Machine Learning and Audio Processing

A thesis presented in partial fulfilment of the requirements for the degree of

Doctor of Philosophy

in

Computer Science

at Massey University, Albany, Auckland,

New Zealand.

Junbo Ma

2019

Abstract

In this thesis, we addressed two important theoretical issues in deep neural networks and clustering, respectively. Also, we developed a new approach for polyphonic sound event detection, which is one of the most important applications in the audio processing area.

The developed three novel approaches are:

(i) The Large Margin Recurrent Neural Network (LMRNN), which improves the discriminative ability of original Recurrent Neural Networks by introducing a large margin term into the widely used cross-entropy loss function. The developed large margin term utilises the large margin discriminative principle as a heuristic term to navigate the convergence process during training, which fully exploits the information from data labels by considering both target category and competing categories.

(ii) The Robust Multi-View Continuous Subspace Clustering (RMVCSC) approach, which performs clustering on a common view-invariant subspace learned from all views. The clustering result and the common representation subspace are simultaneously optimised by a single continuous objective function. In the objective function, a robust estimator is used to automatically clip specious inter-cluster connections while maintaining convincing intra-cluster correspondences. Thus, the developed RMVCSC can untangle heavily mixed clusters without pre-setting the number of clusters.

(iii) The novel polyphonic sound event detection approach based on Relational Recurrent Neural Network (RRNN), which utilises the relational reasoning ability of RRNNs to untangle the overlapping sound events across audio recordings. Different from previous works, which mixed and packed all historical information into a single common hidden memory vector, the developed approach allows historical information to interact with each other across an audio recording, which is effective and efficient in untangling the overlapping sound events.

All three approaches are tested on widely used datasets and compared with recently published works. The experimental results have demonstrated the effectiveness and efficiency of the developed approaches.

Acknowledgements

I would like to take this opportunity to express my deepest gratitude to all the people who have supported me on my journey to achieve this qualification.

Firstly, I owe my most sincere gratitude to my supervisor, Prof. Ruili Wang, and co-supervisors, Prof. Xun Wang, Prof. Jianping Yin, Prof. En Zhu, Dr Andrew Gilman, Dr Reza Shahamiri and Dr Helen Zhou, for their invaluable academic guidance and spiritual support throughout my PhD research. They have spent dedicated time and efforts in helping me develop my research skills. It is really enjoyable when discussing research problems with them. They provide not only constructive but also challenging feedbacks to improve my research work. Their intellectual rigour and critical ways of thinking have deeply influenced my academic career. Without their valuable comments, suggestions, and persistent encouragements, it would be impossible for me to finish this PhD study.

I would like to thank all my colleagues at Prof. Ruili Wang's research group. We share ideas, advice and food together, and encourage each other through the stressful PhD study. Most importantly, without their help, I could not build a decent sentence-level Maori corpus and an End-to-end Maori speech recognition system.

I am also grateful to my previous supervisor during my Master study, Prof. Jianping Yin, who guided me into the computer science field. Because of his recommendation, I could meet Prof. Ruili Wang and get the opportunity to pursue

my PhD at Massey University, where I have broadened my horizon and extended my international academic experience.

Lastly, I would like to thank my parents for their unconditional love and caring. For my beloved mum, I am sorry I could not go back to China and be with her when she was in the surgery room.

Contents

Contents vii
List of Figures ix
List of Tables........ x
Chapter 1 Introduction 1
1.1 Introduction and Motivations 1
1.1.1 Machine Learning 1
1.1.2 Audio Processing 6
1.1.3 Summary 8
1.2 Research Objectives 9
1.3 Major Contributions 11
1.4 Organisation of the Thesis 13
Chapter 2 Large Margin Recurrent Neural Networks 15
2.1 Introduction 16
2.2 Backgrounds and Preliminaries 20
2.2.1 Recurrent Neural Networks 20
2.2.2 Long Short Term Memory Networks 21
2.2.3 Special Initialized Recurrent Neural Network (IRNN) 24
2.2.4 Preliminaries 25
2.3 Large Margin Recurrent Neural Networks 27
2.3.1 The drawback of Cross-Entropy Loss Function 27
2.3.2 Margin Term and Geometric Interpretation 31
2.3.3 Optimisation 34
2.4 Experiments 40
2.4.1 MNIST dataset 41
2.4.2 Penn Treebank Corpus 44
2.4.3 Discussion 49
2.5 Conclusions 51
Chapter 3 Robust Multi-View Continuous Subspace Clustering 53
3.1 Introduction 54
3.2 Related Work 57
3.2.1 Multi-view Subspace Clustering 57
3.2.2 Continuous Clustering 58

