mean and standard deviation [3]; 2) wavelet transform [4]; 3) empirical model decomposition [5]. Manual feature extraction can be labor-intensive and time-consuming.

To overcome the weakness of manual feature extraction, deep learning (DL) based representational learning methods learn features automatically from raw data [18, 19, 103]. For example, restricted Boltzmann machine (RBM) learns representations by approximating the probability distribution of the original inputs. Hierarchical representations can be obtained by stacking RBMs to form a deep structure, which is known as deep belief networks (DBN). Autoencoder (AE) learns representations by forcing the reconstructed outputs equal to the original inputs [11]. Similar to DBN, stacking AEs forms stacked autoencoder (SAE) that learns hierarchical representations. Convolutional neural network (CNN) extracts features by applying a set of learnable filters on the data points [8]. However, the existing DL methods do not consider any relationship among data points. An important property of representations is to preserve the geometry of data points, i.e., the local and global geometries.

Local geometry is preserved by retaining the same relationship between a data point and its neighbors before and after representation learning. For example, local linear embedding (LLE) first reconstructs each data point from a linear combination of its neighbors and then learns representations that have the same linear relationship among data points [25]. Laplacian eigenmaps (LE) is a graph-based method that preserves the distances between data points and their neighbors while mapping them from the input data space to the representation space [26]. Although these methods can be seen as a nonlinear shallow neural network, they fail to discover deep representations [12]. In [77], manifold regularization was proposed to preserve local geometry of input data. The manifold regularization can be integrated into DL methods to learn deep representations with the local geometry preserved.

Global geometry is preserved by retaining the same relationship among all data points before and after the representation learning. For example, linear discriminant analysis rotates the axes to maximize the between-class variance and minimize the within-class variance [27]. Isometric feature mapping retains the geodesic distances among all data points during the learning process [28].

In this work, we propose a representation learning method, which is named Fast Auto-encoder with the Local and Global Penalties (FAE-LG), with the following properties:

- (a). FAE-LG learns representations directly from the un-normalized raw input data.
- (b). The learned representations preserve the local geometry of the data.
- (c). The learned representations preserve the global geometry of the data.
- (d). FAE-LG learns representations with discriminative capability efficiently.

Specifically, the first property is achieved by minimizing the error between the reconstructed data and the original input data. The second property is achieved by minimizing the distance between each data point and its nearest neighbor. The third property is achieved by minimizing the difference between the representations and the random projected input data points. The last property is achieved by minimizing the error between the predicted labels and the ground-truth. Hierarchical representations are obtained by directly stacking multiple FAE-LGs, denoted as Stacked FAE-LGs (SFAE-LG), without any additional tuning step.

## 3.2 Proposed Method

Let  $\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_n] \in \mathbb{R}^{d \times n}$  be the training set, where  $\mathbf{x}_i \in \mathbb{R}^d$  is the  $i$ -th sample and  $n$  is the number of samples. And let  $\mathbf{Y} = [\mathbf{y}_1, \mathbf{y}_2, \dots, \mathbf{y}_n] \in \mathbb{R}^{m \times n}$  be the label set, where  $\mathbf{y}_i$  is the corresponding label of  $i$ -th sample.  $\mathbf{y}_i$  is a  $m$ -dimensional one-hot row vector that only one element, which correspond to the class of  $i$ -th sample, equals to 1 and the other elements equal to 0.

### 3.2.1 Cost Functions

We learn representations by minimizing four cost functions, which are reconstruction cost, local geometry preserving cost, global geometry preserving cost and discrimination cost.

#### 3.2.1.1 Reconstruction Cost

The reconstruction cost measures the capability of the learned representations to reconstruct the input data. Inspired by tied-weight AE, the cost function is defined as the mean squared error between input data  $\mathbf{x}_i$  and the reconstructed input data  $\tilde{\mathbf{x}}_i = \beta_X \mathbf{h}_i$ , which is shown as following:

$$\frac{1}{2n} \sum_{i=1}^n \|\beta_X \mathbf{h}_i - \mathbf{x}_i\|^2 \quad (3.1)$$

The weight matrix  $\beta_X \in \mathbb{R}^{d \times l}$ , connects the input layer and the hidden layer.  $\mathbf{h}_i \in \mathbb{R}^l$  represents the output of the hidden layers with respect to sample  $\mathbf{x}_i$ , which can be calculated as following:

$$\mathbf{h}_i = g(\beta_X^\top \mathbf{x}_i) \quad (3.2)$$

where  $g(\cdot)$  is the activation function, e.g. the sigmoid function in (2.19).

### 3.2.1.2 Local Geometry Preserving Cost

The local geometry preserving cost, which is shortened as the local penalty, is defined as the mean squared error between the representations with respect to the sample  $\mathbf{x}_i$  and its nearest neighbor  $\hat{\mathbf{x}}_i$ , which is shown in (3.3).

$$\frac{1}{2n} \sum_{j=1}^n \|\mathbf{h}_i - \hat{\mathbf{h}}_i\|^2 \quad (3.3)$$

$\hat{\mathbf{h}}_i$ , which can be calculated by (3.2), represents the output of the hidden layers with respect to  $\hat{\mathbf{x}}_i$ , and  $\hat{\mathbf{x}}_i$  is the nearest neighbor of sample  $\mathbf{x}_i$  in terms of Euclidean distance. By minimizing the local penalty, the local geometry in the input space, as captured by the nearest neighbor relationship, is preserved in the learned representation space.

It is worth noting the local penalty is a special case of the manifold regularization, which aims to ensure the local smoothness of data points [77, 104]. In contrast to the original manifold regularization, the proposed local penalty used only one nearest neighbor of the data point to reduce the computation complexity.

### 3.2.1.3 Global Geometry Preserving Cost

The global geometry preserving cost is shortened as the global penalty and is defined as the mean squared error between representations and the random projected input data, which is shown in (3.4).

$$\frac{1}{2n} \sum_{i=1}^n \|\mathbf{h}_i - \mathbf{A}\mathbf{x}_i\|^2 \quad (3.4)$$

$\mathbf{A} \in \mathbb{R}^{l \times d}$  represents an orthogonal random weight matrix, where  $\mathbf{A}^\top \mathbf{A} = \mathbf{I}$  if  $d \geq l$  and  $\mathbf{A}\mathbf{A}^\top = \mathbf{I}$  if  $d < l$ .  $\mathbf{I}$  is an identity matrix. To construct the orthogonal random matrix  $\mathbf{A}$ , a singular value decomposition (SVD) is applied on a randomly generated matrix, and the unitary matrix produced by SVD can be used as the orthogonal random matrix  $\mathbf{A}$ .

Theorem 3.2.1 proves that by minimizing the global penalty with a linear activation function, the Euclidean distances among all input data are preserved in

representation space without distorting by more than a factor of  $(1 \pm \varepsilon)$  for any  $0 < \varepsilon < 1/2$ .

**Theorem 3.2.1.** The representations  $\mathbf{h} \in \mathbb{R}^l$  can be obtained by minimizing  $\sum_{i=1}^n \|\mathbf{h}_i - \mathbf{A}\mathbf{x}_i\|^2$ , where  $\mathbf{A}$  is an orthogonal random matrix. The representations  $\mathbf{h}$  with  $l \geq \frac{24}{3\varepsilon^2 - 2\varepsilon^3} \ln(n)$  have the following property:

$$(1 - \varepsilon)\|\mathbf{x}_i - \mathbf{x}_j\|^2 \leq \|\mathbf{h}_i - \mathbf{h}_j\|^2 \leq (1 + \varepsilon)\|\mathbf{x}_i - \mathbf{x}_j\|^2$$

$$\forall \quad i, j = 1, \dots, n \quad \text{s.t.} \quad i \neq j, \quad 0 < \varepsilon < 1/2$$

where  $\mathbf{h}_i = g(\beta_X^\top \mathbf{x}_i)$  and  $\mathbf{h}_j = g(\beta_X^\top \mathbf{x}_j)$  are linear function with zero bias.

*Proof.*

$$\min_{\beta_X} \sum_{i=1}^n \|\mathbf{h}_i - \mathbf{A}\mathbf{x}_i\|^2 \quad (3.5)$$

When  $\mathbf{h}_i = g(\beta_X^\top \mathbf{x}_i)$  is a linear function with zero bias, i.e.,  $\mathbf{h}_i = c\beta_X^\top \mathbf{x}_i$ , (3.5) can be written as following:

$$\min_{\beta_X} \sum_{i=1}^n \|c\beta_X^\top \mathbf{x}_i - \mathbf{A}\mathbf{x}_i\|^2 \quad (3.6)$$

where  $c$  is a scalar. There always exists a solution  $\beta_X = c^{-1}\mathbf{A}^\top$  such that (3.6) can achieve the minimum value of zero. Hence, the following relationship holds:

$$\mathbf{h}_i = \mathbf{A}\mathbf{x}_i, \quad \forall j = 1, \dots, n \quad (3.7)$$

The Johnson-Lindenstrauss lemma [105] proves that for any set with  $n$  samples, the linear map function  $f : \mathbb{R}^d \rightarrow \mathbb{R}^l$  of any pair of points  $u$  and  $v$  satisfies the following relationship with  $l \geq \frac{24}{3\varepsilon^2 - 2\varepsilon^3} \ln(n)$ , when  $0 < \varepsilon < 1$ :

$$(1 - \varepsilon)\|u - v\|^2 \leq \|f(u) - f(v)\|^2 \leq (1 + \varepsilon)\|u - v\|^2 \quad (3.8)$$

We obtain the following relationship by substituting the paired samples  $\mathbf{x}_i$  and  $\mathbf{x}_j$ , where  $j = 1, \dots, n$ , into (3.8):

$$(1 - \varepsilon)\|\mathbf{x}_i - \mathbf{x}_j\|^2 \leq \|f(\mathbf{x}_i) - f(\mathbf{x}_j)\|^2 \leq (1 + \varepsilon)\|\mathbf{x}_i - \mathbf{x}_j\|^2$$

$$\forall \quad i, j = 1, \dots, n \quad \text{s.t.} \quad i \neq j, \quad 0 < \varepsilon < 1 \quad (3.9)$$

Furthermore, it is proved that the function  $f$  in (3.8) can be an orthogonal random projection [106]. Therefore, (3.8) can be further simplified to:

$$(1 - \varepsilon)\|\mathbf{x}_i - \mathbf{x}_j\|^2 \leq \|\mathbf{A}\mathbf{x}_i - \mathbf{A}\mathbf{x}_j\|^2 \leq (1 + \varepsilon)\|\mathbf{x}_i - \mathbf{x}_j\|^2 \quad (3.10)$$

$$\forall \quad i, j = 1, \dots, n \quad \text{s.t.} \quad i \neq j, \quad 0 < \varepsilon < 1/2$$

Hence, by substituting (3.7) into (3.10), we obtain that:

$$(1 - \varepsilon)\|\mathbf{x}_i - \mathbf{x}_j\|^2 \leq \|\mathbf{h}_i - \mathbf{h}_j\|^2 \leq (1 + \varepsilon)\|\mathbf{x}_i - \mathbf{x}_j\|^2$$

$$\forall \quad i, j = 1, \dots, n \quad \text{s.t.} \quad i \neq j, \quad 0 < \varepsilon < 1/2$$

□

#### 3.2.1.4 Discrimination Cost

The discrimination cost is defined as the mean squared error between the predicted label  $\tilde{\mathbf{y}}_j = \boldsymbol{\beta}_T \mathbf{h}_j$  and the true label  $\mathbf{y}_j$ , which is shown in (3.11).

$$\frac{1}{2n} \sum_{i=1}^n \|\boldsymbol{\beta}_T \mathbf{h}_i - \mathbf{y}_i\|^2 \quad (3.11)$$

The weight matrix  $\boldsymbol{\beta}_T \in \mathbb{R}^{m \times l}$ , where  $m$  is the number of classes, denotes the weight between hidden layer and output layer. Because of the discrimination cost, the proposed algorithm does not require an additional fine-tuning step to obtain discriminative representations; therefore, the proposed method is expected to have less computational complexity, compared with SAE, which uses two-step training process.



### 3.2.2 Fast Autoencoder with The Local and Global Penalties

Based on the cost functions, we propose a fast autoencoder with local and global penalties. The mathematical formulation is shown as the follows:

$$\begin{aligned}
\underset{\beta_X, \beta_T}{\text{minimize}} \quad & \frac{\alpha_{AE}}{2n} \|\beta_X \mathbf{H} - \mathbf{X}\|_F^2 \\
& + \frac{\alpha_L}{2n} \|\mathbf{H} - \hat{\mathbf{H}}\|_F^2 \\
& + \frac{\alpha_G}{2n} \|\mathbf{H} - \mathbf{A}\mathbf{X}\|_F^2 \\
& + \frac{1}{2n} \|\beta_T \mathbf{H} - \mathbf{Y}\|_F^2 \\
& + \frac{C_X}{2} \|\beta_X\|_F^2 + \frac{C_T}{2} \|\beta_T\|_F^2
\end{aligned} \tag{3.12}$$

where  $\mathbf{H} = [\mathbf{h}_1, \mathbf{h}_2, \dots, \mathbf{h}_n]$  is the vectorized hidden output of input data  $\mathbf{X} \in \mathbb{R}^{d \times n}$ , and  $\hat{\mathbf{H}} = [\hat{\mathbf{h}}_1, \hat{\mathbf{h}}_2, \dots, \hat{\mathbf{h}}_n]$  is the vectorized hidden output of nearest neighbors  $\hat{\mathbf{X}} \in \mathbb{R}^{d \times n}$  of input data samples. As described above,  $\beta_X$  and  $\beta_T$  are output weights in reconstruction cost and discrimination cost, respectively.  $\mathbf{A}$  is an orthogonal random weight matrix that randomly mapping input samples into a random space.  $\mathbf{Y} \in \mathbb{R}^{m \times n}$  is the one-hot label set that corresponds to the input data  $\mathbf{X}$ . In addition,  $\alpha_{AE}$ ,  $\alpha_L$ ,  $\alpha_G$ ,  $C_X$  and  $C_T$  are the hyper-parameters determined by users. The values of hyper-parameters determine the importance of each term in the objective function.

We find the optimal solution by using alternating minimization method, which alternatively minimizes the objective function with respect to one variable while fixing the others.

In each iteration, we first minimize the objective function with respect to  $\beta_X$ . We use the Limited memory Broyden Fletcher Goldfarb Shanno (L-BFGS) algorithm<sup>1</sup> [107] from the software package *minfunc* [108] to update the values of  $\beta_X$ .

---

<sup>1</sup>L-BFGS belongs to the family of quasi-Newton methods, which uses the most recent gradients to approximate the inverse Hessian matrix.

**Algorithm 1** SFAE-LG learning algorithm

---

**Input:** the training data  $\{\mathbf{X}, \mathbf{Y}\} = \{\mathbf{x}_i, \mathbf{y}_i\}_{i=1}^n$ , the number of hidden layers  $q$ , the size of each hidden layer  $\{l^p\}_{p=1}^q$ , the maximum iteration number  $max\_iter$

**Hyper-parameters:**  $\alpha_{AE}$ ,  $\alpha_G$ ,  $\alpha_L$ ,  $C_X$  and  $C_T$

**Output:** the weight of each hidden layer  $\{\beta_X^p\}_{p=1}^q$ , the output of each hidden layer  $\{\mathbf{H}^p\}_{p=1}^q$ , where  $\mathbf{H}^p = [(\mathbf{h}_1^p)^\top, (\mathbf{h}_2^p)^\top, \dots, (\mathbf{h}_n^p)^\top]^\top$

- 1: **for**  $p \leftarrow 1, q$  **do**
- 2:   Construct an FAE-LG network with  $l^p$  hidden neurons
- 3:   Initiate  $\beta_X^p$  and  $\beta_T^p$  with random values
- 4:   Obtain the nearest neighbors  $\hat{\mathbf{X}}$  of  $\mathbf{X}$
- 5:   Generate an orthogonal random  $\mathbf{A}$
- 6:    $iter \leftarrow 1$ , where  $iter$  is the count of iterations
- 7:   **while**  $iter < max\_iter$  **do**
- 8:     Compute  $\beta_X^p$  using L-BFGS [107]
- 9:     Compute  $\beta_T^p$  using (3.15)
- 10:     $iter \leftarrow iter + 1$
- 11:   **end while**
- 12:   Construct the  $p$ -th hidden layer of SFAE-LG with  $\beta_X^p$
- 13:   Compute  $\mathbf{H}^p$  using (3.2)
- 14:    $\mathbf{X} \leftarrow \mathbf{H}^p$
- 15: **end for**
- 16: **return**  $\{\beta_X^p\}_{p=1}^q$

---

The partial derivatives of  $\frac{\partial E}{\partial \beta_X}$  with respect to  $\beta_X$  is calculated by:

$$\begin{aligned}
\frac{\partial E}{\partial \beta_X} = & \alpha_{AE} \left( \mathbf{H} \circ (\mathbf{1} - \mathbf{H}) \circ (\beta_X^\top \beta_X \mathbf{H} - \beta_X^\top \mathbf{X}) \right) \mathbf{X}^\top \\
& + \alpha_{AE} \mathbf{H} (\beta_X \mathbf{H} - \mathbf{X})^\top \\
& + \left( \mathbf{H} \circ (\mathbf{1} - \mathbf{H}) \circ (\beta_T^\top \beta_T \mathbf{H} - \beta_T^\top \mathbf{Y}) \right) \mathbf{X}^\top \\
& + \alpha_G \left( \mathbf{H} \circ (\mathbf{1} - \mathbf{H}) \circ (\mathbf{A} \mathbf{A}^\top \mathbf{H} - \mathbf{A} \mathbf{X}) \right) \mathbf{X}^\top \\
& + \alpha_L \left( (\mathbf{H} - \hat{\mathbf{H}}) \circ \mathbf{H} \circ (\mathbf{1} - \mathbf{H}) \right) \mathbf{X}^\top \\
& - \alpha_L \left( (\mathbf{H} - \hat{\mathbf{H}}) \circ \hat{\mathbf{H}} \circ (\mathbf{1} - \hat{\mathbf{H}}) \right) \hat{\mathbf{X}}^\top \\
& + C_X \beta_X
\end{aligned} \tag{3.13}$$

where “ $\circ$ ” means element-wise product and the matrix  $\hat{\mathbf{X}} = [\hat{\mathbf{x}}_1, \hat{\mathbf{x}}_2, \dots, \hat{\mathbf{x}}_n]$ . To simplify the algorithm,  $\beta_X$  is updated once in each iteration.

Then, we minimize the objective function with respect to  $\beta_T$ . When  $\beta_X$  is fixed, the optimization problem with respect to  $\beta_T$  is convex and has global optimal

solution, which can be calculated analytically by solving  $\frac{\partial E}{\partial \beta_T} = 0$ :

$$\frac{\partial E}{\partial \beta_T} = \mathbf{H}(\beta_T \mathbf{H} - \mathbf{Y})^\top + C_T \beta_T^\top = 0 \quad (3.14)$$

Hence,

$$\begin{aligned} C_T \beta_T^\top &= -\mathbf{H}(\beta_T \mathbf{H} - \mathbf{Y})^\top \\ \mathbf{H} \mathbf{Y}^\top &= (\mathbf{H} \mathbf{H}^\top + C_T \mathbf{I}) \beta_T^\top \\ \beta_T^\top &= (\mathbf{H} \mathbf{H}^\top + C_T \mathbf{I})^{-1} \mathbf{H} \mathbf{Y}^\top \end{aligned} \quad (3.15)$$

Similar to SAE in Figure 2.2, SFAE-LG is trained in the greedy layer-wise approach. The first FAE-LG learns weight matrix  $\beta_X^1$  of the first hidden layer by using  $\mathbf{X}$  as the input. After that, the output of the first hidden layer  $\mathbf{H}^1$  is used as the input of the second FAE-LG to learn  $\beta_X^2$ . Generally, the  $p$ -th FAE-LG uses  $\mathbf{H}^{p-1}$  as the input to learn  $\beta_X^p$ . The details are summarized in Algorithm 1. Moreover, a linear regression layer is stacked after the last hidden layer to predict the label  $\tilde{\mathbf{T}}$ .

### 3.3 Experiments

The proposed representation learning method, SFAE-LG, was tested on two public benchmark datasets to validate its effectiveness and efficiency with comparison to the state-of-the-art related methods.

#### 3.3.1 Datasets Descriptions

**Bearing Dataset [109]** The CWRU bearing dataset contains the vibrations signal of motor bearings provided by Case Western Reserve University. As shown in Figure 3.1, the magnetic-based accelerometer is mounted on the housing of the test motor. The vibrations signals were collected by the accelerometer with 12 kHz sampling frequency, and under four different health conditions of the bearing: 1) normal condition (N), 2) roller fault (RF), 3) outer raceway fault (ORF), and 4) inner raceway fault (IRF). Also, each health condition of the bearing includes three different severity levels, which are 0.18, 0.36, and 0.53 mm cracks. Hence, there are a total of ten different classes in this dataset, and the details are shown in Table 3.1. The load motor provides four different load conditions, which are 0, 1, 2, and 3 hp, and the vibration signals are repeatedly collected under each load condition. In this study, to increase the generality, we used the data collected from all load conditions to test the proposed method, i.e., the data collected under the same health condition but different load condition is treated as the same class. In this experiment, we created 2400 samples for each conditions, where the dimension of each sample is 200.

**IMS Bearing Dataset [110]** Three run-to-failure tests produced different bearing faults: 1) inner race failure; 2) outer race failure; 3) roller failure. The vibration signals in 1-second duration were collected every 10 min with 20kHz sampling rate. Each run-to-failure test could be divided into three stages that are the normal stage, degraded stage, and failure stage [111]. The details of different stages are shown in Figure 3.2. Hence, we created a classification problem with seven classes that including normal (N), degraded roller (DR), roller failure (RF), degraded inner raceway (DIR), inner race failure (IRF), degraded outer raceway (DOR), outer raceway failure (ORF). This experiment not only defines the health conditions but

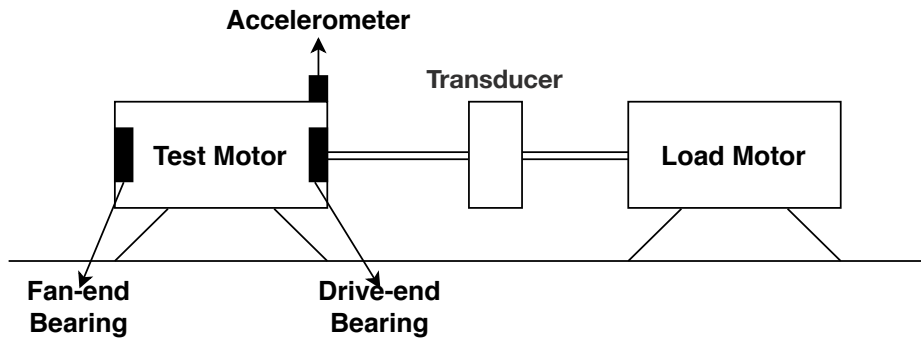
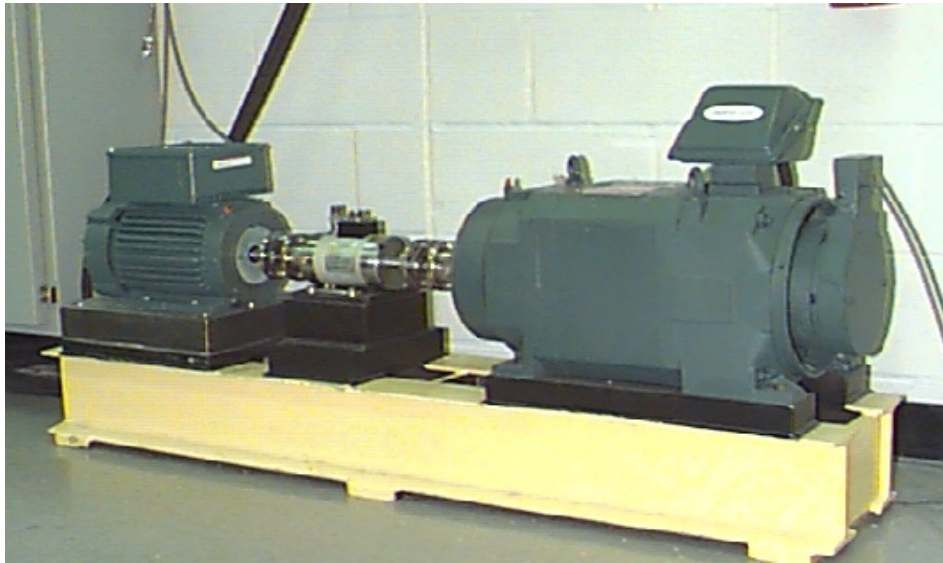


FIGURE 3.1: The test rig of CWRU bearing dataset.

TABLE 3.1: Descriptions of bearing conditions in CWRU dataset.

Health Conditions		Description
N	Normal	Bearing operates in good condition
RF1	Roller fault	Roller has 0.18 mm crack
RF2	Roller fault	Roller has 0.36 mm crack
RF3	Roller fault	Roller has 0.53 mm crack
ORF1	Outer raceway fault	Outer raceway has 0.18 mm crack
ORF2	Outer raceway fault	Outer raceway has 0.36 mm crack
ORF3	Outer raceway fault	Outer raceway has 0.53 mm crack
IRF1	Inner raceway fault	Inner raceway has 0.18 mm crack
IRF2	Inner raceway fault	Inner raceway has 0.36 mm crack
IRF3	Inner raceway fault	Inner raceway has 0.53 mm crack

also detects the failures in the early stage. The dataset includes 291520 samples, and the dimension of each sample is 256. The details of each class are listed in Table 3.2.

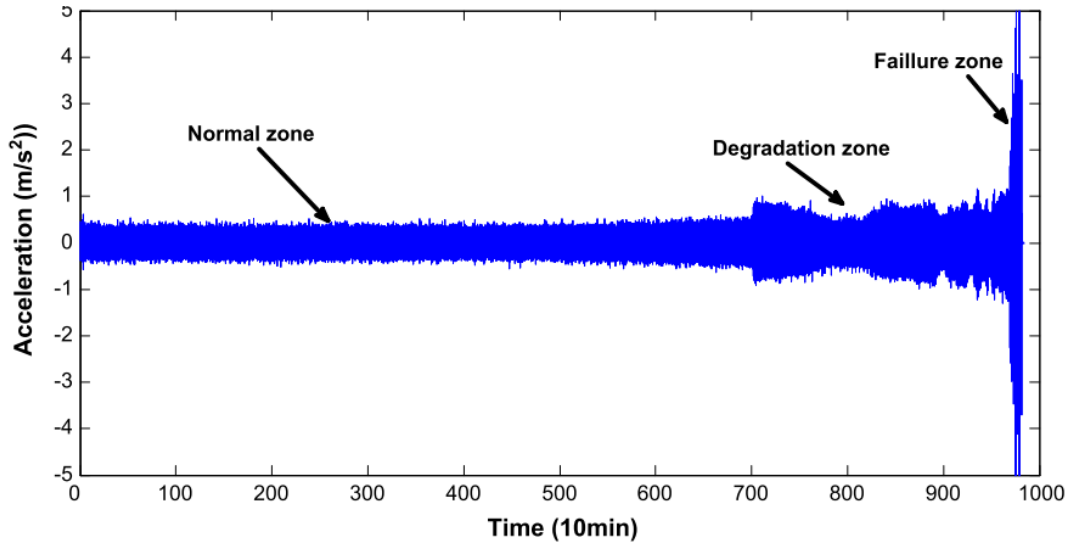


FIGURE 3.2: Run-to-failure vibration signal with outer race failure in IMS bearing dataset.

TABLE 3.2: Number of samples in each class in IMS bearing dataset.

Classes	N	DR	RF	DIR	IRF	DOR	ORF
Sample #	264000	8000	8000	8000	800	1600	1120
Label	1	2	3	4	5	6	7

TABLE 3.3: Hyper-parameters selection range for cross-validation.

Methods	Hyper-parameters	Range
	Neuron numbers	300 - 2000
SAE/SDA	learning_rate	0.001 - 1
	corrupt_rate	0.05 - 0.8
	sparsity	0.05 - 0.8
SFAE-LG	$\alpha_{AE}$	0 - 1e10
	$\alpha_G$	0 - 1e10
	$\alpha_L$	0 - 1e10
	$C_X$	1e-10 - 1e10
	$C_T$	1e-10 - 1e10
SVM	$C$	1e-10 - 1e10
	Gamma	1e-10 - 1e10
Random Forest	# of trees	50 - 1000
ELM	$C$	1e-10 - 1e10

### 3.3.2 Experimental Setups

Three experiments were designed to evaluate the proposed method. In all experiments, the training and test data were divided by using 5-fold cross-validation. Specifically, 20% of dataset is used as training data and the other 80 % is used as test data.

**Evaluation of Local and Global Penalties** This experiment aimed to analyze the effects of local and global penalties of SFAE-LG. Besides the proposed SFAE-LG, we implemented and compared three other formulations with one or both of the local and global penalties remove:

- SFAE without any penalties (SFAE)
- SFAE with local penalty (SFAE-L)
- SFAE with global penalty (SFAE-G)
- SFAE with both local and global penalties (SFAE-LG)

**Evaluation of Computational Efficiency** We hypothesized the discrimination cost (3.11) can reduce the computational complexity of the proposed method. Hence, this experiment compared training and test time between the proposed method and the other AE-based methods in two settings: 1) using the selected architectures, i.e., architectures that produce the best performance in cross-validation; 2) using the same architecture with the proposed method. To elaborate the results clearly, we also reported the ratio of other methods to SFAE-LG. The ratio ( $R$ ) was calculated by:

$$R = \frac{\text{Training/Test time of AE-based method}}{\text{Training/Test time of SFAE-LG}} \quad (3.16)$$

**Evaluation of Noise Robustness** This experiment aimed to evaluate the noise robustness of the proposed method. In [21], the noise robustness of SAE and SDA has been proved on the machine fault diagnosis task with four health conditions. Hence, we used SAE and SDA as the baseline to evaluate the noise robustness of SFAE-LG with the same experiment settings. This experiment was conducted

on noisy CWRU dataset, and the noises were injected into the test data with nine different signal-to-noise-ratio (SNR) values. By injecting noises to test data instead of training data, this experiment evaluated the models' capability of adapting to environments with changing noise levels.

**Comparison with Related Methods** This experiment aimed to evaluate the effectiveness of the proposed methods for machine fault diagnosis in comparison with the related methods on both CWRU and IMS bearing datasets. We implemented three state-of-the-art DL methods that have been applied to machine fault diagnosis, i.e., SAE [21, 54], stacked denoising autoencoder (SDA) [21, 58], and CNN [103, 112]. Besides DL methods, we also implemented some state-of-the-art shallow classifiers, i.e., random forest [113], support vector machine (SVM) [114] and extreme learning machine (ELM) [35].

All DL methods were implemented with four hidden layers. The shallow classifiers were applied on the manually extracted features, where nine statistical features in the time domain and six features in the time-frequency domain were extracted [90]. All of the hyper-parameters were selected by 4-fold cross-validation with grid searching method. Specifically, we randomly divided the training data into four parts and used each part as the training data accordingly with the other three parts as the validation data. The final values of the hyper-parameters were selected as the value corresponding to the highest mean accuracy as defined in the next paragraph. The ranges of the possible values of all hyper-parameters were listed in Table 3.3. The key parameters of the SFAE-LG are given in Table 3.4.

TABLE 3.4: Hyper-parameters of SFAE-LG model.

Hyper-parameters	Values
Hidden neurons	[700, 150, 700, 300]
$\alpha_{AE}$	[0.06, 3e-4, 1.48, 1e-4]
$\alpha_G$	[1.43, 3e-10, 0.13, 1e-4]
$\alpha_L$	[0.02, 3e-4, 1, 1e-4]
$C_X$	[3e-4, 1e-4, 1.02e-3, 1.2e-3]
$C_T$	[1e-9, 1e-6, 1e-9, 1e-5]

Moreover, the experiments results were evaluated in terms of the following metrics: accuracy ( $Acc$ ), precision ( $Pre$ ), recall ( $Rec$ ) and f-score ( $f$ ). The accuracy [115] is the standard evaluation metric for classification problem. Also, since IMS dataset



is class imbalanced as shown in Table 3.2, this work also used precision ( $Pre$ ), recall ( $Rec$ ) and f-score ( $f$ ) to evaluate the results. The precision [115] is related to the correctly classified samples, i.e., true positives (TP), and samples misclassified as positives, i.e., false positives (FP). The recall [115] is related to TP and the misclassified samples, i.e., false negatives (FN). The details are shown as following:

$$Pre = \frac{1}{M} \sum_{m=1}^M \frac{TP_m}{TP_m + FP_m} \quad (3.17)$$

$$Rec = \frac{1}{M} \sum_{m=1}^M \frac{TP_m}{TP_m + FN_m} \quad (3.18)$$

where  $TP_m$ ,  $FP_m$  and  $FN_m$  are TP, FP and FN of class  $m$  respectively. The f-score combines precision and recall, which is shown as following [115].

$$f = 2 \frac{Pre \times Rec}{Pre + Rec} \quad (3.19)$$

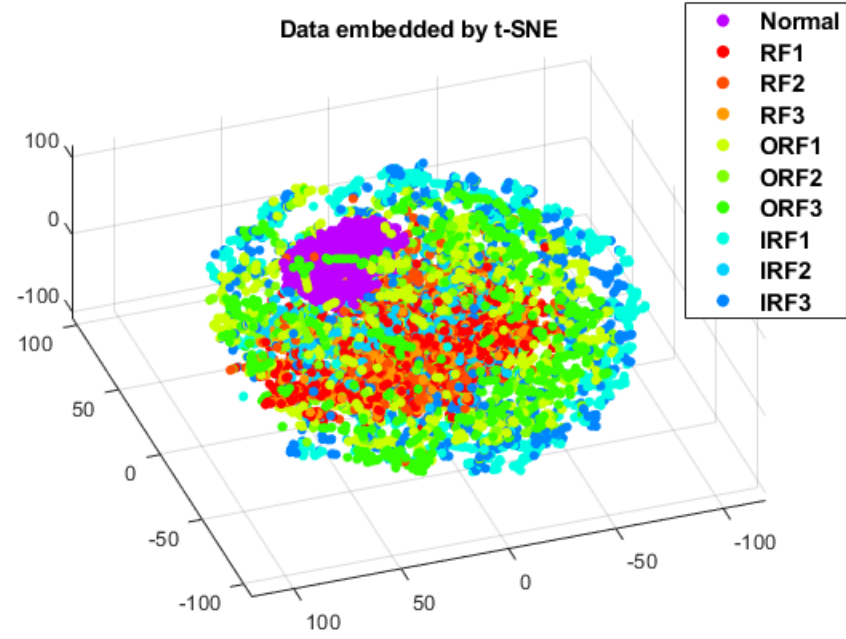
### 3.3.3 Evaluation of Local and Global Penalties

Table 3.5 shows the classification accuracy of alternative formulations with/without local and global penalties. By adding a local penalty, SFAE-L improved the classification accuracy from 94.61% to 95.41%. By adding a global penalty, SFAE-G achieved 96.94% classification accuracy compared with 94.61% achieved by SFAE. Furthermore, SFAE-LG, which includes both of the local and global penalties, achieved 97.29% classification accuracy. Hence, we conclude that preserving both local and global geometries in representations are beneficial for machine fault diagnosis.

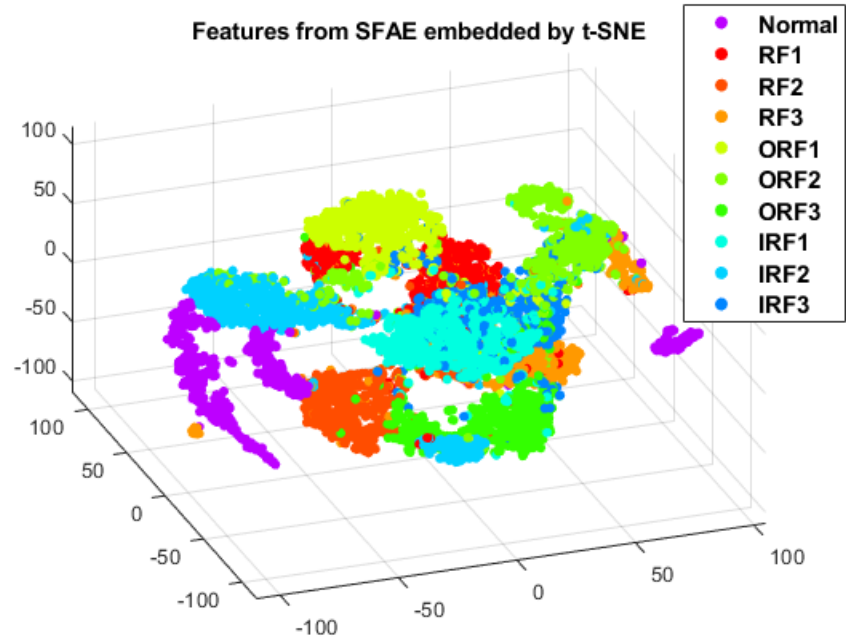
TABLE 3.5: Evaluation of local and global geometries for SFAE-LG with four hidden layers.

Settings	SFAE	SFAE-L	SFAE-G	SFAE-LG
Accuracy (%)	94.61	95.41	96.94	97.29

To investigate the effects of the local and global penalties of SFAE-LG, we reduced the dimensions of the learned representations using t-Distributed Stochastic



(A) Original data



(B) SFAE

FIGURE 3.3: Scatter plots of the learned data representations of the CWRU bearing dataset by using t-SNE. (a) Original data. (b) SFAE.

Neighbor Embedding<sup>2</sup> (t-SNE) [116] and visualized the results in Figure 3.3, Figure 3.4, and Figure 3.5. We observed that the data points of the same fault type are closer than those of the different fault types in the input data space, as shown in Figure 3.3a. For example, the data points of the inner race fault (IRF1, IRF2, and IRF3) are all located on the external surface, and the data points of the roller fault (RF1, RF2, and RF3) are concentrated in the center. It is desired to remain the same global distribution of data points from the input data space to the representation space.

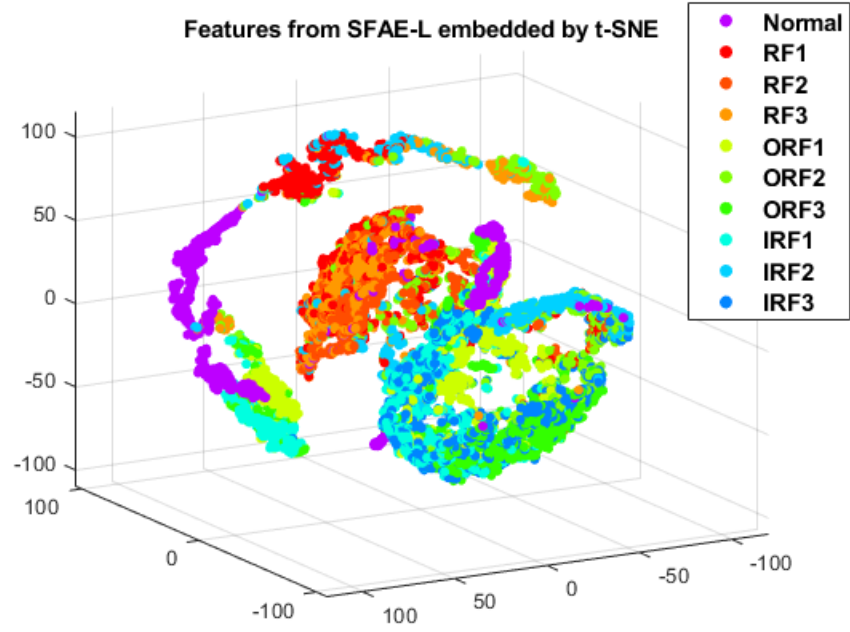
The representations learned by SFAE (without local and global penalties) failed to remain the distribution of the data points. For example, the data points of normal condition are separated to three parts in Figure 3.3b, and all classes were mixed regardless of the fault types. On the contrary, the representations learned by SFAE-G and SFAE-LG retained the same distribution of data points as the input data space. For example, the representations of SFAE-G are shown in Figure 3.4b: 1) IRF1, IRF2, and IRF3 concentrated in the left, 2) RF1, RF2, and RF3 concentrated in the right, 3) ORF1, ORF2 and ORF3 concentrated in the center, and 4) N concentrated in the bottom. The representations of SFAE-LG are shown in Figure 3.5a and Figure 3.5b with two different view angles, and also shows similar observation with Figure 3.4b. Therefore, we conclude that the global penalty contributes to the preservation of the geometry of all data points.

Moreover, the learned representations should have small within-class variance to improve the classification accuracy. The representations learned by SFAE-LG, as shown in Figure 3.5a and Figure 3.5b, showed smaller within-class variance compared to the other three formulations: 1) SFAE, as shown in Figure 3.3b, 2) SFAE-L, as shown in Figure 3.4a, and 3) SFAE-G, as shown in Figure 3.4b. Hence, we concluded that using both local and global penalty can reduce the within-class variance of learned representations.

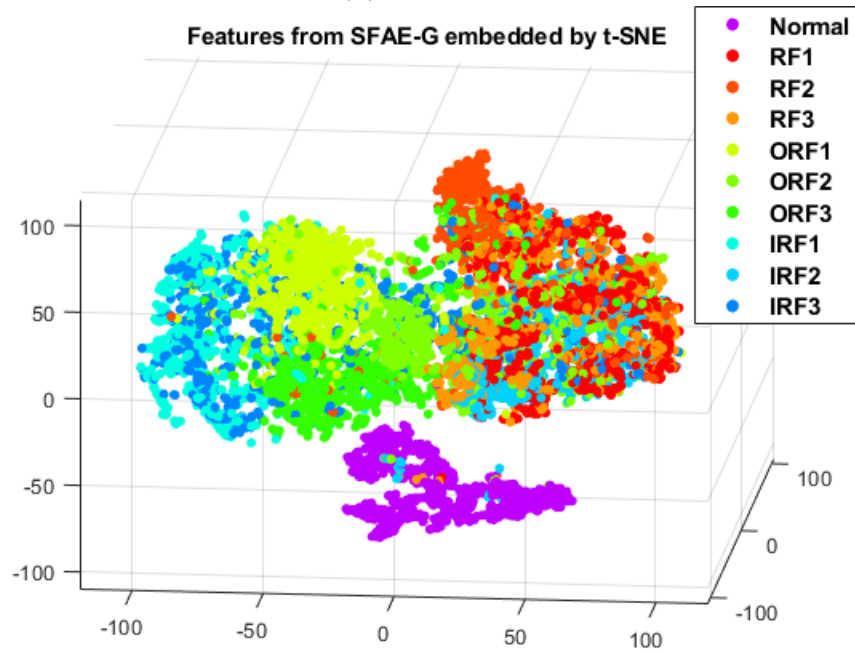
### 3.3.4 Evaluation of Computational Efficiency

Table 3.6 presents the number of hidden neurons selected by SAE, SDA and SFAE-LG based on validation classification accuracy. SFAE-LG required a much smaller

<sup>2</sup>t-SNE is a non-linear dimensionality reduction method by minimizing the divergence between two distributions: a distribution describes the similarity of points in high-dimensional space and a distribution describes the similarity in low-dimensional space.

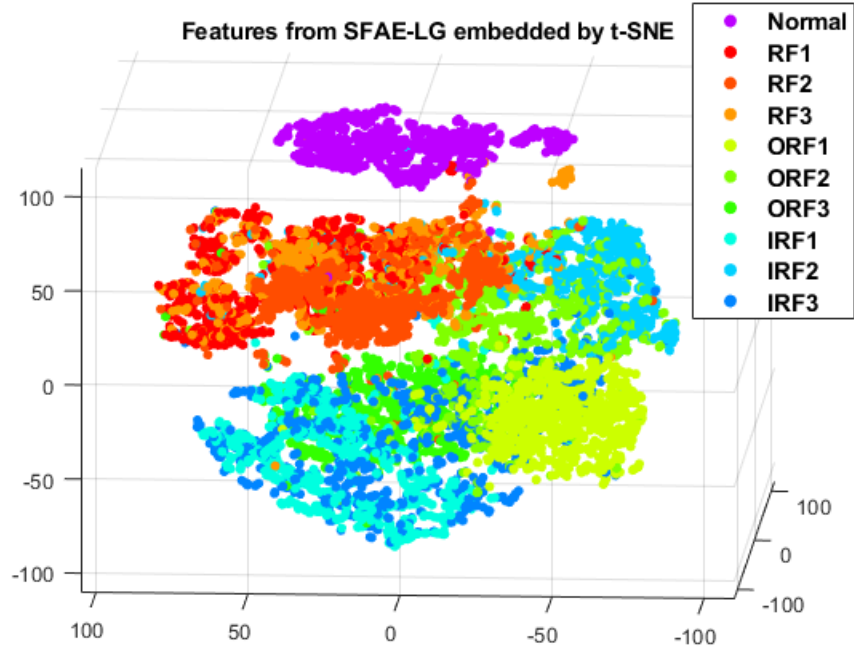


(A) SFAE-L

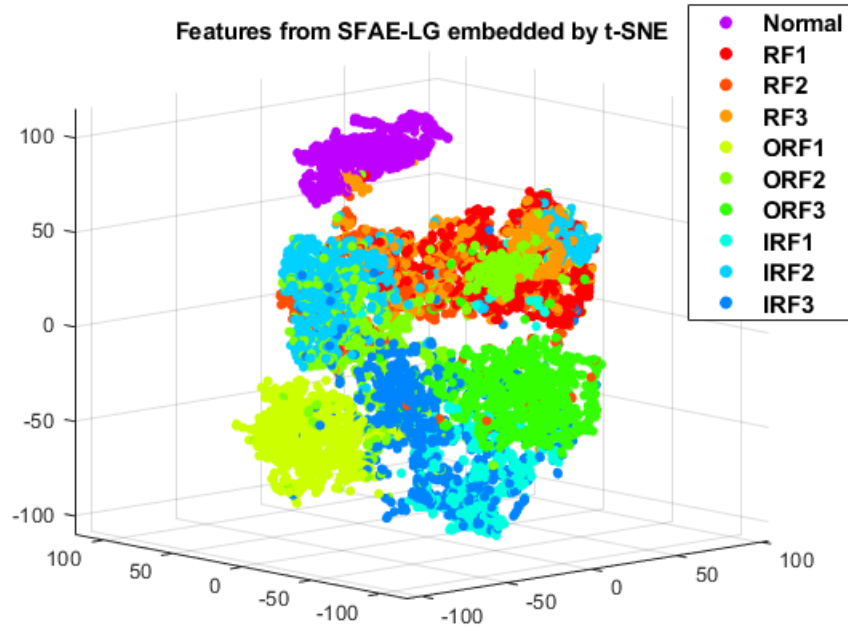


(B) SFAE-G

FIGURE 3.4: Scatter plots of the learned data representations of the CWRU bearing dataset by using t-SNE. (a) SFAE-L. (b) SFAE-G.



(A) SFAE-LG



(B) SFAE-LG

FIGURE 3.5: Scatter plots of the learned data representations of the CWRU bearing dataset by using t-SNE. (a) and (b) show representations learned by SFAE-LG in different angles.

number of hidden neurons than the other two methods. In the cross-validation, SFAE-LG selected the number of neurons in each layer independently via discrimination cost. SAE and SDA selected the number of neurons of the whole network by using fine-tuning. We hypothesize that the proposed method can select a smaller network than SAE and SDA because of the discrimination cost used in the training process of each hidden layer. It is worth noting that the smaller number of hidden neurons can bring forth a faster test time. For example, the test time of SAE, SDA and SFAE-LG are 0.6672s, 1.3349s, and 0.3256s respectively. Hence, it is meaningful to investigate the relationship between the selected number of hidden neurons and discrimination cost in future work. Table 3.7 shows the training time

TABLE 3.6: The comparison of hidden neurons used by AE-based methods.

# of neurons	SAE	SDA	SFAE-LG
Hidden layer 1	800	300	700
Hidden layer 2	700	500	150
Hidden layer 3	400	2000	700
Hidden layer 4	400	400	300
Total # of parameters	1164000	2014000	<b>563000</b>

of SFAE-LG compared with the other AE-based methods. The training time of the proposed method is 32 to 98 times shorter than SAE and SDA when all methods used the architectures with the best performance, and 15 to 23 times shorter when all methods used the same network architecture. FAE-LG uses a one-step training process with shorter processing time than the two-step training process used by SAE.

TABLE 3.7: The comparison of training time on CWRU bearing dataset.

Methods	Training time (with the best performance)		Training time (with the same architecture)	
	Second (s)	Ratio	Second (s)	Ratio
2-layers	SAE	7576	92	1813
	SDA	5845	71	1948
	SFAE-LG	<b>82</b>	<b>1</b>	<b>82</b>
3-layers	SAE	10253	43	3555
	SDA	23301	98	3883
	SFAE-LG	<b>237</b>	<b>1</b>	<b>237</b>
4-layers	SAE	13875	32	6771
	SDA	28595	65	7149
	SFAE-LG	<b>438</b>	<b>1</b>	<b>438</b>

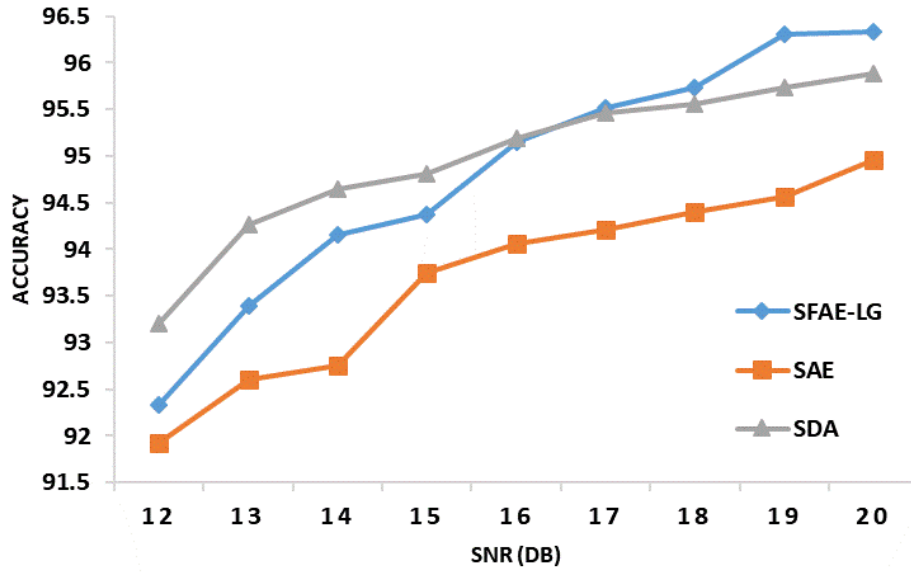


FIGURE 3.6: The classification accuracies of SAE, SDA and SFAE-LG with different SNRs (in dB).

### 3.3.5 Evaluation of Noise Robustness

Figure 3.6 shows the classification accuracies of SAE, SDA and the proposed method on noisy test data with nine different SNR values. The performances of all methods decreased when the power of noise increased, i.e., the value of SNR decreased. It is noticed that the proposed method achieved the best performance when the noise is moderate (SNR higher than 16 dB). Therefore, the proposed method can adapt to a moderate change of the environment, i.e., the SNR of noises higher than 16 dB. When the noise is strong (SNR lesser than 16 dB), although SFAE-LG underperformed SDA, which is a specially designed method with de-noising capability, it still performed better than SAE. It is important to improve the noise robustness of the proposed method in future work.

### 3.3.6 Comparison with Related Methods

TABLE 3.8: The comparison of classification accuracy ( $Acc$ ), precision ( $Pre$ ), recall ( $Rec$ ) and f-score ( $f$ ) on CWRU bearing dataset.

Methods	$Acc$ (%)	$Pre$ (%)	$Rec$ (%)	$f$ -score(%)
SVM	91.73	91.74	91.73	91.73
Random Forest	87.06	88.14	87.06	87.59
ELM	93.65	93.68	93.65	93.66
SAE	95.42	95.43	95.42	95.42
SDA	95.86	95.86	95.86	95.86
CNN	95.87	95.88	95.87	95.87
SFAE-LG	<b>97.29</b>	<b>97.34</b>	<b>97.29</b>	<b>97.31</b>

TABLE 3.9: The comparison of classification accuracy ( $Acc$ ), precision ( $Pre$ ), recall ( $Rec$ ) and f-score ( $f$ ) on IMS bearing dataset.

Methods	$Acc$ (%)	$Pre$ (%)	$Rec$ (%)	$f$ -score(%)
SVM	96.48	90.54	75.15	80.00
Random Forest	96.42	90.29	73.99	78.99
ELM	96.70	90.06	76.56	81.48
SAE	96.92	91.56	76.98	82.09
SDA	97.01	91.89	77.69	82.77
CNN	97.31	92.57	79.99	84.84
SFAE-LG	<b>97.84</b>	<b>94.09</b>	<b>84.54</b>	<b>88.57</b>

Table 3.8 shows a comprehensive comparison in terms of classification accuracy, precision, recall, and f-score on CWRU bearing dataset and Table 3.9 shows the



comparison on IMS bearing dataset. It can be observed that the DL methods, i.e., SAE, SDA, CNN, and SFAE-LG, performed better than the shallow classifiers including SVM, random forest, and ELM. Hence, the importance of deep representations has been verified. Among the DL methods, the proposed method achieved 97.31% f-score on CWRU dataset, and outperformed the other methods. On IMS dataset, the proposed method also outperformed the other methods with 88.57% f-score. Therefore, the proposed method with the local and global penalty is beneficial in machine fault diagnosis tasks.

TABLE 3.10: The comparison of classification accuracy among published performances.

Dataset	Methods	Sample dimensions	Input dimensions	# of traing samples (%)	# of health condition	Acc (%)
CWRU	[117]	2048	33	75	10	88.9
	[118]	2400	5	40	11	97.91
	[21]	200	200	40	4	95.58
	[119]	300	300	70	4	93.07
	SFAE-LG	200	200	20	10	97.29
IMS	[111]	20480	18	75	7	93
	SFAE-LG	256	256	20	7	97.84

While Table 3.8 and Table 3.9 compare the performance between the proposed method and the other benchmark methods by using the same experimental settings, Table 3.10 reports the performances of the related methods published in their original papers. In [117] and [118], SVM was used to classify machine health conditions with different manually extracted features: the wavelet leaders multi-fractal features [117] and the combination of permutation entropy and ensemble empirical mode decomposition [118]. They obtained the accuracy of 88.9% and 97.91% respectively on the CWRU dataset. Furthermore, the input dimension of the training data was decreased by using AE-based methods. SAE and SDA were applied to CWRU dataset in [119] and [21] respectively. When the length of the input data is 300, SAE obtained the accuracy of 93.07% in [119]. SDA obtained the accuracy of 95.58% in [21] when the length of the input data reduces to 200. In [111], empirical mode decomposition and artificial neural network were applied on IMS dataset to classify seven health conditions, and obtained the accuracy of 93%. From Table 3.10, the proposed method obtained a comparable result with the state-of-the-art methods by using lesser training samples and smaller input dimensions of the training data.

### 3.4 Summary

This chapter describes the first work that addresses the research objectives of this thesis, i.e., to efficiently learn data representations that can improve the performance of machine fault diagnosis tasks, and to exploit and preserve geometry information of input data while learning data representations. Section 3.1 reviews existing deep learning-based representation learning algorithms and other algorithms that preserve local and global geometry. Section 3.3 introduces a representational learning method, which is called FAE-LG, and its deep structure, SFAE-LG. Section 3.3 analyzes the sensitivity of hyper-parameters and experimentally demonstrates the effectiveness of FAE-LG on machine fault diagnosis tasks. Specifically, Compared to other deep learning-based algorithms, the proposed algorithm simultaneously preserves the local and global geometries of input data, and the importance of preserving both geometries have been verified in Section 3.3. Furthermore, in contrast to the two-step training process used by most of the deep learning-based algorithms, the proposed algorithm completes training in one step, which significantly reduces the training time. Moreover, the discrimination cost of the proposed algorithm reduces the number of neurons required in hidden layers, which reduces the test time compared to other algorithms. On two benchmark datasets, the proposed algorithm outperforms all other comparison algorithms, i.e., CNN, SAE, and SDA, in terms of both classification accuracy and f-score. Therefore, it is proved an efficient tool to provide accurate information about the machine conditions, which can be used to assist maintenance planning to save maintenance costs.



## Chapter 4

# Learning Local Discriminative Representations via Extreme Learning Machine

*In order to improve the efficiency of representation learning algorithms and retain the performances on machine fault diagnosis tasks, Chapter 4 introduces an ELM-based algorithm that preserves the local geometry of input data and exploits the local discrimination information in data representations. Section 4.1 reviews ELM and its variants, especially the ELM-based representation learning algorithms with geometry information exploited, and Section 4.2 introduces the details of the proposed LDELM-AE. Section 4.3 experimentally evaluates the effectiveness and efficiency of the proposed algorithm.*

## 4.1 Background and Motivation

Extreme learning machine (ELM) [31, 32] is a efficient and effective single-hidden layer feedforward network (SLFN). The key idea of ELM is to randomly generate the weights between the input layer and the hidden layer, and analytically calculate the output weights by using Moore-Penrose generalized inverse. In contrast to traditional training approaches of SLFNs, ELM achieves much faster training speed and maintains the universal approximation capability by using fixed hidden neurons and tunable output weights [33, 120]. It is able to solve variant tasks including regression [35], classification [121], clustering [122], and etc. ELM has been successfully applied in many applications, e.g., facial expression [123], image classification [124], taste recognition [125], video anomaly detection [126], etc. Moreover, ELM has been extended to learn data representations. Kasun et al. [13, 14] proposed ELM autoencoder (ELM-AE) to map the input data into the representation space. They also introduced a multilayer ELM autoencoder (ML-ELM) by stacking multiple ELM-AEs to learn hierarchical representations. In contrast to other deep learning algorithms, ML-ELM model can be trained without iterations and fine-tuning. Hence, ML-ELM can learn hierarchical representations efficiently. D. Cui et al. [39] proposed the extreme learning machine network (ELMNet) based stacked ELM-AE on image patches to learn data representations and achieved a good result on the handwritten dataset. However, the above algorithms do not consider the geometry information of data points, which is proved an important property of data representations in Chapter 3.

Recently, preserving the local geometry of input data points have been proved as an essential property of representations. Preserving the local geometry aims to retain the same relationship, e.g., the Euclidean distance, between a data point and its neighbors before and after map them into the representation space. Hence, it increases the within-class compactness of the learned representations. Laplacian eigenmaps (LE) [26] is a graph-based algorithm aims to preserve local geometry by minimizing the Euclidean distances between data points and their neighbors in the representation space. Local linear embedding (LLE) preserves local geometry by retaining the same linear relationships of each data point and its neighbors. More specifically, each data point can be reconstructed by the same linear combination of its neighbors before and after map the data point into representation space. Although LE and LLE can preserve the local geometry in the representation space,

they only process the training data points, and it is hard to apply them on the new test points. Therefore, X. He et al. [49] introduced the locality preserving projection (LPP) to preserve the local geometry, i.e., neighborhood structure, of the input data. As similar as LE, LPP also minimizes the Euclidean distances between each data point and its neighbors in the representation space. Furthermore, LPP maps data points to representations by using a linear transformation matrix that can be easily applied to the new test points, but LPP does not investigate the non-linear representation mapping.

To efficiently learn representations that preserve the local geometry of input data, various extensions of ELM-AE and ML-ELM were proposed. K. Sun et al. [74] introduced the generalized extreme learning machine autoencoder (GELM-AE), which used the manifold regularization to constrain ELM-AE to learn the local geometry preserving representations. Moreover, they stacked several GELM-AEs into a deep representation learning model named as multilayer generalized extreme learning machine autoencoder (ML-GELM). Furthermore, H. Ge et al. [127] proposed a graph embedded denoising extreme learning machine autoencoder (GDELM-AE), which integrated the local Fisher discrimination analysis (LFDA) into ELM-AE to discover both local geometry and global discriminative information in the representation space. Similar to ML-ELM and ML-GELM, multiple GDELM-AE can be stacked to build a deep model named as a stacked graph embedded denoising extreme learning machine (SGD-ELM).

In this chapter, a algorithm based on ELM-AE is proposed to learn local discrimination preserving representations. The proposed algorithm is named as the Local Discriminant Preserving Extreme Learning Machine Auto-encoder (LDELM-AE). The proposed algorithm incorporates a graph-based penalty that inspired by marginal fisher analysis (MFA) [63] that exploits both local geometry structure and local discriminant information of input data by maximizing the within-class compactness and between-class separability.

## 4.2 Proposed Method

Let the training dataset  $\{\mathbf{x}_i, \mathbf{y}_i\}_{i=1}^n$  contains  $n$  samples, where  $\mathbf{x}_i \in \mathbb{R}^d$  is the data sample, and  $c_i \in \{1, \dots, m\}$  is the corresponding class label of the  $i$ -th sample.  $\mathbf{y}_i$  is the corresponding one-hot vector, which  $\mathbf{y}_i = [y_{i1}, y_{i2}, \dots, y_{im}]$ ,  $y_{ik} = 1$  only if  $c_i = k$ , otherwise,  $y_{ik} = 0$ .

### 4.2.1 Local discriminant preserving extreme learning machine autoencoder

ELM-AE [13] exploits the intrinsic information of unlabeled data, and GELM-AE [74] improves ELM-AE to discover the latent manifold structure of the input data by integrating the manifold regularization. Furthermore, GDELM-AE [127] integrates LFDA into ELM-AE to discover both local and global structure of the input data. However, LFDA is based on the assumption that the data is Gaussian distributed. Hence, if the input data does not follow the Gaussian distribution assumption, LFDA cannot well characterize the separability of different classes. The proposed LDELM-AE preserves the local geometry and exploits the local discriminative information of the input data. Furthermore, LDELM-AE does not require the Gaussian distribution assumption of the input data.

The structure of LDELM-AE is same as the ELM-AE, which is shown in Figure 2.4. Firstly, the input data is mapped to ELM feature space by an orthogonal random matrix with a non-linear activation function. Secondly, based on the reconstructing cost together with the local discriminative penalty, LDELM-AE utilizes the local geometry and local discriminative information to enhance representations by maximizing the within-class compactness and between-class separability.

Inspired by MFA [63], the within-class compactness measures the sum of Euclidean distances between each data point and the  $k_w$ -nearest neighbor data points of it within the same class. The between-class separability measures the sum of Euclidean distances between the margin data points and their  $k_b$ -nearest neighbor data points in the different classes. The margin data points are the data points located at the border of each class. The illustration of the within-class compactness and the between-class separability are shown in Figure 4.1.

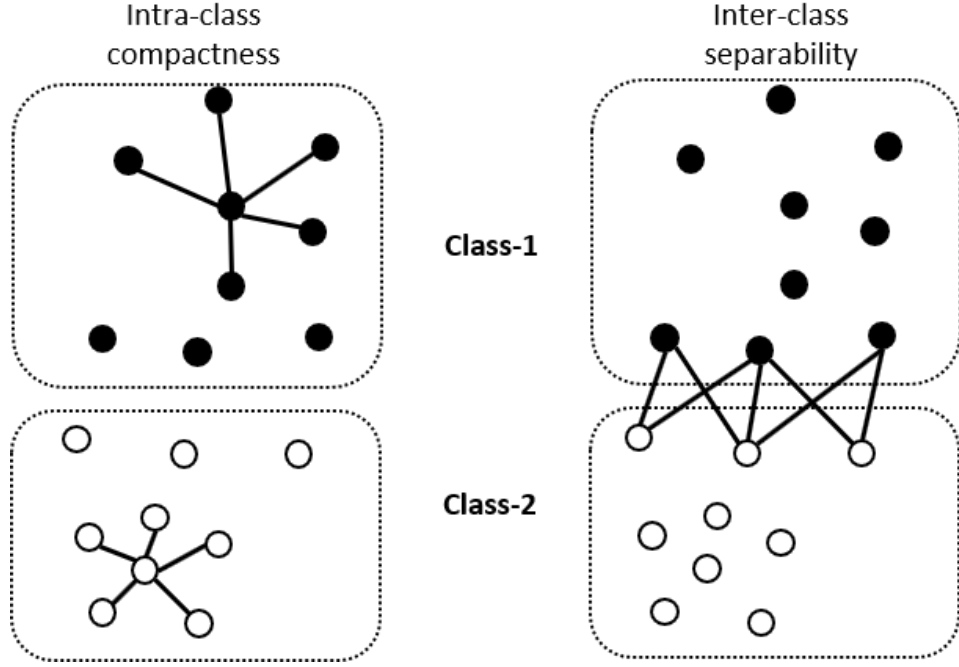


FIGURE 4.1: The black data points belong to class 1 and white data points belong to class 2. The left part shows the within-class compactness that is measured by using the data points and their neighbors from the same class. The right part shows the between-class separability that is measure by using the margin data points and their neighbors from the different class.

In LDELM-AE, the within-class compactness is characterized as the following:

$$\begin{aligned}
 S^w &= \frac{1}{2} \sum_{i,j} s_{ij}^w \|\beta \mathbf{h}_i - \beta \mathbf{h}_j\|^2 \\
 &= \text{Tr}(\beta \mathbf{H} \mathbf{L}^w \mathbf{H}^\top \beta^\top)
 \end{aligned} \tag{4.1}$$

where

$$\begin{aligned}
 L^w &= D^w - S^w \\
 s_{ij}^w &= \begin{cases} \exp\left(\frac{-\|\mathbf{h}_i - \mathbf{h}_j\|^2}{2\sigma^2}\right), & \text{if } \mathbf{x}_j \in \mathcal{N}_i^+ \\ & \text{or } \mathbf{x}_i \in \mathcal{N}_j^+ \\ 0, & \text{else} \end{cases}
 \end{aligned} \tag{4.2}$$

where  $\sigma$  is the parameter scaling the Euclidean distance, and the value of  $\sigma$  normally uses the mean value of Euclidean distances. As similar as the manifold regularization,  $D^w$  is a diagonal matrix with its diagonal elements are  $D_{ii}^w = \sum_j s_{ij}^w$ .  $\mathcal{N}_i^+$  represents the  $k_w$  nearest neighbors of the sample  $\mathbf{x}_i$  within the same class.  $\mathbf{h}_i$  is



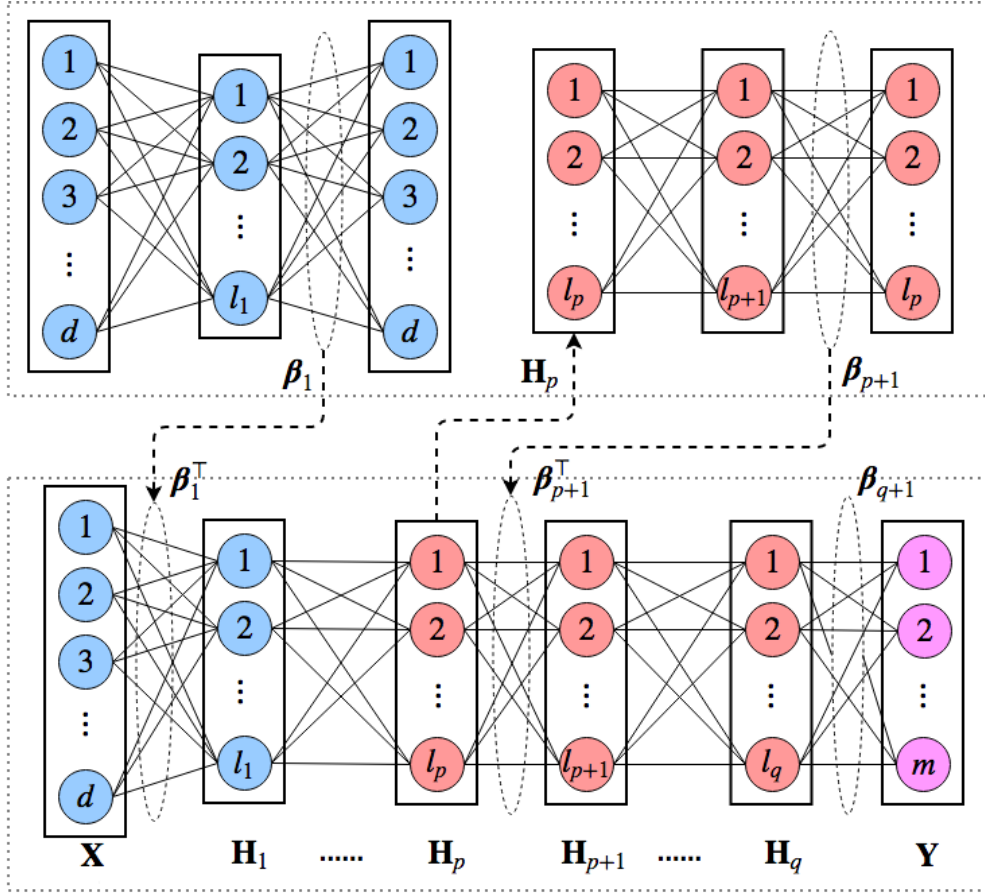


FIGURE 4.2: Architecture of ML-LDELM.

the output of the hidden layer corresponds to the  $i$ -th sample  $\mathbf{x}_i$ , and it can be determined as:

$$\mathbf{h}_i = \frac{1}{1 + \exp(-\mathbf{A}^\top \mathbf{x}_i)} \quad (4.3)$$

where  $\mathbf{I}$  is an identity matrix, and  $\mathbf{A}$  is a weight matrix that generated randomly and orthogonally, i.e.,  $\mathbf{A}^\top \mathbf{A} = \mathbf{I}$  if  $d < l$  and  $\mathbf{A}\mathbf{A}^\top = \mathbf{I}$  if  $d \geq l$ .

**Remark 4.2.1.** Assume two neighbor data points from the same class form a data pair  $\{(\mathbf{x}_i, \mathbf{x}_j), \forall \mathbf{x}_i \in \mathcal{N}_j^+, \mathbf{x}_j \in \mathcal{N}_i^+\}$ .  $\mathbf{S}^w$  weights all data pairs in the dataset, smaller the distance between two data points, higher the weight value of the data pair. Hence, high penalty values are assigned to those neighbor data points that are too different in the representation space. By minimizing (4.1), the closed data points from the same class can be mapped to similar representations and the compactness of each class can be increased. Therefore, the representations preserve the local geometry from the input data.

However, preserving local geometry does not consider the separation between different classes. Hence, to separate data points from different classes in the representation space, the between-class separability is characterized as the following:

$$\begin{aligned} \mathbf{S}^b &= \frac{1}{2} \sum_{i,j} s_{ij}^b \|\beta \mathbf{h}_i - \beta \mathbf{h}_j\|^2 \\ &= \text{Tr}(\beta \mathbf{H} \mathbf{L}^b \mathbf{H}^\top \beta^\top) \end{aligned} \quad (4.4)$$

where

$$\begin{aligned} \mathbf{L}^b &= \mathbf{D}^b - \mathbf{S}^b \\ s_{ij}^b &= \begin{cases} \exp(-\frac{\|\mathbf{h}_i - \mathbf{h}_j\|^2}{2\sigma^2}), & \text{if } (\mathbf{x}_i, \mathbf{x}_j) \in \mathcal{P}(c_i) \\ & \text{or } (\mathbf{x}_j, \mathbf{x}_i) \in \mathcal{P}(c_j) \\ 0, & \text{else} \end{cases} \end{aligned} \quad (4.5)$$

where  $\mathcal{P}(c_i)$  is a set of sample pairs contains the  $k_b$  shortest pairs in  $\{(\mathbf{x}_i, \mathbf{x}_j), \forall \mathbf{x}_i \in c_i, \mathbf{x}_j \notin c_i\}$ .

**Remark 4.2.2.** Assume two data points from the different class form a data pair  $\{(\mathbf{x}_i, \mathbf{x}_j), \forall \mathbf{x}_i \in c_i, \mathbf{x}_j \notin c_i\}$ .  $\mathbf{S}^b$  weights  $k_b$  shortest data pairs in the dataset, smaller the distance between two data points, higher the weight value of the data pair. By choosing  $k_b$  shortest data pairs, we only consider the distances between margin data points of different class. By maximizing (4.4), the separability between margin points from different class is increased in the representation space. Since the between-class separability is increased by margin points locally, the representations exploit local discriminative information from the input data.

Combining the within-class compactness and the between-class separability, the objective function of LDELM-AE can be formulated as the following:

$$\begin{aligned} \arg \min_{\beta} & \frac{1}{2} \|\beta\|_F^2 + \frac{C}{2} \|\beta \mathbf{H} - \mathbf{X}\|_F^2 + \frac{\lambda}{2} (\mathbf{S}^w - \gamma \mathbf{S}^b) \\ \text{s.t.} \quad & \mathbf{S}^w = \text{Tr}(\beta \mathbf{H} \mathbf{L}^w \mathbf{H}^\top \beta^\top) \\ & \mathbf{S}^b = \text{Tr}(\beta \mathbf{H} \mathbf{L}^b \mathbf{H}^\top \beta^\top) \end{aligned} \quad (4.6)$$

By substituting the constraints into the objective function, it can be rewritten as the following:

$$\begin{aligned} \arg \min_{\boldsymbol{\beta}} \frac{1}{2} \|\boldsymbol{\beta}\|_F^2 + \frac{C}{2} \|\boldsymbol{\beta}\mathbf{H} - \mathbf{X}\|_F^2 + \\ \frac{\lambda}{2} (\text{Tr}(\boldsymbol{\beta}\mathbf{H}(\mathbf{L}^w - \gamma\mathbf{L}^b)\mathbf{H}^\top \boldsymbol{\beta}^\top)) \end{aligned} \quad (4.7)$$

where  $C$ ,  $\lambda$  and  $\gamma$  are the trade-off hyper-parameters determined by users.

Since the objective function (4.7) is convex, it can be minimized by solving the equation  $\nabla_{\text{LDELM-AE}} = 0$ , where

$$\begin{aligned} \nabla_{\text{LDELM-AE}} &= \boldsymbol{\beta} + C(\boldsymbol{\beta}\mathbf{H} - \mathbf{X})\mathbf{H}^\top + \lambda\boldsymbol{\beta}\mathbf{H}(\mathbf{L}^w - \gamma\mathbf{L}^b)\mathbf{H}^\top \\ &= \boldsymbol{\beta}(\mathbf{I} + C\mathbf{H}\mathbf{H}^\top + \lambda\mathbf{H}(\mathbf{L}^w - \gamma\mathbf{L}^b)\mathbf{H}^\top) - \mathbf{X}\mathbf{H}^\top \end{aligned} \quad (4.8)$$

Hence,  $\boldsymbol{\beta}$  can be determined in closed form with two different situations. While the number of training samples is greater than the number of hidden neurons, i.e.,  $n > l$ ,  $\boldsymbol{\beta}$  is calculated as follows:

$$\boldsymbol{\beta}^* = \mathbf{X}\mathbf{H}^\top (\mathbf{I}_l + C\mathbf{H}\mathbf{H}^\top + \lambda\mathbf{H}(\mathbf{L}^w - \gamma\mathbf{L}^b)\mathbf{H}^\top)^{-1} \quad (4.9)$$

where  $\mathbf{I}_l$  is the identity matrix of dimension  $l$ .

While the number of training samples is smaller than the number of hidden neurons, i.e.,  $n < l$ ,  $\boldsymbol{\beta}$ ,  $\mathbf{H}$  contains more columns than rows. Hence, it is beneficial to let  $\boldsymbol{\beta} = \boldsymbol{\alpha}\mathbf{H}^\top$ , where  $\boldsymbol{\alpha} \in \mathbb{R}^{n \times d}$ . Therefore, the closed form solution of  $\boldsymbol{\beta}$  is calculated as follows:

$$\boldsymbol{\beta}^* = \mathbf{X}(\mathbf{I}_n + C\mathbf{H}^\top \mathbf{H} + \lambda\mathbf{H}^\top \mathbf{H}(\mathbf{L}^w - \gamma\mathbf{L}^b))^{-1} \mathbf{H}^\top \quad (4.10)$$

where  $\mathbf{I}_n$  is the identity matrix of dimension  $n$ .

For the given training data  $\mathbf{X}$ , the embedded data representations  $\mathbf{X}_{proj} \in \mathbb{R}^{l \times n}$  can be determined by  $\mathbf{X}_{proj} = \boldsymbol{\beta}^\top \mathbf{X}$ . The learned representations can be used for further classification tasks. The details of LDELM-AE is shown in Algorithm 2.

---

**Algorithm 2** LDELM-AE algorithm.

---

**Input:**

- The input data  $\{\mathbf{X}, \mathbf{Y}\} = \{\mathbf{x}_i, \mathbf{y}_i\}_{i=1}^n$ .
- The number of hidden neurons  $l$ .
- The hyper-parameters  $C$ ,  $\lambda$  and  $\gamma$ .

**Output:**

- The output weights  $\beta$ .
  - The representations  $\mathbf{X}_{proj}$  of the input data  $\mathbf{X}$ .
- 1: Randomly initiate the input weights  $\mathbf{A}$ .
  - 2: Apply the singular value decomposition (SVD) on  $\mathbf{A}$  to generate the orthogonalized  $\mathbf{A}$ .
  - 3: Compute the outputs of the hidden layer  $\mathbf{H}$  using (4.3).
  - 4: Compute the graph Laplacian matrix  $L^w$  and  $L^b$  using (4.2) and (4.5), respectively.
  - 5: **if**  $n > l$  **then**
  - 6:     Compute the output weights  $\beta$  using (4.9).
  - 7: **else**
  - 8:     Compute the output weights  $\beta$  using (4.10).
  - 9: **end if**
  - 10: Compute the representations of input data by  $\mathbf{X}_{proj} = \beta^\top \mathbf{X}$ .
  - 11: **return**  $\mathbf{X}_{proj}$ ,  $\beta$
-

### 4.2.2 Multilayer local discriminant preserving extreme learning machine autoencoder

ML-LDELM is proposed by stacking multiple LDELM-AEs to learn hierarchical representations from the input data. As shown in Figure 4.2, ML-LDELM is trained in the greedy layer-wise approach.

In Figure 4.2, the first LDELM-AE uses  $\mathbf{X}$  as the input data to learn the output weight matrix  $\beta_1$ . Outputs of the first hidden layer can be found by the transpose of the output weight matrix:  $\mathbf{H}_1 = \beta_1^\top \mathbf{X}$ . In another words,  $\beta_1^\top$  is used as the input weight matrix of the first hidden layer in ML-LDELM. The input weight matrix  $\beta_2^\top$  of the second hidden layer is learned by using the same approach. The only difference is that instead of using  $\mathbf{X}$ ,  $\mathbf{H}_1$  is used as the input data of the second LDELM-AE. In general, the  $p$ -th LDELM-AE uses  $\mathbf{H}_{p-1}$  as the input data to learn  $\beta_p$ , and  $\beta_p^\top$  is used as the input weight matrix of the hidden layer  $\mathbf{H}_p$  in ML-LDELM.

Assume there are total  $q$  hidden layers in ML-LDELM, the output weight matrix  $\beta_{q+1}$  between the last hidden layer, i.e., the  $q$ -th hidden layer, and the output layer in ML-LDELM are computed by optimizing the regularized least squares, which can be formulated as the following:

$$\min_{\beta_{q+1}} \frac{1}{2} \|\beta_{q+1}\|_F^2 + \frac{C}{2} \|\mathbf{Y} - \beta_{q+1} \mathbf{H}_q\|_F^2 \quad (4.11)$$

where  $\mathbf{Y} = [\mathbf{y}_1, \mathbf{y}_2, \dots, \mathbf{y}_n]^\top \in \mathbb{R}^{m \times n}$  is the label matrix consists of one-hot vectors, and  $\mathbf{H}_q$  is the output of the last hidden layer in ML-LDELM. The closed-form solution of  $\beta_{q+1}$  can be determined as the following:

$$\beta_{q+1}^* = \begin{cases} \mathbf{X} \mathbf{H}_q^\top \left( \mathbf{H}_q \mathbf{H}_q^\top + \frac{\mathbf{I}_{l_q}}{C} \right)^{-1} & \text{if } N > l \\ \mathbf{X} \left( \mathbf{H}_q^\top \mathbf{H}_q + \frac{\mathbf{I}_n}{C} \right)^{-1} \mathbf{H}_q^\top & \text{otherwise} \end{cases} \quad (4.12)$$

The details of using ML-LDELM for classification tasks are summarized in Algorithm 3.

---

**Algorithm 3** ML-LDELM algorithm for classification tasks.

---

**Input:**

- The training data  $\{\mathbf{X}_{train}, \mathbf{Y}_{train}\} = \{\mathbf{x}_i, \mathbf{y}_i\}_{i=1}^n$ .
- The test data  $\{\mathbf{X}_{test}, \mathbf{Y}_{test}\} = \{\mathbf{x}_i, \mathbf{y}_i\}_{i=1}^m$ .
- The number of hidden layers  $q$ .
- The number of neurons of each hidden layer  $\{l_p\}_{p=1}^q$ .
- The hyper-parameters  $C$ ,  $\lambda$  and  $\gamma$ .

**Output:**

- The input weights of each hidden layer  $\{\boldsymbol{\beta}_p^\top\}_{p=1}^q$ .
- The predicted labels of the input data.

**Train:**

- 1: Initialize  $\mathbf{H}_0 = \mathbf{X}_{train}$ .
- 2: **for**  $p = 1 : q$  **do**
- 3:   Use  $\mathbf{H}_{p-1}$  as the inputs of the  $p$ -th LDELM-AE to train the output weights  $\boldsymbol{\beta}_p$  by Algorithm 2.
- 4:   Use  $\boldsymbol{\beta}_p^\top$  as the input weights and  $\mathbf{H}_{p-1}$  as the inputs of the  $p$ -th hidden layer in ML-LDELM to compute  $\mathbf{H}_p$  by (4.3).
- 5: **end for**
- 6: Compute  $\boldsymbol{\beta}_{q+1}$  using (4.12).

**Test:**

- 7: Initialize  $\mathbf{H}_0 = \mathbf{X}_{test}$ .
  - 8: **for**  $p = 1 : q$  **do**
  - 9:   Use trained  $\boldsymbol{\beta}_p^\top$  as the input weights and  $\mathbf{H}_{p-1}$  as the inputs of the  $p$ -th hidden layer in ML-LDELM to compute  $\mathbf{H}_p$  by (4.3).
  - 10: **end for**
  - 11: Compute the predicted label  $\tilde{\mathbf{Y}} = \mathbf{H}_q \boldsymbol{\beta}_{q+1}$ .
  - 12: **return**  $\tilde{\mathbf{Y}}, \{\boldsymbol{\beta}_p^\top\}_{p=1}^q$
-

### 4.3 Experiments

The proposed representation learning algorithms, LDELM-AE and ML-LDELM, are compared with the related algorithms on several benchmark datasets. We also apply the proposed LDELM-AE on a bearing fault dataset to validate its effectiveness on the machine fault diagnosis tasks. The performances of the proposed algorithms are compared with state-of-the-art algorithms.

TABLE 4.1: Descriptions of benchmark datasets.

Type	Datasets	# of Classes	# of Dimensions	# of Samples
UCI	IRIS	3	4	150
	WINE	3	13	178
	LIVER	2	6	345
	SATIMAGE	6	36	6435
Image	COIL20	20	1024	1440
	USPST	10	256	2007

#### 4.3.1 Datasets Descriptions and Experiment Setups

The summary of the benchmark datasets is shown in Table 4.1. The first four datasets, i.e., IRIS, WINE, LIVER, SATIMAGE, are low-dimensional datasets from UCI machine learning repository [128]. The COIL20 [129] contains 1440 gray-scale images of 20 objects. There are 72 images were taken in different poses of each object. The USPST is a subset of USPS [130], which contains ten categories of gray-scale handwritten digit images.

Furthermore, the CWRU bearing dataset aims to classify different machine health conditions based on the vibration signal. The dataset was collected under four health conditions, which were the normal condition, roller fault, outer raceway fault, and inner raceway fault. Each health condition includes three severity levels. Hence, the task is to classify the normal condition and nine different faults of the bearing. The details of the the CWRU bearing dataset are described in Section 3.3.1. In this experiment, since the original vibration signal is non-stationary, we use the wavelet packet transform, with 5-level wavelet packet decomposition

using *db1*, to transfer the time domain vibration signal to the time-frequency domain.

To evaluate the effectiveness of the proposed algorithms, we compared the proposed LDELM-AE and ML-LDELM with several related algorithms on the first six datasets of Table 4.1. The related algorithms are shown in the following:

- 1) ELM-AE and ML-ELM [13].
- 2) GELM-AE and ML-GELM [74].
- 3) GDELM-AE and SGD-ELM [127].
- 4) Autoencoder (AE) and stacked AE (SAE) [54].
- 5) Deep belief network (DBN) [9].

To evaluate the performance of the proposed LDELM-AE on the machine fault diagnosis task, we compared it with the state-of-the-art algorithms that have been applied to the same application. W. Du et al. [117] extracted multifractal features based on the discrete wavelet transform and used the support vector machine (SVM) to classify the ten machine conditions in CWRU dataset. X. Zhang et al. [118] also used SVM as the classifier, but they extracted features by using the ensemble empirical mode decomposition (EEMD). C. Lu et al. [21] used both stacked denoising AE (SDA) and SAE to classify four machine conditions in CWRU dataset, i.e., four types of machine conditions without the consideration of severity levels. LDELM-AE is also compared with SFAE-LG introduced in Chapter 3 on CWRU dataset.

In the experiments, all of the hyper-parameters were selected using cross-validation. The number of hidden neurons is selected from 100 to 5000 with an interval of 100 for all algorithms. The trade-off hyper-parameters, e.g.,  $C$ ,  $\lambda$  and  $\gamma$ , were selected from the exponential sequence  $[1e-5, \dots, 1e10]$  according to the validation performances. The ELM-based algorithms used the sigmoid nonlinear activation function. We run each algorithm for 50 times independently to reduce the influences of the randomness.

The classification accuracy (ACC) is used as the evaluation metric in this study of the proposed algorithm and the other related algorithms. ACC measures the



percentage of the correctly classified data points among all of the data points, and it is defined as:

$$ACC = \frac{\sum_i^N \delta(\mathbf{y}_i - \hat{\mathbf{y}}_i)}{N} \quad (4.13)$$

where  $\mathbf{y}_i$  and  $\hat{\mathbf{y}}_i$  are the ground truth and predicted label of  $\mathbf{x}_i$ . The function  $\delta(\mathbf{a}, \mathbf{b})$  is defined as:

$$\delta(\mathbf{a}, \mathbf{b}) = \begin{cases} 1, & \text{if } \mathbf{a} = \mathbf{b} \\ 0, & \text{otherwise} \end{cases} \quad (4.14)$$

### 4.3.2 Comparison with the related methods

The comparison results of all algorithms with one, two and three hidden layers are shown in Table 4.2, Table 4.3 and Table 4.4, respectively. We run each algorithm for 50 times independently, and reported the average (avg.) value, the standard deviation, and the best value among the 50 accuracies in the tables.

In Table 4.2, we observed that the proposed LDELM-AE outperformed all of the other algorithms on the tested datasets. Besides, observing from Table 4.3 and Table 4.4, when the number of hidden layers increased, the performances of each algorithm are also increased. This observation verifies that the hierarchical representations can improve the classification performances.

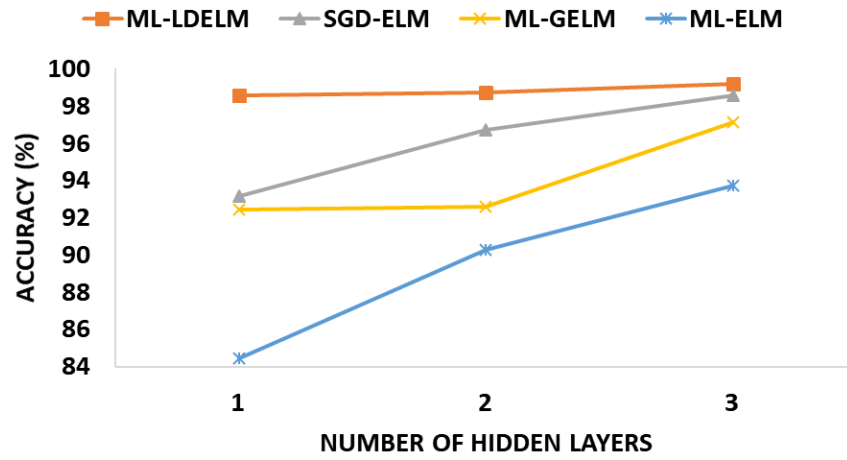
Moreover, we observed that the proposed ML-LDELM outperformed all of the other algorithms with different network structures, i.e., the different number of hidden layers, and it proved the effectiveness of preserving the local geometry and exploiting the local discriminant information of the input data in the representations. To observe the performances on each dataset more intuitively, we plot the average accuracies of all ELM-based algorithms with the different number of hidden layers in Figure 4.3, Figure 4.4, and Figure 4.5.

TABLE 4.2: The comparison of classification accuracies for each algorithm with one hidden layer.

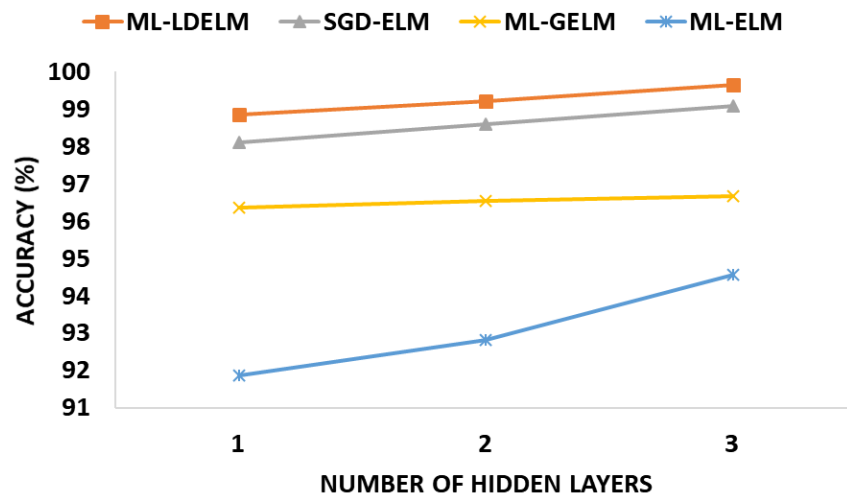
Methods	Acc	IRIS	WINE	LIVER	SATIMAGE	COIL20	USPST
DBN	Avg.	95.11+3.58	92.14+0.46	50.98+8.16	83.42+0.96	88.33+2.45	86.14+0.81
	Best	98.67	93.13	58.14	84.81	92.08	87.91
AE	Avg.	94.16+1.23	88.37+0.77	55.02+2.68	83.16+1.58	58.86+3.87	85.89+1.67
	Best	97.33	89.08	58.14	85.09	66.01	88.9
ELM-AE	Avg.	84.44+0.63	91.87+0.9	54.55+0.12	84.57+0.15	59.39+4.66	83.95+0.47
	Best	85.33	94.38	54.78	84.98	68.96	85.33
GELM-AE	Avg.	92.44+3.94	96.37+0.78	56.83+3.52	80.93+2.64	61.78+4.99	87.22+0.61
	Best	94	97.19	60.9	84.48	71.25	88.92
GDELM-AE	Avg.	93.16+0.52	98.11+0.28	60.91+1.86	84.29+1.04	68.47+3.69	88.76+0.42
	Best	93.33	98.32	63.19	86.01	76.18	90.12
LDELM-AE	Avg.	<b>98.59+0.72</b>	<b>98.85+0.64</b>	<b>70.78+1.21</b>	<b>90.98+0.19</b>	<b>99.36+0.21</b>	<b>92.05+0.13</b>
	Best	<b>100</b>	<b>100</b>	<b>74.42</b>	<b>91.67</b>	<b>100</b>	<b>92.52</b>

TABLE 4.3: The comparison of classification accuracies for each algorithm with two hidden layers.

Methods	$Acc$	IRIS	WINE	LIVER	SATIMAGE	COIL20	USPST
DBN	Avg.	96.35+1.82	94.62+0.92	53.91+7.21	85.91+3.14	94.77+1.62	87.23+1.24
	Best	98.67	95.79	58.14	87.34	97.22	89.88
SAE	Avg.	94.57+0.46	90.08+0.84	58.03+3.79	85.87+3.48	94.12+0.98	91.11+3.04
	Best	97.33	90.89	62.21	87.88	96.98	93.81
ML-ELM	Avg.	90.26+0.78	92.81+0.03	57.95+0.04	85.33+0.21	96.06+0.14	88.34+0.21
	Best	93.33	95.45	62.23	85.97	96.39	88.92
ML-GELM	Avg.	92.58+0.91	96.55+0.79	62.48+3.49	83.81+2.48	97.81+0.12	89.84+0.27
	Best	97.33	97.73	73.84	86.69	99.58	90.52
SGD-ELM	Avg.	96.71+0.69	98.61+1.03	66.74+2.36	86.07+3.29	98.87+0.26	89.85+0.23
	Best	94.67	99.54	70.35	87.84	97.92	90.62
ML-LDELM	Avg.	<b>98.75+0.57</b>	<b>99.21+0.62</b>	<b>71.96+1.22</b>	<b>91.63+0.32</b>	<b>99.62+0.23</b>	<b>93.76+0.46</b>
	Best	<b>100</b>	<b>100</b>	<b>76.16</b>	<b>92.83</b>	<b>100</b>	<b>94.97</b>

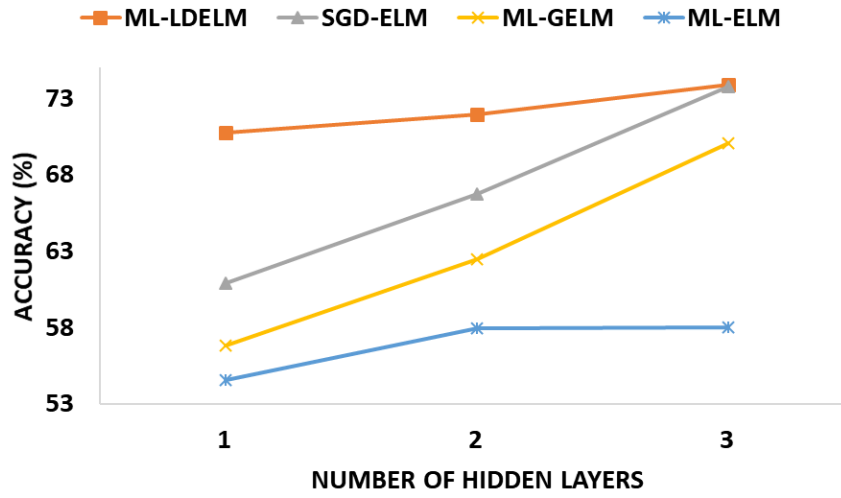


(A) IRIS

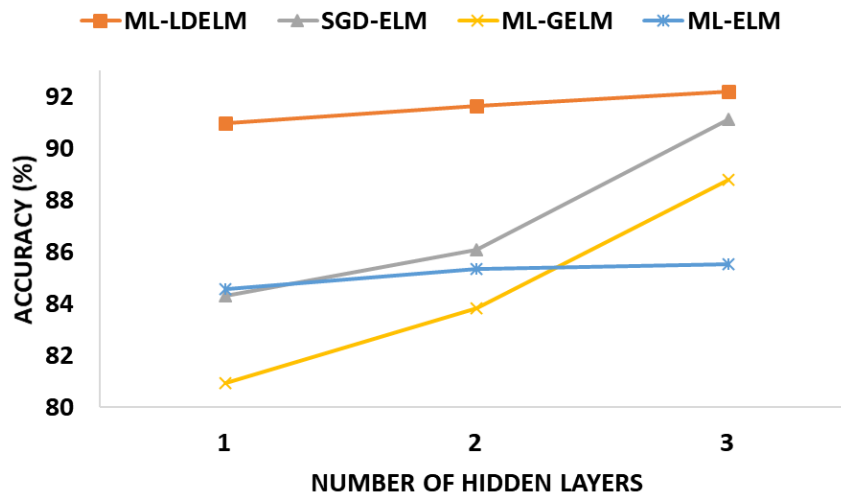


(B) WINE

FIGURE 4.3: Average accuracy of different number of hidden layers for ELM-based algorithms on: (a) IRIS dataset and (b) WINE dataset.

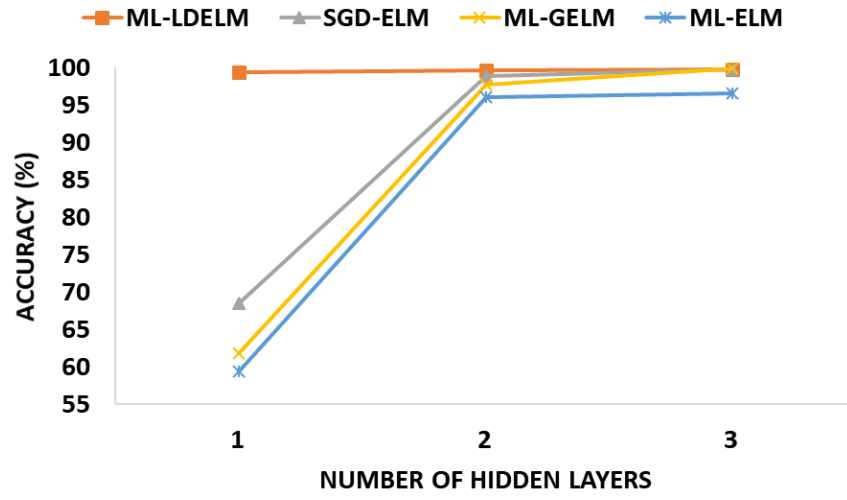


(A) LIVER

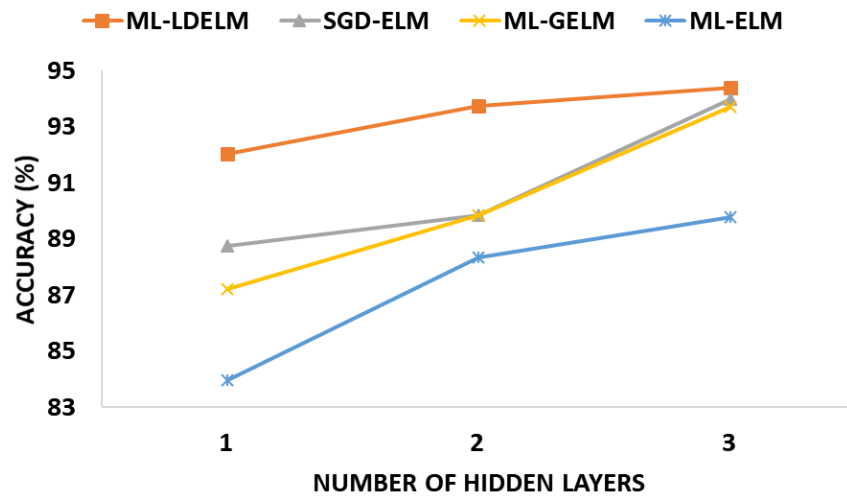


(B) SATIMAGE

FIGURE 4.4: Average accuracy of different number of hidden layers for ELM-based algorithms on: (a) LIVER dataset and (b) SATIMAGE dataset.



(A) COIL20



(B) USPST

FIGURE 4.5: Average accuracy of different number of hidden layers for ELM-based algorithms on: (a) COIL20 dataset and (b) USPST dataset.

TABLE 4.4: The comparison of classification accuracies for each algorithm with three hidden layers.

Methods	Acc	IRIS	WINE	LIVER	SATIMAGE	COIL20	USPST
DBN	Avg.	96.67 + 1.44	96+1.83	59.65+0.53	86.2+0.66	98.85+0.09	88.96+0.81
	Best	98.67	98.89	60.69	87.36	99.03	90.25
SAE	Avg.	95.07+2.95	90.78+0.54	61.97 + 3.96	87.67+1.33	98.81+0.2	93.69+0.33
	Best	<b>100</b>	91.11	65.9	88.97	99.17	94.13
ML-ELM	Avg.	93.73+2.78	94.56+1.85	58.03+3.61	85.52+0.53	96.58+0.7	89.78+1.58
	Best	97.33	96.67	64.74	86.64	97.5	92.04
ML-GELM	Avg.	97.15+0.54	96.67+0.48	70.08+2.96	88.77+0.44	<b>99.9+0.99</b>	93.72+0.48
	Best	97.33	98.89	75.72	89.53	<b>100</b>	94.53
SGD-ELM	Avg.	98.58+0.78	99.09+0.96	73.8+1.26	91.13+0.2	99.85+0.15	93.98+0.55
	Best	<b>100</b>	<b>100</b>	76.16	91.43	<b>100</b>	95.02
ML-LDELIM	Avg.	<b>99.2+0.68</b>	<b>99.64+0.54</b>	<b>73.91+2.16</b>	<b>92.19+0.23</b>	99.86+0.06	<b>94.39+0.54</b>
	Best	<b>100</b>	<b>100</b>	<b>78.26</b>	<b>93.22</b>	<b>100</b>	<b>95.98</b>

### 4.3.3 Application on the machine fault diagnosis

To verify the capability of the proposed algorithm on solving machine fault diagnosis tasks, we tested LDELM-AE on CWRU bearing dataset.

Firstly, we investigate the selection of hyper-parameters in LDELM-AE, i.e., the number of hidden numbers  $l$ , trade-off parameters  $C$ ,  $\lambda$  and  $\gamma$ . It should be noted that we run 50 trails for each experiment.

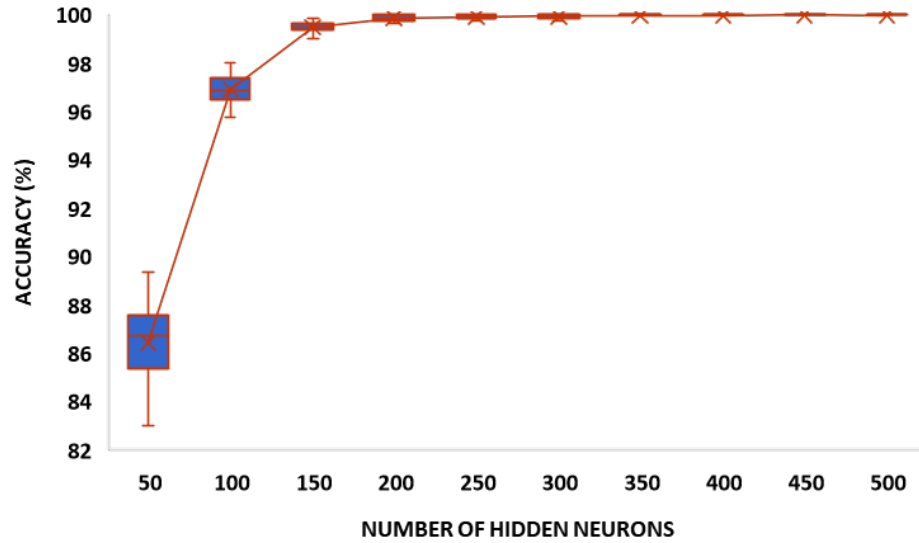


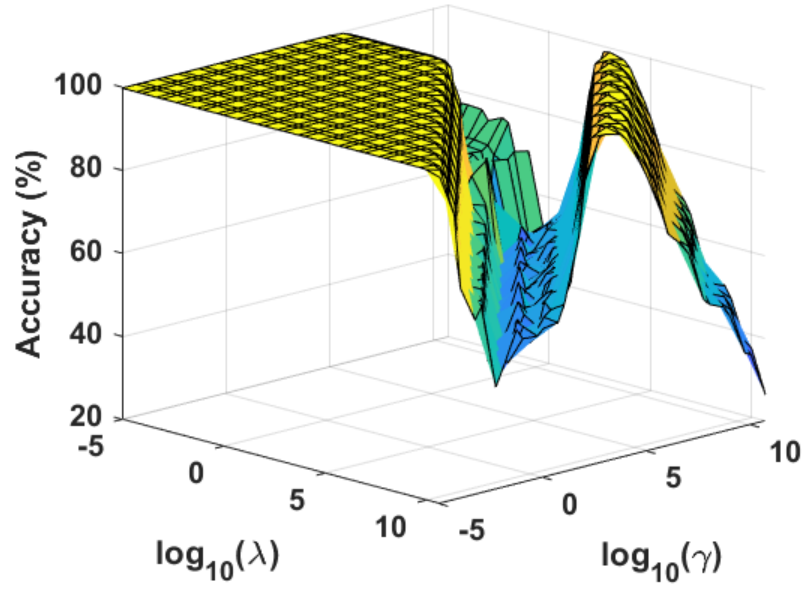
FIGURE 4.6: Machine fault diagnosis performances using different number of hidden neurons.



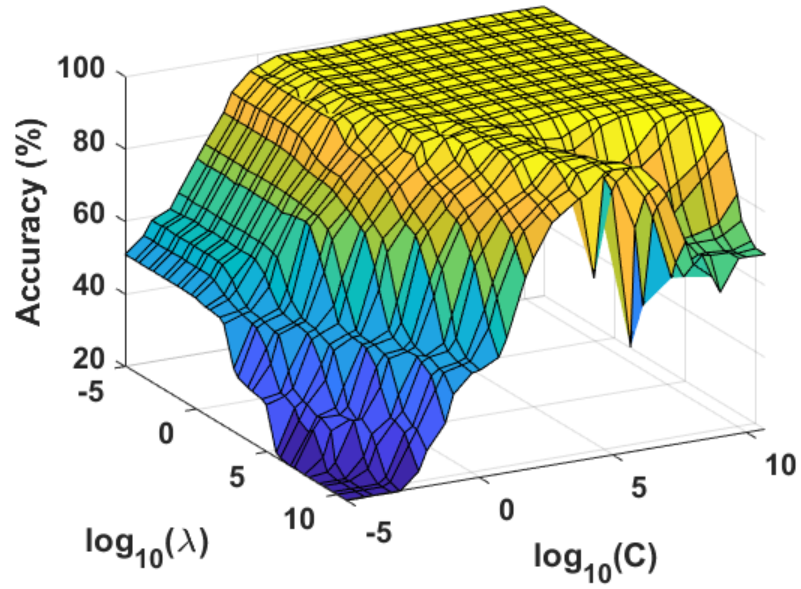
**Selection of the number of hidden numbers** In this experiment, we study how does the number of hidden numbers  $l$  affects the accuracy of the machine fault diagnosis. From Figure 4.6, we noticed that the proposed algorithm obtained higher accuracy and smaller standard deviation while the number of hidden neurons increased. Moreover, while the number of hidden neurons larger than 200, the average accuracies are stable and between 99.88% to 100%. While the number of hidden neurons larger than 350, the standard deviation is stably smaller than 0.05. Therefore, we used 350 hidden neurons in the following experiments.

**Selection of the trade-off parameters** In this experiment, we study how does the trade-off parameters affect the accuracy of the machine fault diagnosis. Figure 4.7a shows the machine fault diagnostic accuracy of LDELM-AE with different combinations of trade-off parameters  $\lambda$  and  $\gamma$ . High diagnosis accuracies are observed while  $\log_{10}(\lambda\gamma) \leq 33$ . This observation shows the proposed algorithm can produce a stable performance, i.e., diagnosis accuracies fluctuate slightly, while trade-off parameters  $\lambda$  and  $\gamma$  are within a certain range. Moreover, Figure 4.7b shows the diagnostic accuracy with different combinations of parameter  $\lambda$  and  $C$ . It can be observed the diagnostic accuracy is stable while  $\log_{10}(C) \geq 0$  and  $\log_{10}(\lambda) \leq 5$ . Therefore, for CWRU bearing dataset, the proposed LDELM-AE is not sensitive to the values of trade-off parameters while  $\{C, \lambda, \gamma\} \in \log_{10}(C) \geq 0 \cap \log_{10}(\lambda) \leq 5 \cap \log_{10}(\lambda\gamma) \leq 33$ .

**Visualization of the representations** To investigate the effects of the local discriminative penalty of LDELM-AE, we visualized the original data points, and the representations learned by LDELM-AE in Figure 4.8. In Figure 4.8a, the data points of different health conditions are clustered together with large within-class variance and small between-class discrimination, e.g., data points from IRF1, IRF3, ORF1, and ORF3 are mixed in the original data space. In Figure 4.8b, we observed that the majority of representations of the same health condition are clustered together and representations of the different health conditions are separated. Hence, compared to the original data space, the learned representations increased the between-class discrimination and the within-class compactness. Therefore, the proposed algorithm could learn representations that improve diagnostic accuracy.

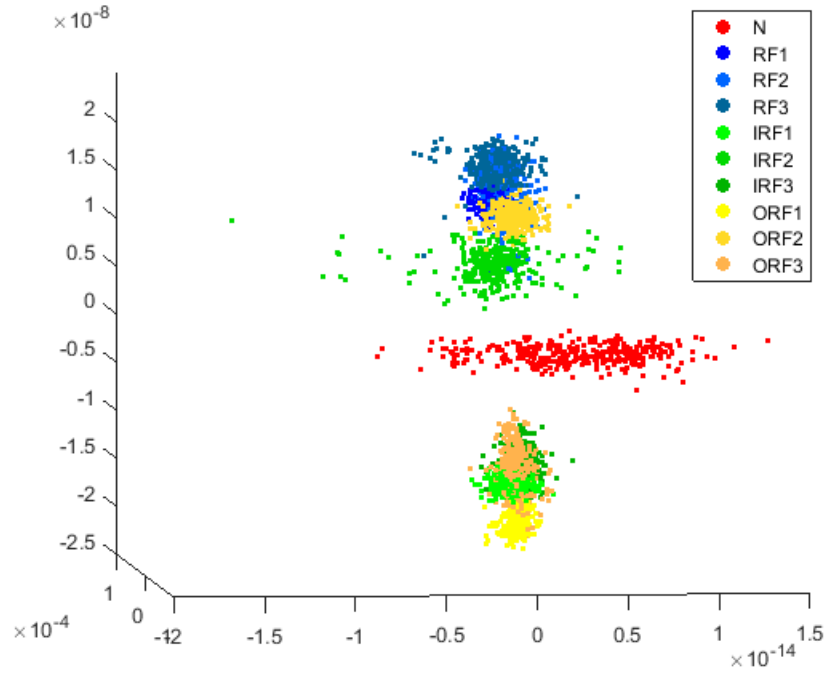


(A)

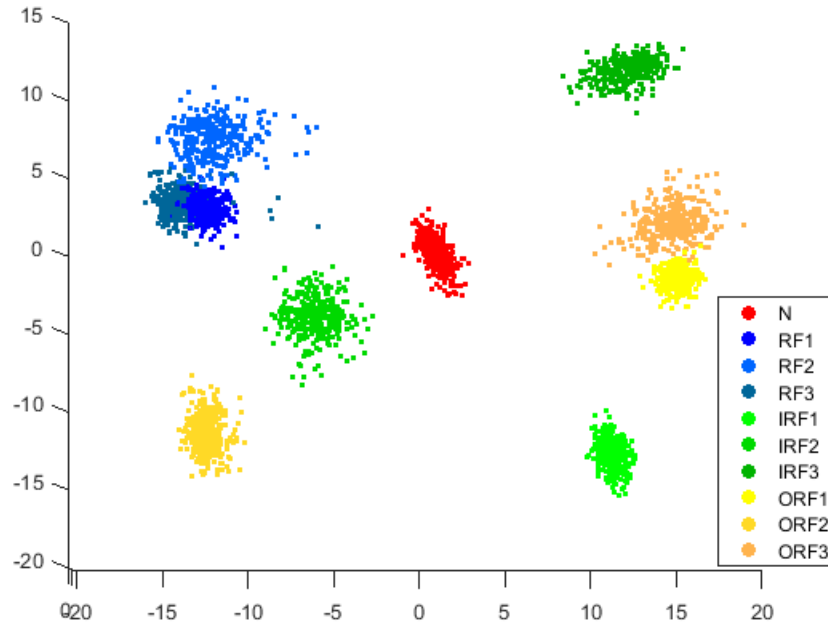


(B)

FIGURE 4.7: Machine fault diagnosis performances using different combinations of trade-off parameters. (a) shows the diagnosis accuracies with different combinations of  $\lambda$  and  $\gamma$ . (b) shows the diagnosis accuracies with different combinations of  $\lambda$  and  $C$ .



(A) Original data



(B) Representations learned by LDELM-AE

FIGURE 4.8: Scatter plots of the original data points and learned data representations of the CWRU bearing dataset.

**Comparison with state-of-the-art algorithms** In Table 4.5, we report the performance of the proposed LDELM-AE tested on CWRU bearing dataset and compared it with the published diagnostic accuracies of state-of-the-art algorithms on the same benchmark dataset. In [117], the wavelet leaders multifractal features and SVM are used to classify the ten classes of the motor bearing and achieved the accuracy of 88.9%. EEMD [118] was applied to decompose the vibration signal into the intrinsic mode functions (IMFs) and calculate permutation entropies, which is used as the features. The diagnostic accuracy of 97.91% was obtained by using SVM as the classifier. SAE and SDA [21] were applied to the time-domain vibration signal to learn representations, and the softmax layer on the top of the last hidden layer achieved 94.4% and 95.58% diagnostic accuracy, respectively, for the four classes of bearing health conditions. The previous algorithm SFAE-LG in Chapter 3 learns representations with local and global geometry preserved from the time-domain vibration signal and achieved the accuracy of 97.29% of the ten machine conditions in CWRU dataset. Compared with the above algorithms, the LDELM-AE obtains the highest diagnostic accuracy of  $99.74 \pm 0.17\%$ . The observation proves the effectiveness of using the proposed algorithm to solve the machine fault diagnosis tasks.

TABLE 4.5: The comparison of classification accuracies on CWRU bearing dataset.

Methods	# of training samples (%)	# of classes	Acc (%)
Multifractal features + SVM [117]	75	10	88.9
EMD + SVM [118]	40	11	97.91
SAE [21]	40	4	94.4
SDA [21]	40	4	95.58
SFAE-LG	20	10	97.29
LDELM-AE	20	10	99.74

**Computational complexity** To investigate the computational complexity of using ELM-based algorithm, Table 4.6 shows the comparison on both training and test time between FAE-LG in Chapter 3 and LDELM-AE in Chapter 4. For a fair comparison, both algorithms used the same network structure, i.e., one hidden layer with 1000 hidden neurons.

TABLE 4.6: Comparison of training and testing time on CWRU dataset.

Methods	Training time (s)	Test time (s)
SFAE-LG	30.88	0.017
LDELM-AE	<b>1.26</b>	<b>0.016</b>

From Table 4.6, it can be noticed that LDELM-AE introduced in this chapter improves the training time significantly. It is because the cost function of the ELM-based representation learning algorithm, LDELM-AE, can be minimized analytically. Compared to FAE-LG introduced in Chapter 3, LDELM-AE does not require BP technique, which is time-consuming, for the training process. Moreover, the test time of both algorithms are comparable, which because they use the same network structure with the same computational complexity in the test.

## 4.4 Summary

This chapter aims to address both of the research objectives of the thesis, i.e., to efficiently learn data representations that can improve the performance of machine fault diagnosis tasks, and to exploit and preserve geometry information of input data while learning data representations. Section 4.1 reviews the ELM-based representation learning algorithm and its variants that exploit geometry information of input data. Section 4.2 introduces a representation learning algorithm LDELMAE, and its multi-layer framework ML-LDELMAE to efficiently learn data representations with two properties: 1) preserving the local geometry; 2) exploiting the local discrimination information from the input data. The local geometry of input data is preserved by minimizing the Euclidean distances between each data point and its nearest neighbors within the same class. The local discrimination information is exploited by maximizing the Euclidean distances between the margin data points and their neighbors in the different classes. Multiple LDELMAEs are stacked to form ML-LDELMAE, which is used to learn hierarchical representations from the input data. In Section 4.3, the experimental results demonstrate that the proposed algorithms outperformed the other related algorithms, e.g., SGD-ELM, ML-GELM, ML-ELM, on several benchmark datasets. The observations proved the effectiveness of the two properties exploited by the proposed algorithms. Furthermore, the proposed algorithms were applied to solve machine fault diagnosis tasks. To provide guidelines for using the proposed algorithm, we analyzed the hyper-parameters while applying it on the bearing fault dataset. Compared to the state-of-the-art algorithms, the proposed algorithm achieved higher diagnostic accuracy with less training time so that it was proved a useful tool to diagnose machine faults accurately.



## Chapter 5

# Simultaneously Learning Affinity Matrix and Data Representations

*Chapter 5 proposes a representation learning algorithm that learns the data representations and affinity matrix simultaneously. Instead of predefining and fixing the affinity matrix, it is treated as a variable and unified in the objective function of the proposed algorithm. The proposed algorithm adjusts the similarities by taking into account its capability of capturing the geometry information in both original data space and non-linearly mapped representation space. Meanwhile, the geometry information of original data can be preserved in the embedded representations with the help of the affinity matrix. Section 5.1 reviews the existing representation learning and adaptive graph learning algorithms, and Section 5.2 describes the details of the proposed LELMAE-AN. Section 5.3 experimentally evaluates the effectiveness and efficiency of the proposed algorithm.*



## 5.1 Background and Motivation

The graph-based representation learning algorithm is one of the most common algorithms that taking advantages of geometry information while learning representations. Laplacian eigenmaps (LE) [26] firstly constructs an affinity matrix based on the Euclidean distance between data points and their neighbors. The affinity matrix is then used to minimize the distances between each data point and its neighbors in the representation space. In this thesis, Chapter 3 proposed a representation learning algorithm named as FAE-LG, which can preserve the local and global geometries of original data points from vibration data of motors. Chapter 4 proposed another representation learning algorithm named as LDELM-AE, which preserves the local geometry of input data and exploits the local discrimination information in data representations. Although LE, FAE-LG, and LDELM-AE preserve the geometry information in data representations, it requires a predefined and fixed affinity matrix under an assumed prior knowledge, e.g., uses a  $k$ -nearest neighbor graph with binary or Gaussian edge weights to represent the geometry relationship between data points. However, the assumed prior knowledge might not precisely represent the real geometry relationships between data points. Also, the potential relationship between the affinity matrix and the classes are not fully exploited since the affinity matrix is constructed independently of the following tasks. Furthermore, the existing deep-learning-based algorithms require an iterative training procedure, which is time-consuming.

One way to address the weakness of the predefined affinity matrix is to treat it as a variable and learn it in the training process. The adaptive graph learning algorithm was firstly proposed by Nie et al. [131] in a clustering algorithm. Instead of manually constructing the affinity matrix, the clustering with adaptive neighbors (CAN) [131] adaptively adjusts it during the clustering procedure. CAN is then extended to the projected clustering with adaptive neighbors (PCAN) [131]. PCAN adjusts the affinity matrix based on the linearly projected data points and forces it suitable for the clustering task. T. Liu et al. [75] adaptively adjusted the affinity matrix based on the non-linear data embeddings obtained by an ELM-based algorithm. Inspired by the adaptive neighbor techniques used in the above studies, we can learn the affinity matrix and the non-linear data embeddings simultaneously in representation learning algorithms. Furthermore, to reduce the training time and improve the efficiency of representation learning algorithms, the extreme learning

machine autoencoder (ELM-AE) [13, 14] is proposed. ELM-AE is an extension of ELM and learns data representations based on singular values. The details of ELM-AE is introduced in Section 2.2.2.

In this chapter, a representation learning algorithm, which is named as LELMAE-AN, is proposed. LELMAE-AN consists of two learning parts, which are the affinity matrix learning and representation learning. In the affinity matrix learning, it learns an affinity matrix with its elements represents the similarities of all pairwise data samples. It penalizes the pairwise samples with a far Euclidean distance in both original data space and the embedded representation space. Therefore, the value of similarities is forced to be large if two data samples are close to each other and small if they are far away. In the representation learning, it learns the data representation by minimizing the error between the reconstructed data and the original input data. Since both of the affinity matrix learning and representation learning parts are convex, they can be optimized efficiently.

## 5.2 Proposed Method

Let the training set  $\{\mathbf{X}, \mathbf{Y}\}$  has  $n$  samples, where  $\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_n] \in \mathbb{R}^{d \times n}$  is the data matrix, and  $\mathbf{Y} = [\mathbf{y}_1, \mathbf{y}_2, \dots, \mathbf{y}_n] \in \mathbb{R}^{m \times n}$  contains corresponding labels of each sample. Specifically,  $\mathbf{y}_i$  is a  $m$ -dimensional one-hot column vector that only one element, which correspond to the class of  $i$ -th sample, equals to 1 and the other elements equal to 0.

### 5.2.1 Objective function

Let  $s_{ij}$ , which is an element of the affinity matrix  $\mathbf{S} \in \mathbb{R}^{n \times n}$ , represents the similarity between the sample  $\mathbf{x}_i$  and the sample  $\mathbf{x}_j$ . In this method, the similarity can be treated as a probability by applying the sum-to-one constraint for each column and the nonnegative constraint for each element:

$$\begin{aligned} \sum_{j=1}^n s_{ij} &= 1 \\ 0 &\leq s_{ij} \leq 1 \end{aligned} \tag{5.1}$$

**LELMAE-AN** The non-linearly embedded data representations and the affinity matrix can be simultaneously obtained by jointly minimizing the objective function  $J(\mathbf{S}, \beta)$ :

$$\begin{aligned} J(\mathbf{S}, \beta) &= \frac{1}{2} \sum_{i=1}^n \|\beta \mathbf{h}_i - \mathbf{x}_i\|_2^2 + \frac{C}{2} \|\beta\|_F^2 \\ &\quad + \frac{\alpha_1}{2} \sum_{i,j=1}^n \|\mathbf{x}_i - \mathbf{x}_j\|_2^2 s_{ij} \\ &\quad + \frac{\alpha_2}{2} \sum_{i,j=1}^n \|\beta \mathbf{h}_i - \beta \mathbf{h}_j\|_2^2 s_{ij} \\ &\quad + \frac{\alpha_3}{2} \sum_{i,j=1}^n \|\mathbf{y}_i - \mathbf{y}_j\|_2^2 s_{ij} + \frac{\gamma}{2} s_{ij}^2 \\ s.t. \quad &\sum_{j=1}^n s_{ij} = 1, 0 \leq s_{ij} \leq 1 \end{aligned} \tag{5.2}$$

where  $\mathbf{h}_i = 1/(1 + \exp(-\mathbf{A}\mathbf{x}_i))$  is the output of the hidden layer corresponds to the sample  $\mathbf{x}_i$ .  $\mathbf{A} \in \mathbb{R}^{l \times d}$  is a randomly and orthogonally generated weight matrix, where  $\mathbf{A}^\top \mathbf{A} = \mathbf{I}_d$  if  $d < l$  and  $\mathbf{A}\mathbf{A}^\top = \mathbf{I}_l$  if  $d \geq l$ .  $\mathbf{I}_d$  and  $\mathbf{I}_l$  are the identity matrix with dimension of  $d$  and  $l$  respectively.  $C$ ,  $\alpha_1$ ,  $\alpha_2$ , and  $\alpha_3$  and  $\gamma$  are the trade-off parameters.

The objective function  $J(\mathbf{S}, \boldsymbol{\beta})$  consists of two learning parts, which are the affinity matrix learning and representation learning. The objective function of affinity matrix learning is determined by fixing the projection matrix  $\boldsymbol{\beta}$  in (5.2). It aims to learn similarities of all pairwise data samples in the dataset. The similarity contains local geometry information and discriminative information of input data. Similarly, the objective function of representation learning is determined by fixing the affinity matrix  $\mathbf{S}$  in (5.2). It aims to learn data representations based on the adjusted affinity matrix.

Compared to other ELM-AE based methods [74, 127], which exploited geometry information in data representations by using various predefined affinity matrix, the proposed method unifies the affinity matrix in the objective function as a variable instead of predefining it. Compared to other adaptive graph learning methods that exploited geometry information in original data space, i.e., CAN [131], or in linearly embedded space, i.e., PCAN [131], individually, the proposed method jointly discovers the geometry information in both original space and non-linearly embedded space.

Compared to ELM-CLR proposed by T. Liu et al. [75], the proposed method learns data representations by using reconstruction-based cost function instead of the non-linear random mapping used in ELM-CLR. The reconstruction-based cost function minimizes the error between the original input data and the reconstructed data that restored from data representations. Hence, it forces the embedded data representations to contain useful information and exhibit capability on reconstructing original data samples. Moreover, our method contains a soft discrimination constraint to learn representations with discrimination for classification tasks. The soft discrimination constraint minimizes the multiplication of the similarity between two samples and the difference between the labels of these samples. Since the labels are known, the constraint forces the similarity between two samples to be low if they are in different classes, i.e., the labels are different.

**Affinity matrix learning** In affinity matrix learning, the affinity matrix,  $\mathbf{S}$ , is a variable without any assumption of prior knowledge in LELMAE-AN, and the projection matrix,  $\beta$ , is fixed as a constant.

The objective function  $J_A(\mathbf{S})$  of affinity matrix learning is defined as:

$$\begin{aligned}
 J_A(\mathbf{S}) = & \frac{\alpha_1}{2} \sum_{i,j=1}^n \|\mathbf{x}_i - \mathbf{x}_j\|_2^2 s_{ij} \\
 & + \frac{\alpha_2}{2} \sum_{i,j=1}^n \|\beta \mathbf{h}_i - \beta \mathbf{h}_j\|_2^2 s_{ij} \\
 & + \frac{\alpha_3}{2} \sum_{i,j=1}^n \|\mathbf{y}_i - \mathbf{y}_j\|_2^2 s_{ij} + \gamma s_{ij}^2 \\
 s.t. \quad & \sum_{j=1}^n s_{ij} = 1, 0 \leq s_{ij} \leq 1
 \end{aligned} \tag{5.3}$$

The first term in (5.3) aims to assign a low similarity  $s_{ij}$  for the pairwise data points  $\mathbf{x}_i$  and  $\mathbf{x}_j$  with the large Euclidean distance in the original data space. Similarly, by fixing the projection matrix  $\beta$ , the second term assigns a low similarity  $s_{ij}$  for  $\mathbf{x}_i$  and  $\mathbf{x}_j$  with the large Euclidean distance in the embedded space. Therefore, the affinity matrix offers high similarities for the data samples that are close to each other and low similarities for the data samples that are far away in both original data space and representation space. The similarities will reduce with the Euclidean distance between two samples increasing. The third term assigns a high similarity  $s_{ij}$  for  $\mathbf{x}_i$  and  $\mathbf{x}_j$  in the same class and a small similarity in different classes. The last term in (5.3) prevents the trivial solution of  $\mathbf{S}$ . The constraints force the sum of each row in  $\mathbf{S}$  equals to 1 and the values of  $s_{ij} \forall i, j = 1, \dots, n$  larger than 0, which gives probability property to similarities.

**Representation learning** The data representations are learned by ELM-AE based algorithm. The affinity matrix is used to constrain ELM-AE to preserve geometry information of the original during the representation learning. Hence, in LELMAE-AN, the objective function of the representation learning part  $J_P(\beta)$  is

defined as:

$$\begin{aligned}
 J_P(\boldsymbol{\beta}) = & \frac{1}{2} \sum_{i=1}^n \|\boldsymbol{\beta} \mathbf{h}_i - \mathbf{x}_i\|_2^2 + \frac{C}{2} \|\boldsymbol{\beta}\|_F^2 \\
 & + \frac{\alpha_2}{2} \sum_{i,j=1}^n \|\boldsymbol{\beta} \mathbf{h}_i - \boldsymbol{\beta} \mathbf{h}_j\|_2^2 s_{ij}
 \end{aligned} \tag{5.4}$$

In (5.4), the first term of  $J_P(\boldsymbol{\beta})$  aims to find a output matrix  $\boldsymbol{\beta}$ , which connects the hidden layer and the output layer, to minimize the error between the original input data  $\mathbf{x}_i$  and the reconstructed data  $\hat{\mathbf{x}}_i = \boldsymbol{\beta} \mathbf{h}_i$ . The second term is a regularization term to prevent over-fitting. The third term minimizes the Euclidean distances between data samples in the embedded representation space. The paired data  $\mathbf{x}_i$  and  $\mathbf{x}_j$  with a higher similarity value  $s_{ij}$  should be closer, i.e., with smaller Euclidean distances, after embedded to representations. In the affinity matrix learning,  $\mathbf{S}$  is determined under the assumption that two data points with smaller distance should have a higher similarity. Therefore, the local geometry of input data can be preserved in data representations, i.e., the data samples are close in the original space should be close also in the representation space.

The transpose of the output matrix  $\boldsymbol{\beta}$  is then used to calculate the data representation  $\mathbf{x}_i^{proj}$  for further classification task, and the mathematical formulation is:

$$\mathbf{x}_i^{proj} = \boldsymbol{\beta}^\top \mathbf{x}_i \tag{5.5}$$

### 5.2.2 Optimization

The objective function  $J(\mathbf{S}, \boldsymbol{\beta})$  in (5.2) is minimized by using the alternating optimizing technique. The technique minimizes the objective function with respect to one variable while fixing the others.

**Update affinity matrix  $\mathbf{S}$**  The affinity matrix  $\mathbf{S}$  in (5.2) is updated with the fixed  $\boldsymbol{\beta}$ . Therefore, in this step, minimizing  $J(\mathbf{S}, \boldsymbol{\beta})$  is equal to minimize  $J_A(\mathbf{S})$  in

(5.3). Firstly, we determine some constants for simplicity:

$$\begin{aligned}
d_{ij}^x &= \|\mathbf{x}_i - \mathbf{x}_j\|_2^2 \\
d_{ij}^p &= \|\beta \mathbf{h}_i - \beta \mathbf{h}_j\|_2^2 \\
d_{ij}^y &= \|\mathbf{y}_i - \mathbf{y}_j\|_2^2 \\
d_{ij} &= \alpha_1 d_{ij}^x + \alpha_2 d_{ij}^p + \alpha_3 d_{ij}^y
\end{aligned} \tag{5.6}$$

Hence, the mathematical formulation of minimizing  $J_A(\mathbf{S})$  in (5.3) can be written as:

$$\begin{aligned}
\min_{\mathbf{S}} \quad & \sum_{i,j=1}^n \left( d_{ij} s_{ij} + \gamma s_{ij}^2 \right) \\
s.t. \quad & \sum_{j=1}^n s_{ij} = 1, 0 \leq s_{ij} \leq 1
\end{aligned} \tag{5.7}$$

Since the row vectors  $\{\mathbf{s}_i\}_{i=1}^n$  of the affinity matrix  $\mathbf{S}$  are not correlated to each other, they can be determined separately. Hence, (5.7) can be written as:

$$\begin{aligned}
& \min_{\mathbf{s}_i} \sum_{j=1}^n \left( d_{ij} s_{ij} + \gamma s_{ij}^2 \right) \\
&= \min_{\mathbf{s}_i} \sum_{j=1}^n \left[ \gamma \left( s_{ij} + \frac{1}{2\gamma} d_{ij} \right)^2 - \frac{d_{ij}^2}{4\gamma} \right] \\
&= \min_{\mathbf{s}_i} \sum_{j=1}^n \left( s_{ij} + \frac{1}{2\gamma} d_{ij} \right)^2 \\
&s.t. \quad \mathbf{s}_i \mathbf{1} = 1, \mathbf{s}_i \geq 0
\end{aligned} \tag{5.8}$$

where  $\mathbf{1}$  is a column vector with all elements equal to 1. For  $i$ -th row of the affinity matrix, (5.8) can be vectorized as:

$$\min_{\mathbf{s}_i \mathbf{1}=1, \mathbf{s}_i \geq 0} \left\| \mathbf{s}_i + \frac{1}{2\gamma} \mathbf{d}_i \right\|_2^2 \tag{5.9}$$

where  $\mathbf{s}_i = [s_{i1}, s_{i2}, \dots, s_{in}]$  is the row vector of  $\mathbf{S}$  and  $\mathbf{d}_i = [d_{i1}, d_{i2}, \dots, d_{in}]$ . Furthermore, it is easier to calculate  $\gamma_i$  for each row independently. The overall  $\gamma$  can be determined by the average value of  $\gamma_1, \gamma_2, \dots, \gamma_n$ . Hence, The Lagrangian function of (5.9) is:

$$\mathcal{L}(\mathbf{s}_i, \eta, \boldsymbol{\lambda}) = \frac{1}{2} \left\| \mathbf{s}_i + \frac{1}{2\gamma_i} \mathbf{d}_i \right\|_2^2 - \eta (\mathbf{s}_i \mathbf{1} - 1) - \boldsymbol{\lambda} \mathbf{s}_i^\top \tag{5.10}$$

where  $\eta$  and  $\lambda$  are the Lagrangian multipliers. By applying the Karush–Kuhn–Tucker condition, the optimal solution of  $\mathbf{s}_i$  is:

$$s_{ij}^* = -\frac{d_{ij}}{2\gamma_i} + \eta \quad (5.11)$$

In this work, to reduce the computational complexity, we learn the affinity matrix based on the  $k$ -nearest neighbors of each data point, i.e.,  $\mathbf{s}_i$  has  $k$  nonzero elements. Based on the constraint  $\sum_{j=1}^n s_{ij} = 1$  in (5.7), we get

$$\sum_{j=1}^k \left( -\frac{d_{ij}}{2\gamma_i} + \eta \right) = 1 \quad (5.12)$$

Hence, the analytical solution of  $\eta$  is

$$\eta = \frac{1}{k} + \frac{1}{2k\gamma_i} \sum_{j=1}^k d_{ij} \quad (5.13)$$

Additionally, assuming the distances  $d_{i1}, d_{i2}, \dots, d_{iN}$  are sorted from small to large, and we get

$$\begin{cases} \mathbf{s}_{ij} > 0, & \text{if } j \leq k \\ \mathbf{s}_{ij} = 0, & \text{if } j \geq k+1 \end{cases} \quad (5.14)$$

Therefore, according to (5.11), we get

$$\begin{cases} -\frac{d_{ik}}{2\gamma_i} + \eta > 0 \\ -\frac{d_{i,k+1}}{2\gamma_i} + \eta \leq 0 \end{cases} \quad (5.15)$$

From (5.13) and (5.15), the lower bound and upper bound of  $\gamma_i$  exists as following:

$$\frac{k}{2}d_{ik} - \frac{1}{2} \sum_{j=1}^k d_{ij} < \gamma_i \leq \frac{k}{2}d_{i,k+1} - \frac{1}{2} \sum_{j=1}^k d_{ij} \quad (5.16)$$

Hence, one possible solution for  $\gamma_i$  to obtain  $\mathbf{s}_i$  with exact  $k$  nonzero values can be

$$\gamma_i = \frac{k}{2}d_{i,k+1} - \frac{1}{2} \sum_{j=1}^k d_{ij} \quad (5.17)$$



The optimal solution of  $\mathbf{s}_i$  can be obtained by substituting (5.13) and (5.17) in (5.11):

$$\mathbf{s}_i^* = \frac{d_{i,k+1} - \mathbf{d}_i}{kd_{i,k+1} - \sum_{j=1}^k d_{ij}} \quad (5.18)$$

**Update output matrix  $\beta$**  The output matrix  $\beta$  in (5.2) is updated with the adjusted and fixed  $\mathbf{S}$ . In this step, minimizing  $J(\mathbf{S}, \beta)$  is equal to minimize  $J_P(\beta)$  in (5.4). Therefore, (5.4) can be vectorized as the following:

$$\begin{aligned} J_P(\beta) = & \frac{1}{2} \|\beta \mathbf{H} - \mathbf{X}\|_F^2 + \frac{C}{2} \|\beta\|_F^2 \\ & + \frac{\alpha_2}{2} \text{Tr}(\beta \mathbf{H} \mathbf{L} \mathbf{H}^\top \beta^\top) \end{aligned} \quad (5.19)$$

where  $\mathbf{L} = \mathbf{D} - \mathbf{S}$  is the graph Laplacian.  $\mathbf{D}$  is a diagonal matrix with its diagonal elements  $D_{ii} = \sum_j s_{ij}$ .

As (5.19) is convex, it can be minimized by solving the equation  $\frac{\partial J_P}{\partial \beta} = 0$ . We consider two situations for the optimization problem:

- i The number of samples  $n$  is larger than the number of hidden neurons  $l$ ;
- ii The number of samples  $n$  is not larger than the number of hidden neurons  $l$ ;

In the case of  $n > l$ ,  $\beta$  can be minimized by solving  $\frac{\partial J_P}{\partial \beta} = 0$ :

$$\begin{aligned} \frac{\partial J_P}{\partial \beta} = & (\beta \mathbf{H} - \mathbf{X}) \mathbf{H}^\top + C\beta + \alpha_2 \beta \mathbf{H} \mathbf{L} \mathbf{H}^\top \\ = & \beta (C\mathbf{I}_l + \mathbf{H} \mathbf{H}^\top + \alpha_2 \mathbf{H} \mathbf{L} \mathbf{H}^\top) - \mathbf{X} \mathbf{H}^\top \end{aligned} \quad (5.20)$$

Hence, the optimal solution of  $\beta$  is

$$\beta^* = \mathbf{X} \mathbf{H}^\top (C\mathbf{I}_l + \mathbf{H} \mathbf{H}^\top + \alpha_2 \mathbf{H} \mathbf{L} \mathbf{H}^\top)^{-1} \quad (5.21)$$

Furthermore, in the case of  $n \leq l$ , we introduce an additional variable  $\mathbf{a} \in \mathbb{R}^{d \times N}$ , which is smaller than  $\beta \in \mathbb{R}^{d \times l}$ , to optimize (5.19) efficiently. Specifically, we let  $\beta = \mathbf{a} \mathbf{H}^\top$ , and solve  $\mathbf{a}$  with a smaller dimensionality instead of solving  $\beta$ . Therefore,

(5.19) can be written as:

$$J_P(\mathbf{a}) = \frac{1}{2} \|\mathbf{a}\mathbf{H}^\top \mathbf{H} - \mathbf{X}\|_F^2 + \frac{C}{2} \|\mathbf{a}\mathbf{H}^\top\|_F^2 + \frac{\alpha_2}{2} \text{Tr}(\mathbf{a}\mathbf{H}^\top \mathbf{H} \mathbf{L} \mathbf{H}^\top \mathbf{H} \mathbf{a}^\top) \quad (5.22)$$

Next,  $\beta$  can be minimized by solving  $\frac{\partial J_P}{\partial \mathbf{a}} = 0$ :

$$\begin{aligned} \frac{\partial J_P}{\partial \mathbf{a}} &= [\mathbf{a}\mathbf{H}^\top \mathbf{H} - \mathbf{X} + C\mathbf{a} + \alpha_2 \mathbf{a}\mathbf{H}^\top \mathbf{H} \mathbf{L}] \mathbf{H}^\top \mathbf{H} \\ &= \mathbf{a}\mathbf{H}^\top \mathbf{H} - \mathbf{X} + C\mathbf{a} + \alpha_2 \mathbf{a}\mathbf{H}^\top \mathbf{H} \mathbf{L} \\ &= \mathbf{a}(C\mathbf{I}_N + \mathbf{H}^\top \mathbf{H} + \alpha_2 \mathbf{H}^\top \mathbf{H} \mathbf{L}) - \mathbf{X} \end{aligned} \quad (5.23)$$

Therefore, the optimal solution of  $\mathbf{a}$  is

$$\mathbf{a}^* = \mathbf{X}(C\mathbf{I}_N + \mathbf{H}^\top \mathbf{H} + \alpha_2 \mathbf{H}^\top \mathbf{H} \mathbf{L})^{-1} \quad (5.24)$$

and

$$\beta^* = \mathbf{X}(C\mathbf{I}_N + \mathbf{H}^\top \mathbf{H} + \alpha_2 \mathbf{H}^\top \mathbf{H} \mathbf{L})^{-1} \mathbf{H}^\top \quad (5.25)$$

In summary, the optimal solution of  $\beta$  is

$$\beta^* = \begin{cases} \mathbf{X}\mathbf{H}^\top (C\mathbf{I}_l + \mathbf{H}\mathbf{H}^\top + \alpha_2 \mathbf{H}\mathbf{L}\mathbf{H}^\top)^{-1}, & \text{if } n > l \\ \mathbf{X}(C\mathbf{I}_N + \mathbf{H}^\top \mathbf{H} + \alpha_2 \mathbf{H}^\top \mathbf{H} \mathbf{L})^{-1} \mathbf{H}^\top, & \text{if } n \leq l \end{cases} \quad (5.26)$$

**Complete training process** The complete LELMAE-AN algorithm is described in Algorithm 4. Based on (5.18), we initialize each row of the affinity matrix  $\mathbf{s}_i$  as following:

$$\mathbf{s}_i^* = \frac{d_{i,k+1} - \mathbf{d}_i^x}{kd_{i,k+1} - \sum_{j=1}^k d_{ij}} \quad (5.27)$$

where  $\mathbf{d}_i^x = [d_{i1}^x, d_{i2}^x, \dots, d_{in}^x]$  contains the Euclidean distances between the data point  $\mathbf{x}_i$  and the other data points in the original data space. After that, the output matrix  $\beta$  and the affinity matrix are updated iteratively until they are converged. It usually takes 2 to 5 iterations before the convergence. The learned output matrix  $\beta$  is then used to obtain data representations by  $\mathbf{X}_{proj} = \beta^\top \mathbf{X}$ , and the representations  $\mathbf{X}_{proj}$  can be used for further classification tasks. In this study, we use the linear regression to map the representations  $\mathbf{X}_{proj}$  to the predicted label

---

**Algorithm 4** The LELMAE-AN algorithm.

---

**Inputs:** The training data  $\{\mathbf{X}, \mathbf{Y}\} = \{\mathbf{x}_i, \mathbf{y}_i\}_{i=1}^n$ , the number of hidden neurons  $l$ , the number of nearest neighbors  $k$ , the trade-off parameters  $C$ ,  $\alpha_1$ ,  $\alpha_2$  and  $\alpha_3$ .

**Outputs:** The affinity matrix  $\mathbf{S}$  and the output matrix  $\beta$ .

- 1: **for**  $i=1:n$  **do**
- 2:     Initialize the  $i$ -th row of  $\mathbf{S}$  by using (5.27), where  $\mathbf{d}_i^x$  is a vector with its  $j$ -th element  $d_{ij}^x = \|\mathbf{x}_i - \mathbf{x}_j\|_2^2$ .
- 3: **end for**
- 4: Compute  $\mathbf{L} = \mathbf{D} - \mathbf{S}$ , where  $D_{ii} = \sum_j s_{ij}$ .
- 5: Randomly initiate the input weights  $\mathbf{A}$  and compute  $\mathbf{h}_i = \frac{1}{1+\exp(-\mathbf{A}\mathbf{x}_i)}$ .
- 6: **while** not converge **do**
- 7:     Update  $\beta$  by using (5.26).
- 8:     **for**  $i=1:n$  **do**
- 9:         Update the  $i$ -th row of  $\mathbf{S}$  by using (5.18), where  $\mathbf{d}_i$  is a vector with its  $j$ -th element  $d_{ij} = \alpha_1\|\mathbf{x}_i - \mathbf{x}_j\|_2^2 + \alpha_2\|\beta\mathbf{h}_i - \beta\mathbf{h}_j\|_2^2 + \alpha_3\|\mathbf{y}_i - \mathbf{y}_j\|_2^2$ .
- 10:     **end for**
- 11: **end while**
- 12: **return**  $\mathbf{S}, \beta$

---

$\tilde{\mathbf{Y}}$ . The mathematical formulation of the linear regression is:

$$\min_{\beta_T} \|\beta_T \mathbf{X}_{proj} - \mathbf{Y}\|_F^2 + \frac{C_T}{2} \|\beta_T\|_F^2 \quad (5.28)$$

where  $\beta_T \in \mathbb{R}^{d \times m}$  is the output weights between the hidden layer and output layer, and  $C_T$  is the hyper-parameter.  $\mathbf{Y} = [\mathbf{y}_1, \mathbf{y}_2, \dots, \mathbf{y}_n] \in \mathbb{R}^{m \times n}$  contains one-hot labels of the input data.

## 5.3 Experiments

The proposed representation learning algorithm, LELMAE-AN, is compared with the related algorithms on several benchmark datasets. It also applied on a bearing fault dataset to validate its effectiveness on the machine fault diagnosis tasks.

### 5.3.1 Datasets Descriptions

The real-world benchmark datasets used to test the proposed algorithm include four UCI datasets, two objective recognition image datasets, and a machine fault diagnosis dataset. In details, the four UCI datasets [128], i.e., IRIS, WINE, LIVER, SATIMAGE, are low-dimensional datasets. The two objective recognition image datasets are: 1) The Columbia University Image Library dataset of 20 classes (COIL20) [129] contains 20 different objects with 72 different poses of each object; 2) The USPST dataset, which is a subset of the handwriting recognition dataset USPS [130], contains ten classes of gray-scale handwritten digit images. Furthermore, a bearing vibration dataset, i.e., Case Western Reserve University (CWRU) bearing dataset, is used to validate the effectiveness of LELMAE-AN on machine fault diagnosis tasks. The details of the datasets are summarized in Table 4.1.

### 5.3.2 Experimental settings

The classification accuracy (ACC) is used as the evaluation metric in this study of the proposed method and the other related methods. ACC measures the percentage of the correctly classified data points among all of the data points, and it is defined as:

$$ACC = \frac{\sum_i^n \delta(\mathbf{y}_i, \hat{\mathbf{y}}_i)}{n} \quad (5.29)$$

where  $\mathbf{y}_i$  and  $\hat{\mathbf{y}}_i$  are the ground truth and predicted label of  $\mathbf{x}_i$ . The function  $\delta(\mathbf{y}_i, \hat{\mathbf{y}}_i)$  is defined as:

$$\delta(\mathbf{y}_i, \hat{\mathbf{y}}_i) = \begin{cases} 1, & \text{if } \mathbf{y}_i = \hat{\mathbf{y}}_i \\ 0, & \text{otherwise} \end{cases} \quad (5.30)$$

In UCI and image datasets, we compared the proposed method with the following relative methods:

- 1) AE and SAE [54].
- 2) ELM-AE and ML-ELM [13].
- 3) GELM-AE and ML-GELM [74].
- 4) GDELM-AE and SGD-ELM [127].
- 5) Deep belief network (DBN) [9].

To evaluate the effectiveness of LELMAE-AN on the machine fault diagnosis tasks, this study tested the proposed method on the CWRU bearing dataset and compared the performance with the state-of-the-art methods in this area, and the methods are listed as follows:

- 1) Discrete wavelet transform (DWT) + SVM [117];
- 2) Ensemble empirical mode decomposition (EEMD) + SVM [118];
- 3) SAE [119];
- 4) SDA [21];
- 5) SFAE-LG [132];

For fair comparisons, all parameter settings were standardized in this study. The training and test data were divided by using 2-fold cross-validation for UCI and image datasets and 5-fold cross-validation, where 20% of the dataset is used for training and the other 80% for the test, for CWRU bearing dataset. Furthermore, all of the hyper-parameters were selected by using 4-fold cross-validation. The ranges of the parameters are listed in Table 5.1. Specifically, for all methods, the number of hidden neurons is selected from 100 to 5000 with an incremental of 100. The number of neighbors used to construct and learn the affinity matrix is selected from 1 to 15. The corrupt rate and sparsity of SAE and SDA are selected from 0.05 to 0.8 with an interval of 0.05. Moreover, all of the other hyper-parameters are selected from the exponential sequence  $[1e - 10, 1e - 9, \dots, 1e9, 1e10]$ . All methods were tested repetitively for 50 times to reduce the influences of the randomness.

TABLE 5.1: Hyper-parameters selection range for cross-validation.

Algorithms	Hyper-parameters	Range
	# of Hidden neurons	100 - 3000
	# of Neighbors	1 - 20
ELM-AE and ML-ELM	C	1e-10 - 1e10
AE, SAE and SDA	Learning rate	0.001
	Corrupt rate	0.05 - 0.8
	Sparsity	0.05 - 0.8
GELM-AE, GDELM-AE, ML-GELM and SGD-ELM	C	1e-10 - 1e10
	$\lambda$	1e-10 - 1e10
	$\alpha_{AE}$	0 - 1e10
	$\alpha_G$	0 - 1e10
SFAE-LG	$\alpha_L$	0 - 1e10
	$C_X$	1e-10 - 1e10
	$C_T$	1e-10 - 1e10
SVM	$C$	1e-10 - 1e10
	$\gamma$	1e-10 - 1e10
	$C$	1e-10 - 1e10
LELMAE-AN	$\alpha_1$	1e-10 - 1e10
	$\alpha_2$	1e-10 - 1e10
	$\alpha_3$	1e-10 - 1e10

### 5.3.3 The convergence and effect of the affinity matrix

**Convergence** Firstly, we investigated the convergence of the proposed method in all of the benchmark datasets. Figure 5.1 shows how the value of  $\|\mathbf{S}\|_2^2$  changing with iterations proceeding. It is noticed that the proposed method is converged after about 4 to 8 iterations for the different dataset. Hence, the proposed method converges fast and will not increase much training complexity compared to conventional affinity matrix constructing methods.

**Comparison between the learned and manually constructed affinity matrix** In conventional affinity matrix constructing methods, e.g., locality preserving projection (LPP) [49], Laplacian eigenmaps (LE) [26] and GELM-AE [74], the similarities are usually determined by a  $k$ -nearest neighbor graph with binary or

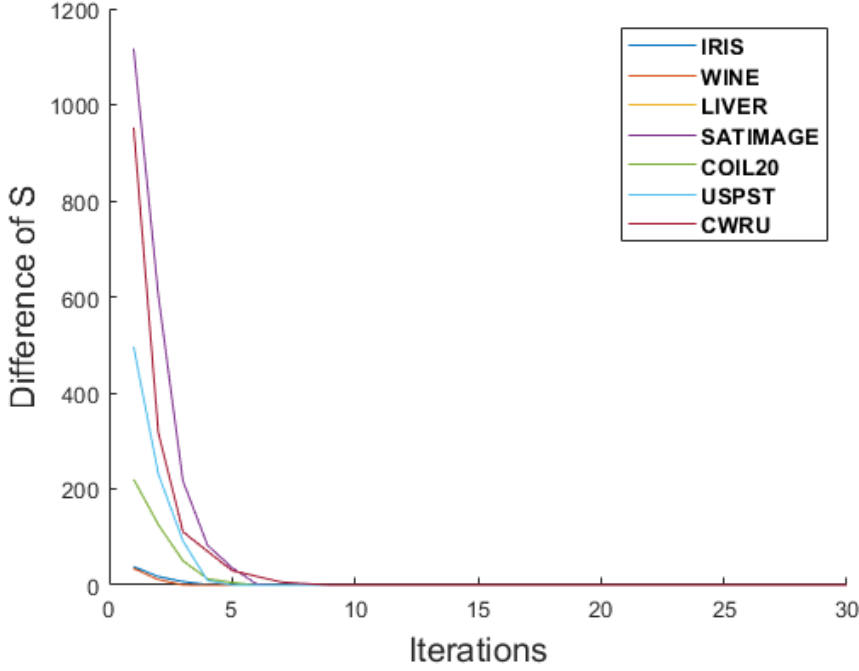
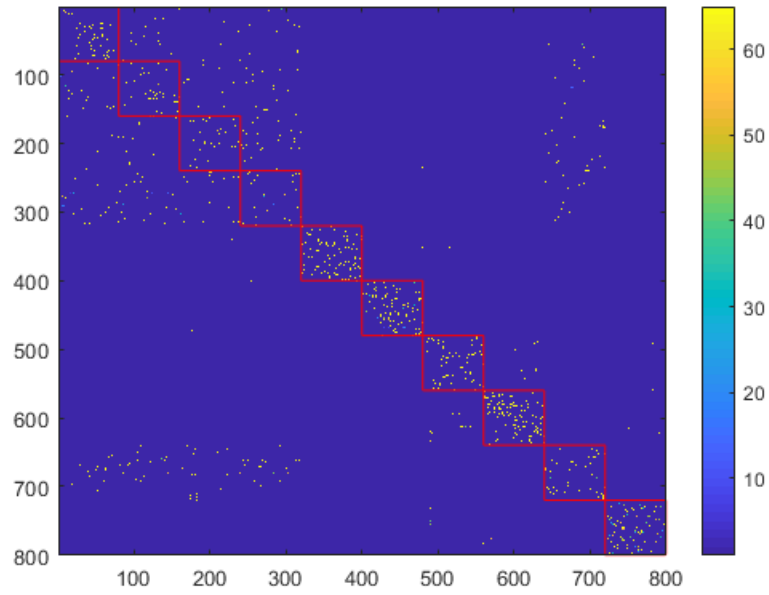


FIGURE 5.1: The value changes in  $\|\mathcal{S}\|_2^2$  with iterations.

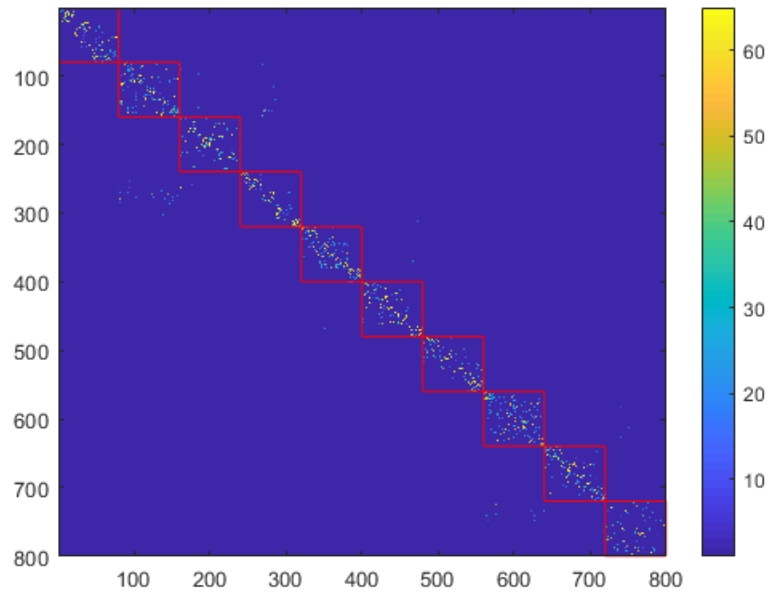
Gaussian edge weights. In this study, we used the CWRU dataset to demonstrate the difference between the learned and manually constructed affinity matrix. Figure 5.2a shows the affinity matrix constructed by the heat function with  $k = 5$ , and the mathematical formulation of the Gaussian function is:

$$s_{ij} = \exp\left(-\frac{\|\mathbf{x}_i - \mathbf{x}_j\|_2^2}{2\sigma^2}\right) \quad (5.31)$$

Figure 5.2b shows the affinity matrix learned by the proposed method. The red rectangles in Figure 5.2 indicate the class of the samples, i.e., the samples in the same rectangle belong to the same class. We noticed that the affinity matrix learned by LELMAE-AN shows a higher discriminability than the constructed affinity matrix. Specifically, the affinity matrix in Figure 5.2b is a block diagonal matrix, and it shows higher similarities between the samples in the same class (in red rectangles). Hence, the proposed method learned an affinity matrix that can both discover geometry and discriminative information.



(A) Constructed affinity matrix



(B) Learned affinity matrix

FIGURE 5.2: Comparison between the learned and manually constructed affinity matrix.



### 5.3.4 Experimental results on the UCI and image datasets

**UCI datasets** The four UCI datasets, i.e., IRIS, WINE, LIVER, SATIMAGE, used in this study were randomly divided into two disjoint equal parts: one used as the training data and another used as the test data. The hyper-parameters were selected by a 4-fold cross-validation from the values listed in Table 5.1.

The columns 2 to 5 of Table 5.2 summarized experimental results on the UCI datasets. The deep models, which include DBN, ML-ELM, SAE, SDA, ML-GELM, and SGD-ELM, are used three hidden layers in the experiments. The other models, including the proposed LELMAE-AN, are single hidden layer models. It can be seen that the proposed method achieved the best results on all UCI datasets. The average accuracy among four datasets is 91.95% compared to the second-highest average accuracy 90.65% achieved by SGD-ELM. Moreover, the standard deviation of the proposed method is lower than the relative methods, i.e., LELMAE-AN achieved 0.4 standard deviation among UCI datasets compared to the second-highest standard deviation 0.45 achieved by ELM-AE.



(A) COIL20



(B) USPST

FIGURE 5.3: Sample images from the image datasets.

TABLE 5.2: The comparison of classification accuracies on UCI and image datasets.

Methods	IRIS	WINE	LIVER	SATIMAGE	COIL20	USPST
DBN	96.67 + 1.44	96+1.83	59.65+0.53	86.2+0.66	98.85+0.09	88.96+0.81
AE	94.16+1.23	88.37+0.77	55.02+2.68	83.16+1.58	58.86+3.87	85.89+1.67
SAE	95.07+2.95	90.78+0.54	61.97 + 3.96	87.67+1.33	98.81+0.2	93.69+0.33
ELM-AE	84.44+0.63	91.87+0.9	54.55+0.12	84.57+0.15	59.39+4.66	83.95+0.47
GELM-AE	92.44+3.94	96.37+0.78	56.83+3.52	80.93+2.64	61.78+4.99	87.22+0.61
GDELM-AE	93.16+0.52	98.11+0.28	60.91+1.86	84.29+1.04	68.47+3.69	88.76+0.42
LDELM-AE	98.59+0.72	98.85+0.64	70.78+1.21	90.98+0.19	99.36+0.21	92.05+0.13
ML-ELM	93.73+2.78	94.56+1.85	58.03+3.61	85.52+0.53	96.58+0.7	89.78+1.58
ML-GELM	97.15+0.54	96.67+0.48	70.08+2.96	88.77+0.44	99.9+0.99	93.72+0.48
SGD-ELM	98.58+0.78	99.09+0.96	73.8+1.26	91.13+0.2	99.85+0.15	93.98+0.55
ML-LDELM	99.2+0.68	99.64+0.54	73.91+2.16	92.19+0.23	99.86+0.06	94.39+0.54
LELMAE-AN	<b>99.67+0.46</b>	<b>99.71+0.41</b>	<b>75.21+0.54</b>	<b>93.22+0.21</b>	<b>99.96+0.18</b>	<b>94.73+0.37</b>

### 5.3.5 Experimental results on COIL20 dataset

We then tested the proposed algorithm on the COIL20 dataset. The COIL20 is an objective classification dataset, which contains 1440 images from 20 different objects. Each object of COIL20 has 72 images taken from different views. Figure 5.3a shows samples of the COIL20 dataset. For the proposed algorithm, we set the number of hidden neurons  $l = 1500$  and the number of neighbors  $k = 5$ . The other hyper-parameters were chosen by cross-validation with  $\alpha_1 = 1$ ,  $\alpha_2 = 1$ ,  $\alpha_3 = 1$ , and  $C = 1$ .

The classification accuracies on COIL20 dataset are listed in column 6 of Table 5.2. We noticed that our method achieved the best performance among the single hidden layer methods in respect of both average accuracy and standard deviation. While comparing to the deep structure methods, the proposed method achieved a comparable accuracy of 99.96% compared to the best accuracy 99.9% by ML-GELM.

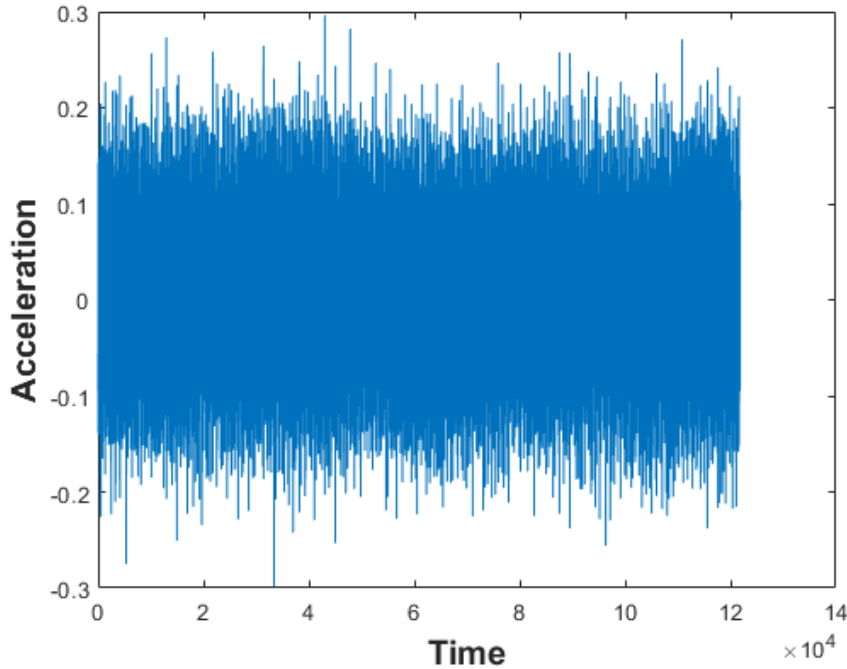


FIGURE 5.4: Vibration signal collected in 10 seconds.

### 5.3.6 Experimental results on USPST dataset

The USPST is a handwriting recognition dataset, which contains 2007 handwriting digit images. Each of the samples is a  $16 \times 16$  grey-scale image from ten different handwriting digits, i.e., 0 to 9. An example of the dataset is shown in Figure 5.3b. For the proposed algorithm, the number of hidden neurons is chosen as 2000 and the other hyper-parameters are  $\alpha_1 = 0.01$ ,  $\alpha_2 = 0.1$ ,  $\alpha_3 = 0.1$ ,  $k = 5$  and  $C = 1$ .

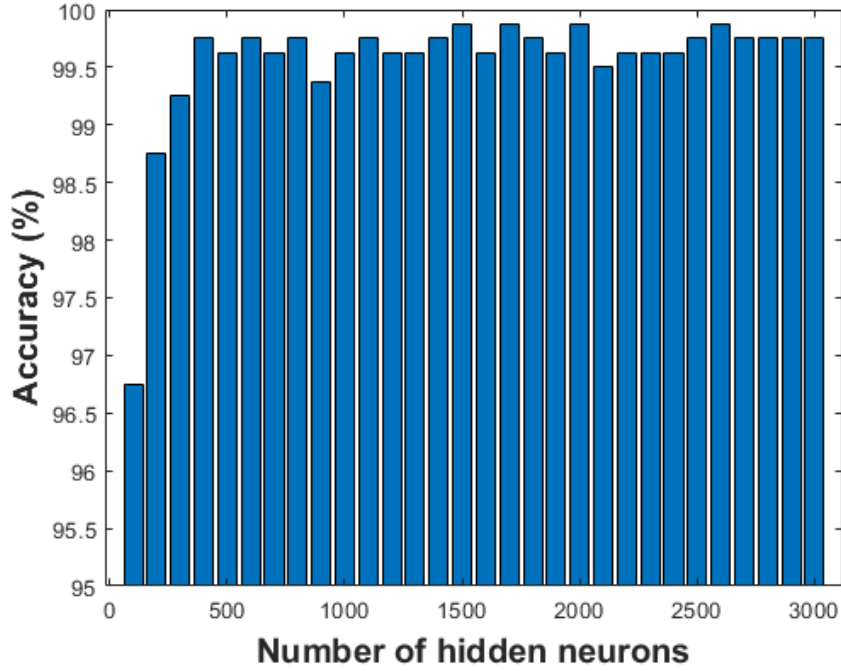


FIGURE 5.5: CWRU dataset motor diagnosis results using various numbers of hidden neurons.

The last column of Table 5.2 shows the performances on USPST dataset. It can be observed that the proposed LELMAE-AN outperformed the other methods with the accuracy of 94.73%. Also, it achieved a better or comparable standard deviation of 0.37 on USPST dataset.

### 5.3.7 Experimental results on CWRU dataset

To evaluate the effectiveness of the proposed method on machine fault diagnosis tasks, we evaluated it on the CWRU dataset. The samples in the CWRU dataset are high-frequency time-series signals, as shown in Figure 5.4. For a high-frequency

signal, both time-domain and frequency-domain information are essential. Hence, we transformed the time-series signal into the time-frequency domain and learned data representations from the time-frequency domain signals.

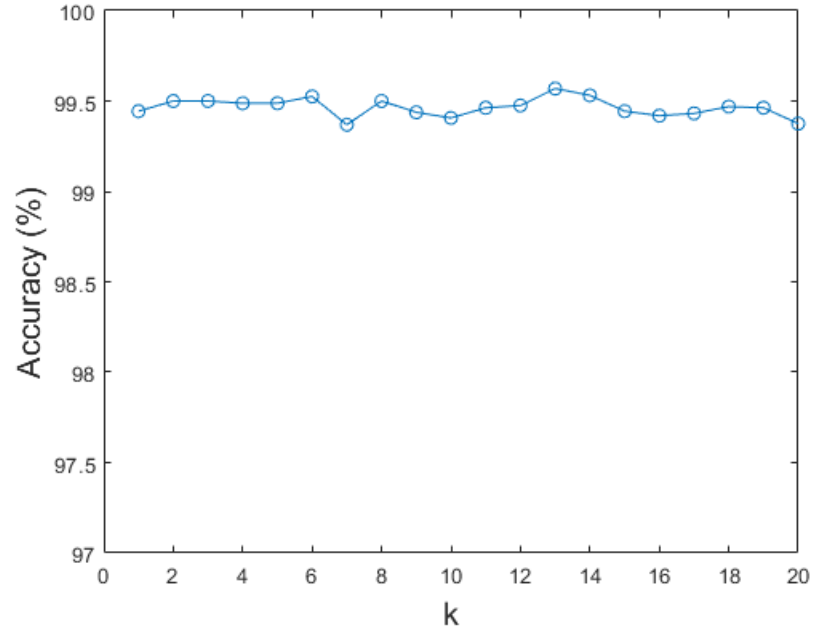
### 5.3.7.1 Sensitivity analysis of hyper-parameters

Firstly, we analyze the sensitivity of hyper-parameters on CWRU bearing dataset.

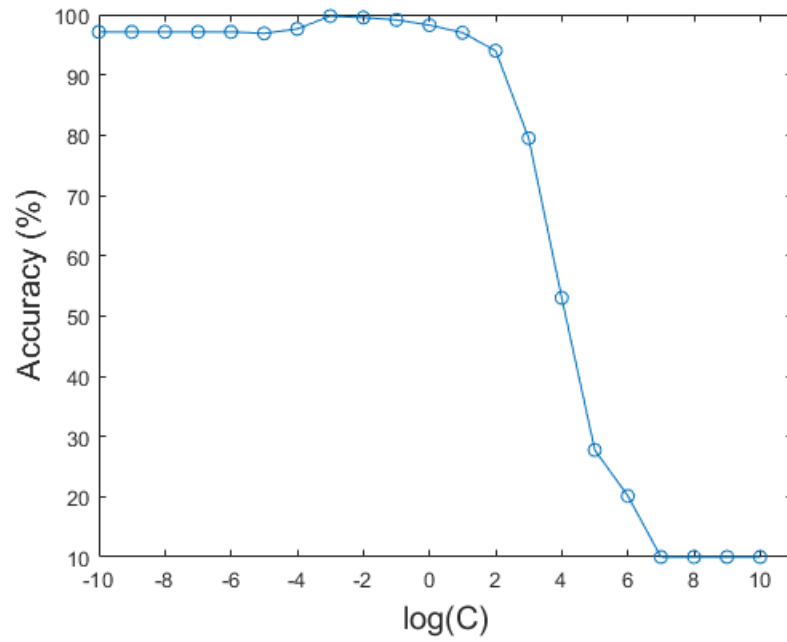
**Number of hidden neurons  $l$**  We investigated the influence of the number of hidden neurons. Figure 5.5 shows the influence of the number of hidden neurons on the diagnostic accuracies, and we observed that the accuracy increases with the number of hidden neurons rises. It can also be noticed that accuracies reached stability while using 400 or more hidden neurons. In other word, the proposed method is not sensitive to the number of hidden neurons with  $l \geq 400$ . Hence, we used the number of hidden neurons  $l = 400$  in our study to reduce the computational complexity.

**Number of neighbors  $k$**  From (5.17), it can be noticed that the value of parameter  $\gamma$  is determined based on the number of neighbors  $k$ . Therefore, we further investigated the effect of different values of  $k$  on the performance of the proposed method. The diagnostic results using different number of neighbors are plotted in Figure 5.6a, and it shows that the proposed method is not sensitive to the number of neighbors  $k$ . In this work, we used  $k = 5$  for the proposed LELMAE-AN.

**Trade-off parameters  $C$ ,  $\alpha_1$ ,  $\alpha_2$  and  $\alpha_3$**  After fixed the number of hidden neurons and neighbors, we investigated the effect of trade-off parameters. Figure 5.6b shows the diagnostic accuracies with various values of the trade-off parameter  $C$ . It can be observed that while the value of  $C$  is in the range from  $1e - 10$  to  $1e2$ , the diagnostic accuracies are stable, and the accuracy decreases rapidly with the value of  $C$  larger than  $1e3$ . It is worth noting that the accuracy achieves the peak value from  $C = 1e - 3$  to  $C = 1$ . Moreover, the trade-off parameters  $\alpha_1$  and  $\alpha_2$  relate to two geometric information discovering terms. Hence, we investigated



(A) Numbers of neighbors

(B)  $C$ FIGURE 5.6: CWRU dataset motor diagnosis results using various values of hyper-parameters  $k$  and  $C$ .

them together in Figure 5.7a. It can be noticed that the accuracy decreases rapidly while the value of  $\alpha_2$  is too large, and the proposed method achieves the highest accuracy when  $\alpha_2 = 100$ . We also observed that the diagnostic accuracies are not sensitive to  $\alpha_1$  while  $\alpha_2 = 100$  and  $\alpha_1 > 1e-5$ . Similarly, the effects of the trade-off parameter  $\alpha_3$  on the diagnostic accuracies are shown in Figure 5.7b. We noticed that the overall accuracies are not sensitive to  $\alpha_3$ , but there is a 0.3% improvement while  $\alpha_3$  is larger than  $1e4$ .

### 5.3.7.2 Soft and hard discrimination constraint

We then analyzed the effect of the proposed soft discrimination constraint on machine fault diagnosis tasks. Conventionally, the discriminative information is contained in the affinity matrix by using a hard constraint, which is formulated as:

$$s_{ij} = 0, \quad \text{if } \mathbf{y}_i \neq \mathbf{y}_j \quad (5.32)$$

In the proposed method, the discriminative information is learned by using the soft discrimination constraint, which is unified in the objective function of LELMAE-AN. Hence, instead of constructing the discriminative information in the affinity matrix independently, the proposed method adjusts the affinity matrix to comprise the discriminative information by jointly minimizing the soft discrimination constraint and other cost functions. We compared the performances of using the soft and hard discrimination constraint in Figure 5.8. It can be observed that using the soft discrimination constraint performs better than using the hard discrimination constraint in LELMAE-AN, especially while the number of training data is small.

### 5.3.7.3 Data number for training

Figure 5.9 shows the diagnostic performances on CWRU dataset with various sizes of training data. It is natural to assume that a better performance can be achieved with more training data. In Figure 5.9, we observed that the average accuracy increased, and the standard deviation decreased while the number of training data increases. This observation confirms the previous assumption. Furthermore, the proposed method can achieve a 99.88% accuracy and a 0.14 standard deviation

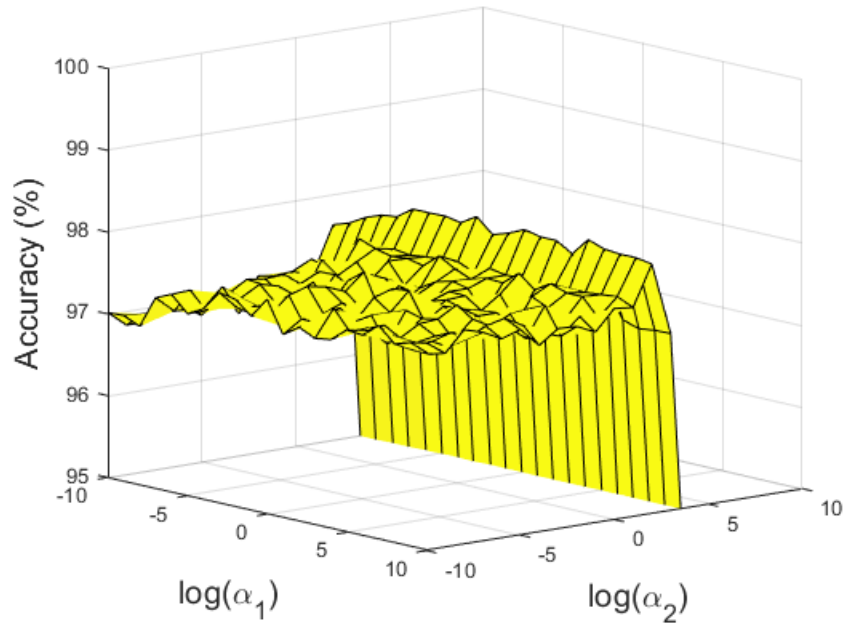
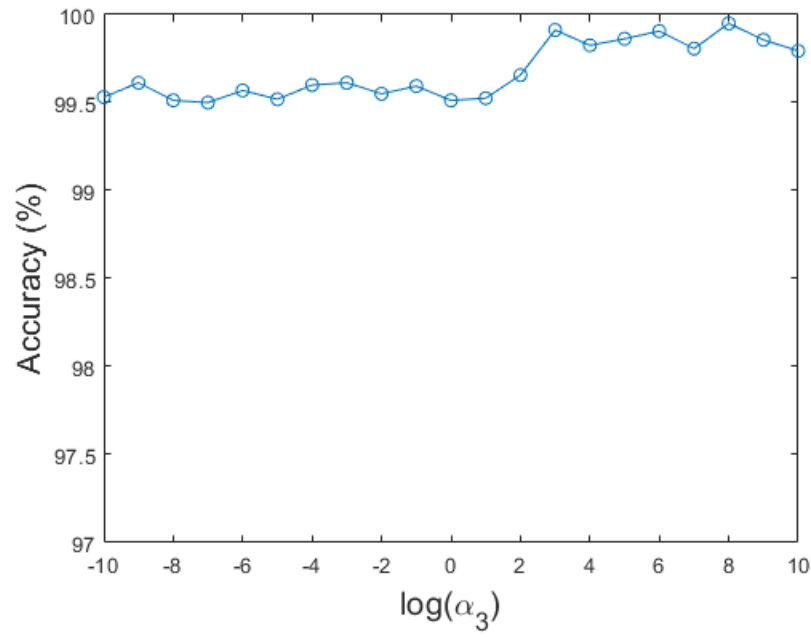
(A)  $\alpha_1$  and  $\alpha_2$ (B)  $\alpha_3$ 

FIGURE 5.7: CWRU dataset motor diagnosis results using various values of hyper-parameters  $\alpha_1$ ,  $\alpha_2$  and  $\alpha_3$ .



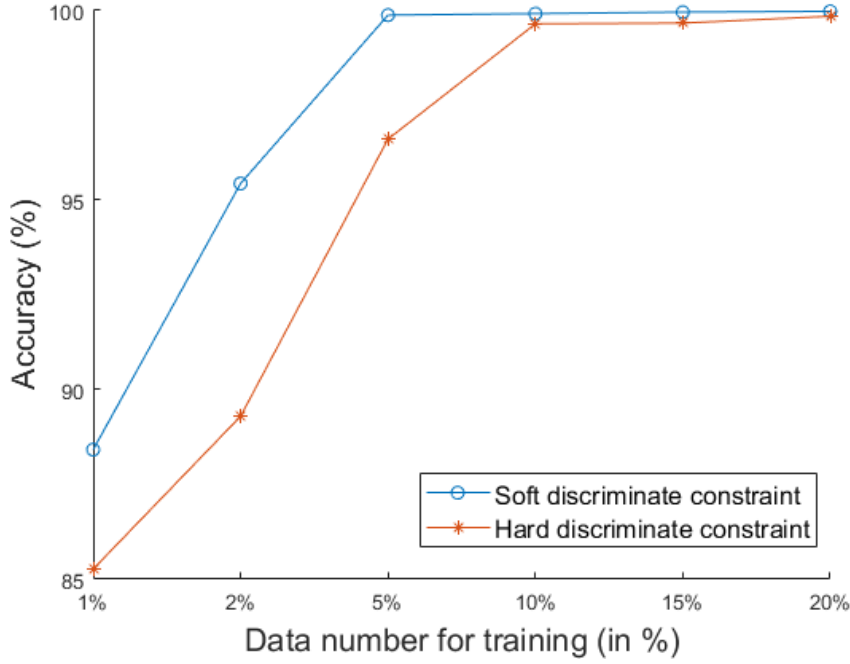


FIGURE 5.8: CWRU dataset motor diagnosis results using the soft and hard discrimination constraint in LELMAE-AN.

with 5% of total samples for training. Hence, LELMAE-AN can perform well on machine fault diagnosis tasks with a small number of training data.

#### 5.3.7.4 Comparison with state-of-the-art methods

TABLE 5.3: The comparison of classification accuracies on CWRU bearing dataset.

Methods	# of training samples (%)	# of classes	ACC (%)
DWT+SVM [117]	75	10	88.9
EEMD+SVM [118]	40	11	97.91
SAE [119]	40	4	94.4
SDA [21]	40	4	95.58
SFAE-LG	20	10	97.29
LDELM-AE	20	10	99.74
LELMAE-AN	5	10	<b>99.88</b>

Table 5.3 reports the comparison of the performances between the proposed method and published state-of-the-art methods on CWRU dataset. In [117], the features were extracted by using the discrete wavelet transform, and they were applied to

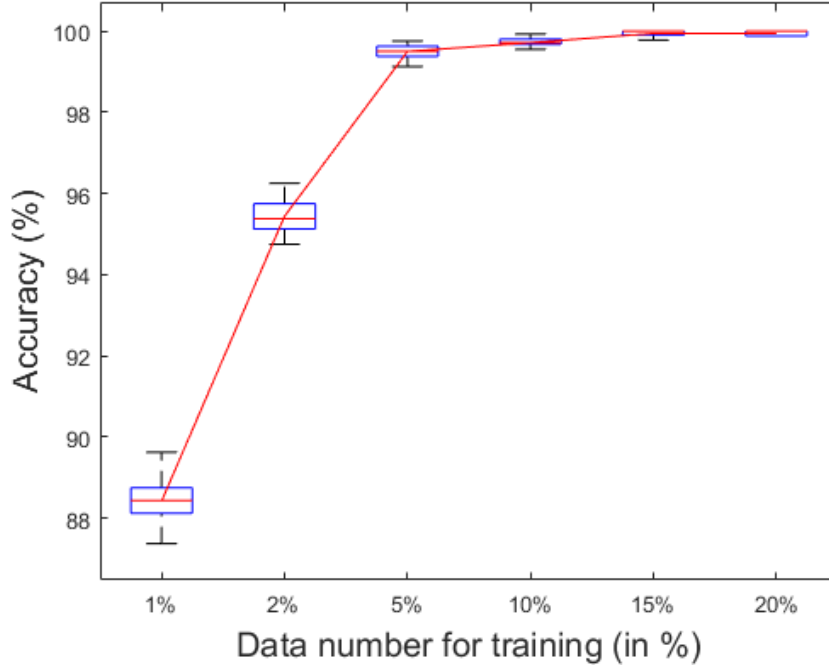


FIGURE 5.9: CWRU dataset motor diagnosis results using various percentages of samples as the training data.

SVM to classify ten machine health conditions. An accuracy of 88.9 % was achieved by using 75% of total samples as the training data. The same classifier, i.e., SVM, was used in [118], but the features were extracted by using the empirical mode decomposition. By using 40% samples in the training process, [118] achieved the accuracy of 97.91%. Other than separately extracted features and classified the machine health conditions, end-to-end methods that learned features and classified health conditions together were applied in machine fault diagnosis tasks. For example, SAE in [119] and SDA in [21] were used to classify four health conditions, and the accuracies 94.4% and 95.58% were achieved, separately. Moreover, in Chapter 3 of this thesis, a method named SFAE-LG, is proposed to preserve both local and global geometries of the input data in the learned representations. It achieved the accuracy of 97.29% by using 20% of samples in training. Also, Chapter 4 introduces another method named LDELM-AE, which preserves the local geometry of the input data and exploits the local discrimination information in data representations. It achieved the accuracy of 99.74% with 20% of samples are used for training. Compared with the above state-of-the-art methods, the proposed LELMAE-AN achieved a higher accuracy of 99.88% by using lesser training data, i.e., using only 5% of total samples as the training data.

## 5.4 Summary

This chapter address the second objective of the thesis, to exploit and preserve geometry information of input data while learning data representations. The representation learning algorithms FAE-LG and LDELM-AE introduced in Chapter 3 and Chapter 4, respectively, exploit and preserve geometry information of input data by using a predefined and fixed affinity matrix. In this chapter, an adaptable affinity matrix without assumed prior knowledge is used to exploit geometry information in data representations. Section 5.1 introduces the necessity of using an adaptable affinity matrix in the representation learning algorithm, and Section 5.2 describes the details of the proposed LELMAE-AN. The proposed algorithm exploits the geometry information from both data and representation space by using an adaptable affinity matrix. The affinity matrix also obtains the discriminative information by a soft discrimination constraint. The input data is efficiently mapped to the non-linear representation space by using ELM-AE based method that is constrained by the learned affinity matrix. Section 5.3 experimentally evaluates the proposed algorithm on various standard benchmark datasets, as well as on the dataset of machine fault diagnosis task. It performs better than the other state-of-the-art methods on these datasets. The experimental results also prove that the proposed method can achieve high diagnostic accuracy with less training samples, i.e., 5% of all the dataset. Hence, LELMAE-AN is an effective algorithm to solve machine fault diagnosis tasks.

# Chapter 6

## Conclusions and Future Works

*Chapter 6 summarizes the contributions based on Chapter 3, Chapter 4, and Chapter 5, and it also draws conclusions regarding to how this thesis addresses the two research objectives that introduced in Chapter 1. This chapter then suggests future research directions.*

## 6.1 Conclusions

This thesis has introduced three representation learning algorithms to address two research objective, which are

- To efficiently learn data representations that can improve the performance of supervised classification with an application for bearing fault diagnosis.
- To exploit and preserve geometry information of input data while learning data representations.

The details of the contributions and discoveries of this thesis are state as the following.

*(1) The local geometry can be preserved from input data space to representation space by using the graph-based cost function.*

Graph-based method treats each data sample as the vertex of the graph and constructs an affinity matrix consists of the weight of edges to reveal the geometry information between each pair of data samples. In my first work, the proposed FAE-LG exploits the local geometry by minimizing the Euclidean distance between each data point and its nearest neighbor in the representation space. This is equivalent to use the graph-based method to preserve the local geometry from input data to data representations, which only the weight between each data sample and the nearest neighbor of it equals 1 in the affinity matrix. In the second work, the proposed LDELM-AE minimizes the sum of Euclidean distances between each data point and  $k_w$ -nearest neighbors of it within the same class. The affinity matrix used in this algorithm is defined by applying the heat function to the pairwise data samples. In my last work, the proposed LELMAE-AN exploits the local geometry by using an adaptable affinity matrix. It simultaneously learns data representations and the affinity matrix by minimizing a unified objective function.

*(2) The global geometry of input data can be preserved in data representations by minimizing the difference between the embedded representations and the random projected input data.*

In the first work, it theoretically proved that the Euclidean distances among all input data are preserved in representation space without distorting by more than

a factor of  $1 \pm \epsilon$  for any  $0 < \epsilon < 0.5$ . This is achieved by minimizing the difference between the embedded data representations and the random projected input data. Also, it experimentally showed the non-linear activation can approximate such preservation.

*(3) Separability maximizing in the representation space is essential for classification tasks, and it can be achieved by utilizing label information in representation learning.*

For classification tasks, such as the bearing fault diagnosis, it is necessary to learn discriminative data representations. Therefore, this thesis investigated to maximize the separability of data representations. In FAE-LG, the discrimination of data representations is achieved by minimizing the error between the predicted labels and the ground-truth while minimizing the reconstruction cost function, which is used to learn data representations. LDELM-AE exploits the local discrimination by maximizing the sum of Euclidean distances between the margin data points and their  $k_b$ -nearest neighbors in the different classes. Additionally, LELMAE-AN introduces a soft discrimination constraint, which is jointly minimized with the objective function, to obtain the discriminative affinity matrix. The affinity matrix is then used to obtain data representations with separability maximized. The experimental results on several benchmark datasets showed that maximizing separability is essential for classification tasks.

*(4) The objective function that unified the discrimination cost function benefit to raise the training efficiency of the deep learning-based representation learning algorithms.*

In FAE-LG, the discrimination cost function is minimized together with the other cost functions to observe the discriminative data representations with local and global geometry preserved. It is also discovered that by using the discrimination cost function, the proposed FAE-LG does not require an additional fine-tuning step to obtain discriminative representations. Therefore, the proposed algorithm reduces training time, compared with traditional deep learning-based algorithms, e.g., SAE, which uses a two-step training process. Moreover, FAE-LG requires fewer hidden neurons in each layer and thus has less training and test time compared with SAEs. Hence, in deep learning-based representation learning algorithms, it is beneficial to utilize the label information to reduce the computational

complexity while learning representations for supervised classification tasks.

*(5) ELM-based algorithms benefit the representation learning task in the aspect of increasing training efficiency.*

In order to increase the training efficiency of representation learning algorithms, the ELM-based algorithms were investigated in this thesis. The algorithms introduced in the second and third works, i.e., LDELM-AE and LELMAE-AN, utilize the random neurons proposed by ELM. Since the hidden neurons of ELM and ELM-based representation learning algorithms, e.g., ELM-AE, LDELM-AE, and LELMAE-AN, are randomly generated and fixed, the training process only solves the projection matrix. Therefore, the training time of ELM-based algorithms is much shorter than the deep learning-based representation learning algorithms. The experiments on multiple benchmark datasets demonstrated that LDELM-AE and LELMAE-AN achieve comparable classification accuracies with a better training efficiency compared to deep learning-based algorithms. Hence, the second and third works further address the first research objective, i.e., to efficiently learn data representations that can improve the performance of machine fault diagnosis tasks.

## 6.2 Future Works

Although this thesis investigated various aspects to preserve local and geometry information in data representations and improve the computational efficiency of representation learning algorithms, there are still many research gaps waiting to be filled:

- i. This thesis investigated to exploit and preserve geometry information in data representations to improve the performance of supervised classification tasks. It is interesting to learn data representations with local and global geometry exploited for other machine learning tasks, e.g., semi-supervised classification, clustering, etc.
- ii. Other than geometry information, this thesis also investigated to use ELM-based algorithms to learn non-linear structure-preserving data representations for supervised classification efficiently. Hence, it is also worthwhile to

investigate the use of ELM-based representation learning for other machine learning tasks.

- iii. Although ELM-based algorithms benefit the representation learning tasks, it requires enough number of hidden neurons to achieve satisfying performance. Therefore, it is natural to investigate whether using feature selection techniques can improve the performance and reduce the computational complexity of ELM-based representation learning algorithms.
- iv. This thesis investigated to preserve the global geometry by keeping the consistency of the pair-wise Euclidean distances of all data points in representation space from the original data space. However, the Euclidean distance may not be suitable for other applications. Hence, it is interesting to investigate more generalized methods to preserve the global geometry.
- v. The proposed representation learning algorithms require much time to determine hyper-parameters. Hence, it is worthwhile to investigate methods that can select hyper-parameters efficiently.





# List of Author's Publications

## Journal Articles

- **Yue Li**, Chamara Kasun Liyanaarachchi Lekamalage, Tianchi Liu, Pinan Chen, Guang-Bin Huang, “Learning representations with local and global geometries preserved for machine fault diagnosis”, *IEEE Transactions on Industrial Electronics.*, 67(3):2360-70, 2019.
- **Yue Li**, Yijie Zeng, Yuanyuan Qing, Guang-Bin Huang, “Learning Local Discriminative Representations via Extreme Learning Machine for Machine Fault Diagnosis”, *Neurocomputing*, minor revision.
- **Yue Li**, Yijie Zeng, Tianchi Liu, Xiaofan Jia, Guang-Bin Huang, “Simultaneously Learning Affinity Matrix and Data Representations for Machine Fault Diagnosis”, *Neural Networks*, 122:395-406, 2020.
- Yijie Zeng, **Yue Li**, Jichao Chen, Xiaofan Jia, Guang-Bin Huang, “ELM Embedded Discriminative Dictionary Learning for Image Classification”, *Neural Networks*, 123: 331-342, 2020.

## Conference Proceedings

- **Yue Li**, Tianchi Liu, Guang-Bin Huang, “Room Occupancy Estimation using Sparse Linear Discriminant Analysis and Extreme Learning Machine”, *The International Conference on Extreme Learning Machines (ELM2016)*, Singapore, 2016.

- Tianchi Liu, **Yue Li**, Bai Zuo, Jaydeep De, Cao Vinh Le, Zhiping Lin, Shih-Hsiang Lin, Guang-Bin Huang, and Dongshun Cui. "Two-stage structured learning approach for stable occupancy detection." *Neural Networks (IJCNN), 2016 International Joint Conference on. IEEE, 2016.*

# Bibliography

- [1] Christopher M Bishop. *Pattern Recognition and Machine Learning*. springer, 2006.
- [2] Jan L Talmon and Peter McNair. The effect of noise and biases on the performance of machine learning algorithms. *International Journal of Biomedical Computing*, 31(1):45–57, 1992.
- [3] B Samanta and KR Al-Balushi. Artificial neural network based fault diagnostics of rolling element bearings using time-domain features. *Mech. Syst. Signal Process*, 17(2):317–328, 2003.
- [4] F Al-Badour, M Sunar, and L Cheded. Vibration analysis of rotating machinery using time–frequency analysis and wavelet techniques. *Mech. Syst. Signal Process*, 25(6):2083–2101, 2011.
- [5] Jian-Hua Zhong, Pak Kin Wong, and Zhi-Xin Yang. Fault diagnosis of rotating machinery based on multiple probabilistic classifiers. *Mech. Syst. Signal Process*, 108:99–114, 2018.
- [6] Maximilian Christ, Nils Braun, Julius Neuffer, and Andreas W Kempa-Liehr. Time series feature extraction on basis of scalable hypothesis tests (tsfresh—a python package). *Neurocomputing*, 307:72–77, 2018.
- [7] Yoshua Bengio, Aaron Courville, and Pascal Vincent. Representation learning: A review and new perspectives. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(8):1798–1828, 2013.
- [8] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in Neural Information Processing Systems*, pages 1097–1105, 2012.
- [9] Geoffrey E Hinton and Ruslan R Salakhutdinov. Reducing the dimensionality of data with neural networks. *Science*, 313(5786):504–507, 2006.
- [10] Geoffrey E Hinton, Simon Osindero, and Yee-Whye Teh. A fast learning algorithm for deep belief nets. *Neural Computation*, 18(7):1527–1554, 2006.
- [11] David E Rumelhart, Geoffrey E Hinton, and Ronald J Williams. Learning internal representations by error propagation. Technical report, California Univ San Diego La Jolla Inst for Cognitive Science, 1985.

- [12] Yoshua Bengio et al. Learning deep architectures for ai. *Foundations and Trends® in Machine Learning*, 2(1):1–127, 2009.
- [13] Liyanaarachchi Lekamalage Chamara Kasun, Hongming Zhou, Guang-Bin Huang, and Chi Man Vong. Representational learning with extreme learning machine for big data. *IEEE Intelligent Systems*, 28(6):31–34, 2013.
- [14] Liyanaarachchi Lekamalage Chamara Kasun, Yan Yang, Guang-Bin Huang, and Zhengyou Zhang. Dimension reduction with extreme learning machine. *IEEE Trans. Image Process.*, 25(8):3906–3918, 2016.
- [15] Quoc Le and Tomas Mikolov. Distributed representations of sentences and documents. In *Proceedings of the 31st International Conference on Machine Learning (ICML-14)*, pages 1188–1196, 2014.
- [16] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 770–778, 2016.
- [17] Geoffrey Hinton, Li Deng, Dong Yu, George E Dahl, Abdel-rahman Mohamed, Navdeep Jaitly, Andrew Senior, Vincent Vanhoucke, Patrick Nguyen, Tara N Sainath, et al. Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups. *IEEE Signal Processing Magazine*, 29(6):82–97, 2012.
- [18] Meng Gan, Cong Wang, et al. Construction of hierarchical diagnosis network based on deep learning and its application in the fault pattern recognition of rolling element bearings. *Mechanical Systems and Signal Processing*, 72: 92–104, 2016.
- [19] Yaguo Lei, Feng Jia, Jing Lin, Saibo Xing, and Steven X Ding. An intelligent fault diagnosis method using unsupervised feature learning towards mechanical big data. *IEEE Transactions on Industrial Electronics*, 63(5):3137–3147, 2016.
- [20] Nishchal K Verma, Vishal Kumar Gupta, Mayank Sharma, and Rahul Kumar Sevakula. Intelligent condition based monitoring of rotating machines using sparse auto-encoders. In *Prognostics and Health Management (PHM), 2013 IEEE Conference on*, pages 1–7. IEEE, 2013.
- [21] Chen Lu, Zhen-Ya Wang, Wei-Li Qin, and Jian Ma. Fault diagnosis of rotary machinery components using a stacked denoising autoencoder-based health state identification. *Signal Processing*, 130:377–388, 2017.
- [22] Gregory W Vogl, Brian A Weiss, and Moneer Helu. A review of diagnostic and prognostic capabilities and best practices for manufacturing. *Journal of Intelligent Manufacturing*, pages 1–17, 2016.

- [23] Bo Li, M-Y Chow, Yodyium Tipsuwan, and James C Hung. Neural-network-based motor rolling bearing fault diagnosis. *IEEE Trans. Ind. Electron.*, 47(5):1060–1069, 2000.
- [24] James Manyika. *The Internet of Things: Mapping the value beyond the hype*. McKinsey Global Institute, 2015.
- [25] Sam T Roweis and Lawrence K Saul. Nonlinear dimensionality reduction by locally linear embedding. *Science*, 290(5500):2323–2326, 2000.
- [26] Mikhail Belkin and Partha Niyogi. Laplacian eigenmaps and spectral techniques for embedding and clustering. In *Adv. Neural Inf. Process. Syst.*, pages 585–591, 2002.
- [27] Peter N. Belhumeur, João P Hespanha, and David J. Kriegman. Eigenfaces vs. fisherfaces: Recognition using class specific linear projection. *IEEE Trans. Pattern Anal. Mach. Intell.*, 19(7):711–720, 1997.
- [28] Joshua B Tenenbaum, Vin De Silva, and John C Langford. A global geometric framework for nonlinear dimensionality reduction. *Science*, 290(5500):2319–2323, 2000.
- [29] Guang-Bin Huang and Lei Chen. Enhanced random search based incremental extreme learning machine. *Neurocomputing*, 71(16-18):3460–3468, 2008.
- [30] Guang-Bin Huang and Lei Chen. Convex incremental extreme learning machine. *Neurocomputing*, 70(16-18):3056–3062, 2007.
- [31] Guang-Bin Huang, Qin-Yu Zhu, and Chee-Kheong Siew. Extreme learning machine: theory and applications. *Neurocomputing*, 70(1-3):489–501, 2006.
- [32] Guang-Bin Huang, Qin-Yu Zhu, Chee-Kheong Siew, et al. Extreme learning machine: a new learning scheme of feedforward neural networks. *Neural Networks*, 2:985–990, 2004.
- [33] Guang-Bin Huang, Lei Chen, Chee Kheong Siew, et al. Universal approximation using incremental constructive feedforward networks with random hidden nodes. *IEEE Trans. Neural Networks*, 17(4):879–892, 2006.
- [34] Guang-Bin Huang, Xiaojian Ding, and Hongming Zhou. Optimization method based extreme learning machine for classification. *Neurocomputing*, 74(1-3):155–163, 2010.
- [35] Guang-Bin Huang, Hongming Zhou, Xiaojian Ding, and Rui Zhang. Extreme learning machine for regression and multiclass classification. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, 42(2):513–529, 2011.
- [36] Guang-Bin Huang, Ming-Bin Li, Lei Chen, and Chee-Kheong Siew. Incremental extreme learning machine with fully complex hidden nodes. *Neurocomputing*, 71(4-6):576–583, 2008.

- [37] Guang-Bin Huang. An insight into extreme learning machines: random neurons, random features and kernels. *Cognitive Computation*, 6(3):376–390, 2014.
- [38] Guang-Bin Huang. What are extreme learning machines? filling the gap between frank rosenblatt’s dream and john von neumann’s puzzle. *Cognitive Computation*, 7(3):263–278, 2015.
- [39] D. Cui, G. Huang, L. L. C. Kasun, G. Zhang, and W. Han. Elmanet: Feature learning using extreme learning machines. In *2017 IEEE International Conference on Image Processing (ICIP)*, pages 1857–1861, Sep. 2017. doi: 10.1109/ICIP.2017.8296603.
- [40] Harold Hotelling. Analysis of a complex of statistical variables into principal components. *Journal of Educational Psychology*, 24(6):417, 1933.
- [41] Ian Jolliffe. *Principal component analysis*. Springer, 2011.
- [42] Matthew A Turk and Alex P Pentland. Face recognition using eigenfaces. In *Proceedings. 1991 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 586–591. IEEE, 1991.
- [43] Ronald A Fisher. The use of multiple measurements in taxonomic problems. *Annals of Eugenics*, 7(2):179–188, 1936.
- [44] Hua Yu and Jie Yang. A direct lda algorithm for high-dimensional data—with application to face recognition. *Pattern Recognition*, 34(10):2067–2070, 2001.
- [45] Mikhail Belkin and Partha Niyogi. Laplacian eigenmaps for dimensionality reduction and data representation. *Neural Computation*, 15(6):1373–1396, 2003.
- [46] Mikhail Belkin and Partha Niyogi. Convergence of laplacian eigenmaps. In *Advances in Neural Information Processing Systems*, pages 129–136, 2007.
- [47] David L Donoho and Carrie Grimes. Hessian eigenmaps: Locally linear embedding techniques for high-dimensional data. *Proceedings of the National Academy of Sciences*, 100(10):5591–5596, 2003.
- [48] Marzia Polito and Pietro Perona. Grouping and dimensionality reduction by locally linear embedding. In *Advances in Neural Information Processing Systems*, pages 1255–1262, 2002.
- [49] Xiaofei He and Partha Niyogi. Locality preserving projections. In *Advances in Neural Information Processing Systems*, pages 153–160, 2004.
- [50] Weiwei Yu, Xiaolong Teng, and Chongqing Liu. Face recognition using discriminant locality preserving projections. *Image and Vision Computing*, 24(3):239–248, 2006.

- [51] Xiaofei He, Deng Cai, and Wanli Min. Statistical and computational analysis of locality preserving projection. In *Proceedings of the 22nd International Conference on Machine learning*, pages 281–288. ACM, 2005.
- [52] Hervé Bourlard and Yves Kamp. Auto-association by multilayer perceptrons and singular value decomposition. *Biological Cybernetics*, 59(4-5):291–294, 1988.
- [53] Geoffrey E Hinton and Richard S Zemel. Autoencoders, minimum description length and helmholtz free energy. In *Advances in Neural Information Processing Systems*, pages 3–10, 1994.
- [54] Yoshua Bengio, Pascal Lamblin, Dan Popovici, and Hugo Larochelle. Greedy layer-wise training of deep networks. In *Advances in Neural Information Processing Systems*, pages 153–160, 2007.
- [55] Hugo Larochelle and Yoshua Bengio. Classification using discriminative restricted boltzmann machines. In *Proceedings of the 25th International Conference on Machine learning*, pages 536–543. ACM, 2008.
- [56] Christopher Poultney, Sumit Chopra, Yann L Cun, et al. Efficient learning of sparse representations with an energy-based model. In *Advances in Neural Information Processing Systems*, pages 1137–1144, 2007.
- [57] Ian Goodfellow, Honglak Lee, Quoc V Le, Andrew Saxe, and Andrew Y Ng. Measuring invariances in deep networks. In *Advances in Neural Information Processing Systems*, pages 646–654, 2009.
- [58] Pascal Vincent, Hugo Larochelle, Yoshua Bengio, and Pierre-Antoine Manzagol. Extracting and composing robust features with denoising autoencoders. In *Proceedings of the 25th International Conference on Machine learning*, pages 1096–1103. ACM, 2008.
- [59] Pascal Vincent, Hugo Larochelle, Isabelle Lajoie, Yoshua Bengio, and Pierre-Antoine Manzagol. Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion. *Journal of Machine Learning Research*, 11(Dec):3371–3408, 2010.
- [60] Pascal Vincent. A connection between score matching and denoising autoencoders. *Neural Computation*, 23(7):1661–1674, 2011.
- [61] Salah Rifai, Pascal Vincent, Xavier Muller, Xavier Glorot, and Yoshua Bengio. Contractive auto-encoders: Explicit invariance during feature extraction. In *Proceedings of the 28th International Conference on International Conference on Machine Learning*, pages 833–840. Omnipress, 2011.
- [62] Fan RK Chung and Fan Chung Graham. *Spectral graph theory*. American Mathematical Soc., 1997.



- [63] Shuicheng Yan, Dong Xu, Benyu Zhang, and Hong-Jiang Zhang. Graph embedding: A general framework for dimensionality reduction. In *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, volume 2, pages 830–837. IEEE, 2005.
- [64] Gao Huang, Shiji Song, Jatinder ND Gupta, and Cheng Wu. Semi-supervised and unsupervised extreme learning machines. *IEEE Transactions on Cybernetics*, 44(12):2405–2417, 2014.
- [65] Kun Han, Dong Yu, and Ivan Tashev. Speech emotion recognition using deep neural network and extreme learning machine. In *Fifteenth Annual Conference of the International Speech Communication Association*, 2014.
- [66] Zhan-Li Sun, Tsan-Ming Choi, Kin-Fan Au, and Yong Yu. Sales forecasting using extreme learning machine with applications in fashion retailing. *Decision Support Systems*, 46(1):411–419, 2008.
- [67] Can Wan, Zhao Xu, Pierre Pinson, Zhao Yang Dong, and Kit Po Wong. Probabilistic forecasting of wind power generation using extreme learning machine. *IEEE Transactions on Power Systems*, 29(3):1033–1044, 2013.
- [68] Abdul Adeel Mohammed, Rashid Minhas, QM Jonathan Wu, and Maher A Sid-Ahmed. Human face recognition based on multidimensional pca and extreme learning machine. *Pattern Recognition*, 44(10-11):2588–2597, 2011.
- [69] Weiwei Zong and Guang-Bin Huang. Face recognition based on extreme learning machine. *Neurocomputing*, 74(16):2541–2551, 2011.
- [70] Nan-Ying Liang, Paramasivan Saratchandran, Guang-Bin Huang, and Narasimhan Sundararajan. Classification of mental tasks from eeg signals using extreme learning machine. *International Journal of Neural Systems*, 16(01):29–38, 2006.
- [71] Jiexiong Tang, Chenwei Deng, Guang-Bin Huang, and Baojun Zhao. Compressed-domain ship detection on spaceborne optical image using deep neural network and extreme learning machine. *IEEE Transactions on Geoscience and Remote Sensing*, 53(3):1174–1185, 2014.
- [72] Wanyu Deng, Qinghua Zheng, and Lin Chen. Regularized extreme learning machine. In *2009 IEEE Symposium on Computational Intelligence and Data Mining*, pages 389–395. IEEE, 2009.
- [73] José M MartíNez-MartíNez, Pablo Escandell-Montero, Emilio Soria-Olivas, José D MartíN-Guerrero, Rafael Magdalena-Benedito, and Juan Gómez-Sanchis. Regularized extreme learning machine for regression problems. *Neurocomputing*, 74(17):3716–3721, 2011.
- [74] Kai Sun, Jianshe Zhang, Chunxia Zhang, and Junying Hu. Generalized extreme learning machine autoencoder and a new deep neural network. *Neurocomputing*, 230:374–381, 2017.

- [75] Tianchi Liu, Chamara Kasun Liyanaarachchi Lekamalage, Guang-Bin Huang, and Zhiping Lin. An adaptive graph learning method based on dual data representations for clustering. *Pattern Recognition*, 77:126–139, 2018.
- [76] Peter L Bartlett. The sample complexity of pattern classification with neural networks: the size of the weights is more important than the size of the network. *IEEE Transactions on Information Theory*, 44(2):525–536, 1998.
- [77] Mikhail Belkin, Partha Niyogi, and Vikas Sindhwani. Manifold regularization: A geometric framework for learning from labeled and unlabeled examples. *J. Mach. Learn. Res.*, 7(Nov):2399–2434, 2006.
- [78] Bojan Mohar, Y Alavi, G Chartrand, and OR Oellermann. The laplacian spectrum of graphs. *Graph theory, Combinatorics, and Applications*, 2(871-898):12, 1991.
- [79] Ky Fan. On a theorem of weyl concerning eigenvalues of linear transformations i. *Proceedings of the National Academy of Sciences of the United States of America*, 35(11):652, 1949.
- [80] PF Albrecht, JC Appiarius, RM McCoy, EL Owen, and DK Sharma. Assessment of the reliability of motors in utility applications-updated. *IEEE Transactions on Energy conversion*, pages 39–46, 1986.
- [81] Bostjan Dolenc, Pavle Boskoski, and Juricic. Distributed bearing fault diagnosis based on vibration analysis. *Mechanical Systems and Signal Processing*, 66:521–532, 2016.
- [82] Naipeng Li, Yaguo Lei, Jing Lin, and Steven X Ding. An improved exponential model for predicting remaining useful life of rolling element bearings. *IEEE Transactions on Industrial Electronics*, 62(12):7762–7773, 2015.
- [83] N Tandon and A Choudhury. A review of vibration and acoustic measurement methods for the detection of defects in rolling element bearings. *Tribology International*, 32(8):469–480, 1999.
- [84] D Bently. Predictive maintenance through the monitoring and diagnostics of rolling element bearings. *Bently Nevada Co., Application Note*, 44:2–8, 1989.
- [85] J Mathew and RJ Alfredson. The condition monitoring of rolling element bearings using vibration analysis. *Journal of Vibration, Acoustics, Stress, and Reliability in Design*, 106(3):447–453, 1984.
- [86] C James Li and Jun Ma. Wavelet decomposition of vibrations for detection of bearing-localized defects. *Ndt & E International*, 30(3):143–149, 1997.
- [87] MS Patil, Jose Mathew, PK Rajendrakumar, and Sandeep Desai. A theoretical model to predict the effect of localized defect on vibrations associated with ball bearing. *International Journal of Mechanical Sciences*, 52(9):1193–1201, 2010.

- [88] Xiaoyuan Zhang and Jianzhong Zhou. Multi-fault diagnosis for rolling element bearings based on ensemble empirical mode decomposition and optimized support vector machines. *Mechanical Systems and Signal Processing*, 41(1-2):127–140, 2013.
- [89] Bubathi Muruganatham, MA Sanjith, B Krishnakumar, and SAV Satya Murty. Roller element bearing fault diagnosis using singular spectrum analysis. *Mechanical Systems and Signal Processing*, 35(1-2):150–166, 2013.
- [90] Xiaohang Jin, Mingbo Zhao, Tommy WS Chow, and Michael Pecht. Motor bearing fault diagnosis using trace ratio linear discriminant analysis. *IEEE Transactions on Industrial Electronics*, 61(5):2441–2451, 2014.
- [91] Muhammad Amar, Iqbal Gondal, and Campbell Wilson. Vibration spectrum imaging: A novel bearing fault classification approach. *IEEE Transactions on Industrial Electronics*, 62(1):494–502, 2015.
- [92] Abdullah M Al-Ghamd and David Mba. A comparative experimental study on the use of acoustic emission and vibration analysis for bearing defect identification and estimation of defect size. *Mechanical Systems and Signal Processing*, 20(7):1537–1571, 2006.
- [93] Saad Al-Dossary, RI Raja Hamzah, and David Mba. Observations of changes in acoustic emission waveform for varying seeded defect sizes in a rolling element bearing. *Applied Acoustics*, 70(1):58–81, 2009.
- [94] DH Pandya, SH Upadhyay, and Suraj Prakash Harsha. Fault diagnosis of rolling element bearing with intrinsic mode function of acoustic emission data using apf-knn. *Expert Systems with Applications*, 40(10):4137–4145, 2013.
- [95] Katsuhiko Shibata, Atsushi Takahashi, and Takuya Shirai. Fault diagnosis of rotating machinery through visualisation of sound signals. *Mechanical Systems and Signal Processing*, 14(2):229–241, 2000.
- [96] M Amarnath, V Sugumaran, and Hemantha Kumar. Exploiting sound signals for fault diagnosis of bearings using decision tree. *Measurement*, 46(3):1250–1256, 2013.
- [97] Jing Lin. Feature extraction of machine sound using wavelet and its application in fault diagnosis. *NDT & e International*, 34(1):25–30, 2001.
- [98] RBW Heng and Mohd Jailani Mohd Nor. Statistical analysis of sound and vibration signals for monitoring rolling element bearing condition. *Applied Acoustics*, 53(1-3):211–226, 1998.
- [99] Bo-Suk Yang, Fengshou Gu, Andrew Ball, et al. Thermal image enhancement using bi-dimensional empirical mode decomposition in combination with relevance vector machine for rotating machinery fault diagnosis. *Mechanical Systems and Signal Processing*, 38(2):601–614, 2013.

- [100] Randy R Schoen, Thomas G Habetler, Farrukh Kamran, and RG Bartfield. Motor bearing damage detection using stator current monitoring. *IEEE Transactions on Industry Applications*, 31(6):1274–1279, 1995.
- [101] Yassine Amirat, Vincent Choqueuse, and Mohamed Benbouzid. Eemd-based wind turbine bearing failure detection using the generator stator current homopolar component. *Mechanical Systems and Signal Processing*, 41(1-2): 667–678, 2013.
- [102] Jafar Zarei and Javad Poshtan. An advanced park’s vectors approach for bearing fault detection. *Tribology International*, 42(2):213–219, 2009.
- [103] Turker Ince, Serkan Kiranyaz, Levent Eren, Murat Askar, and Moncef Gabbouj. Real-time motor fault detection by 1-d convolutional neural networks. *IEEE Trans. Ind. Electron.*, 63(11):7067–7075, 2016.
- [104] David I Shuman, Sunil K Narang, Pascal Frossard, Antonio Ortega, and Pierre Vandergheynst. The emerging field of signal processing on graphs: Extending high-dimensional data analysis to networks and other irregular domains. *IEEE Signal Process. Mag.*, 30(3):83–98, 2013.
- [105] William B Johnson and Joram Lindenstrauss. Extensions of lipschitz mappings into a hilbert space. *Contemp. Math.*, 26(189-206):1, 1984.
- [106] Peter Frankl and Hiroshi Maehara. The johnson-lindenstrauss lemma and the sphericity of some graphs. *J. Comb. Theory, Series B*, 44(3):355–362, 1988.
- [107] Jorge Nocedal. Updating quasi-newton matrices with limited storage. *Math. Comput.*, 35(151):773–782, 1980.
- [108] Mark Schmidt. minfunc: unconstrained differentiable multivariate optimization in matlab. *Software available at <http://www.cs.ubc.ca/~schmidtm/Software/minFunc.htm>*, 2005.
- [109] Xinsheng Lou and Kenneth A Loparo. Bearing fault diagnosis based on wavelet transform and fuzzy inference. *Mech. Syst. Signal Process*, 18(5): 1077–1095, 2004.
- [110] Hai Qiu, Jay Lee, Jing Lin, and Gang Yu. Wavelet filter-based weak signature detection method and its application on rolling element bearing prognostics. *J. Sound. Vib.*, 289(4):1066–1090, 2006.
- [111] Jaouher Ben Ali, Nader Fnaiech, Lotfi Saidi, Brigitte Chebel-Morello, and Farhat Fnaiech. Application of empirical mode decomposition and artificial neural network for automatic bearing fault diagnosis based on vibration signals. *Appl. Acoust.*, 89:16–27, 2015.
- [112] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

- [113] Tin Kam Ho. Random decision forests. In *Proc. 3rd Int. Conf. Document Anal. and Recognition*, volume 1, pages 278–282. IEEE, 1995.
- [114] Corinna Cortes and Vladimir Vapnik. Support-vector networks. *Mach. Learn.*, 20(3):273–297, 1995.
- [115] Marina Sokolova and Guy Lapalme. A systematic analysis of performance measures for classification tasks. *Inf. Process. Manag.*, 45(4):427–437, 2009.
- [116] Laurens van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *J. Mach. Learn. Res.*, 9(Nov):2579–2605, 2008.
- [117] Wenliao Du, Jianfeng Tao, Yanming Li, and Chengliang Liu. Wavelet leaders multifractal features based fault diagnosis of rotating mechanism. *Mech. Syst. Signal Process*, 43(1-2):57–75, 2014.
- [118] Xiaoyuan Zhang, Yitao Liang, Jianzhong Zhou, et al. A novel bearing fault diagnosis model integrated permutation entropy, ensemble empirical mode decomposition and optimized svm. *Measurement*, 69:164–179, 2015.
- [119] Siqin Tao, Tao Zhang, Jun Yang, Xueqian Wang, and Weining Lu. Bearing fault diagnosis method based on stacked autoencoder and softmax regression. In *Control Conf. (CCC), 2015 34th Chinese*, pages 6331–6335. IEEE, 2015.
- [120] Rui Zhang, Yuan Lan, Guang-bin Huang, and Zong-Ben Xu. Universal approximation of extreme learning machine with adaptive growth of hidden nodes. *IEEE Transactions on Neural Networks and Learning Systems*, 23(2): 365–371, 2012.
- [121] G. Huang, Z. Bai, L. L. C. Kasun, and C. M. Vong. Local receptive fields based extreme learning machine. *IEEE Computational Intelligence Magazine*, 10(2):18–29, May 2015. ISSN 1556-603X. doi: 10.1109/MCI.2015.2405316.
- [122] Tianchi Liu, Chamara Kasun Liyanaarachchi Lekamalage, Guang-Bin Huang, and Zhiping Lin. Extreme learning machine for joint embedding and clustering. *Neurocomputing*, 277:78 – 88, 2018. ISSN 0925-2312. doi: 10.1016/j.neucom.2017.01.115. Hierarchical Extreme Learning Machines.
- [123] Dongshun Cui, Guang-Bin Huang, and Tianchi Liu. Elm based smile detection using distance vector. *Pattern Recognition*, 79:356–369, 2018.
- [124] Dongshun Cui, Guanghao Zhang, Wei Han, Liyanaarachchi Lekamalage Chamara Kasun, Kai Hu, and Guang-Bin Huang. Compact feature representation for image classification using elms. In *Computer Vision Workshop (ICCVW), 2017 IEEE International Conference on*, volume pp, pages 1015–1022. IEEE, 2017.
- [125] Lei Zhang, Xuehan Wang, Guang-Bin Huang, Tao Liu, and Xiaoheng Tan. Taste recognition in e-tongue using local discriminant preservation projection. *IEEE Transactions on Cybernetics*, pages 1–14, 2018.

- [126] Siqi Wang, En Zhu, Jianping Yin, and Fatih Porikli. Video anomaly detection and localization by local motion based joint video representation and ocelm. *Neurocomputing*, 277:161 – 175, 2018. ISSN 0925-2312. doi: 10.1016/j.neucom.2016.08.156. Hierarchical Extreme Learning Machines.
- [127] Hongwei Ge, Weiting Sun, Mingde Zhao, and Yao Yao. Stacked denoising extreme learning machine autoencoder based on graph embedding for feature representation. *IEEE Access*, 2019.
- [128] Dheeru Dua and Casey Graff. UCI machine learning repository. University of California, Irvine, School of Information and Computer Sciences, 2017. URL <http://archive.ics.uci.edu/ml>.
- [129] Sameer A. Nene, Shree K. Nayar, and Hiroshi Murase. Columbia object image library (coil-20. Technical report, 1996.
- [130] Jonathan J. Hull. A database for handwritten text recognition research. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 16(5):550–554, 1994.
- [131] Feiping Nie, Xiaoqian Wang, and Heng Huang. Clustering and projected clustering with adaptive neighbors. In *Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 977–986. ACM, 2014.
- [132] Y. Li, C. K. Liyanaarachchi Lekamalage, T. Liu, P. Chen, and G. Huang. Learning representations with local and global geometries preserved for machine fault diagnosis. *IEEE Trans. Ind. Electron.*, pages 1–1, 2019. ISSN 0278-0046. doi: 10.1109/TIE.2019.2905830.