3.3 Model Formulation 59

3.4 Optimisation 63

3.5 Convergence Analysis 68

3.6 Experiments 70

3.6.1 Dataset Descriptions 71

3.6.2 Experimental Setup 72

3.6.3 Experimental Results 74

3.7 Conclusion 79

Chapter 4 Polyphonic Sound Event Detection 81

4.1 Introduction 82

4.2 Related Work 87

4.3 Models 91

4.3.1 Relational Recurrent Neural Networks 91

4.3.2 Proposed Approach 95

4.4 Evaluation 98

4.4.1 Datasets 98

4.4.2 Evaluation Metrics 100

4.4.3 Experiments 101

4.5 Conclusions 106

Chapter 5 Conclusions and Future Works 108

5.1 Research Overview and Summary 108

5.2 Future Works 111

References A-1

Appendix A List of Publications A-21

List of Figures

Figure 2.1 - Recurrent Neural Network (RNN) and its unrolling. 21

Figure 2.2 - A memory cell of Long Short-Term Memory (LSTM) network 23

Figure 2.3 - Example 1 for classification problem with 5 categories. 29

Figure 2.4 - Example 2 for classification problem with 5 categories 30

Figure 2.5 - Geometric interpretation of the proposed margin term. 34

Figure 2.6 - LMRNN accuracy with different margin parameters........................ 43

Figure 2.7 - Margin parameter and corresponding training and validation perplexities 47

Figure 2.8 - Values of cross-entropy term and margin term in each iteration during training 50

Figure 3.1 - The common representations produced by RMVCSC on Digit dataset 76

Figure 3.2 - Convergence behaviours of RMVCSC on three datasets. 77

Figure 4.1 - Sound events in a real-life scenario can occur in isolation or overlap. 84

Figure 4.2 - Relational Memory Core 91

Figure 4.3 - Relational recurrent neural networks for polyphonic sound event detection 96

List of Tables

Table 2.1 - Best results on MNIST dataset 42

Table 2.2 - Configurations of each model... 45

Table 2.3 - Best perplexities of each model.. 48

Table 3.1 - Details of the multi-view datasets.. 71

Table 3.2 - Clustering results of different approaches on the Cal-tech101-7 dataset... 78

Table 3.3 - Clustering results of different approaches on the WebKB dataset..... 78

Table 3.4 - Clustering results of different approaches on the Digit dataset.......... 79

Table 4.1 - The configurations of RRNN-SED for each dataset......................... 103

Table 4.2 - Experimental results on TUT-SED 2016 dataset 104

Table 4.3 - Experimental results on TUT-SED 2017 dataset 105

Chapter 1 Introduction

This chapter provides an overview of this thesis. An introduction to this thesis is presented in Section 1.1. Three research objectives are summarised in Section 1.2. The major contributions of this thesis are presented in Section 1.3. At the end of this chapter, the organisation of this thesis is presented in Section 1.4.

1.1 Introduction and Motivations

In this thesis, three novel approaches are developed in the machine learning and audio processing areas. Specifically, the first and second approaches address two crucial theoretical issues in deep neural networks and clustering, which are the two most popular subfields in the machine learning area in recent years. The third approach is developed for polyphonic sound event detection, which is one of the most important applications in the audio processing area. In this section, a brief introduction about machine learning and audio processing is presented, together with the motivations about each developed approach.

1.1.1 Machine Learning

Machine learning is one of the major research areas in Artificial Intelligence (AI) [3]. The aim of machine learning is to build mathematical models directly from data samples (known as training data) without explicit instructions [4], as some tasks can be easily solved by humans but are hard to explicitly explain how humans solve them. Machine learning fills this gap by letting computers automatically learn mapping models from the data samples, which can project data samples to their desired output (known as data labels) [5].

The concept of machine learning was firstly proposed by Arthur Samuel in 1959 [6]. Back then, the research in machine learning was mainly simple statistical models in the computer gaming field [7]. After several decades' development, a more formal definition of machine learning was proposed by Tom M. Mitchell in 1997, which defines machine learning from the perspective of automatically improving the performance of mathematical models from experiences [8]. Many machine learning techniques have been developed in history. Some worth mentioning milestones are backpropagation of neural networks developed in the 1970s [9], support vector machine developed in 1990s [10], and deep learning developed in 2000s [11].

Machine learning can be categorised into three main paradigms [4]: supervised learning, unsupervised learning, and reinforcement learning. In supervised learning, each data sample is consistently paired with a given label. A mapping model takes the data samples as input and learns to output predictions as close as to their corresponding labels, which means the learning process is "supervised" by the ground truth labels [12]. On the opposite, data samples do not have any labels in unsupervised learning. The mapping model tries to unveil the underlying commonalities in the data samples [13]. Reinforcement learning sits in between supervised learning and unsupervised learning. In reinforcement learning, the data samples do not have direct labels. Instead, a measurement for each action is given, known as a reward. The mapping model aims to find an optimal action sequence to maximise the cumulative reward [14].

Besides these three main machine learning paradigms mentioned above, several sub-paradigms blend across them. For example, semi-supervised learning paradigm is considered as a sub-paradigm blending supervised and unsupervised learning paradigms. It leverages unlabelled data samples in the unsupervised context to augment the supervised learning. This is because the unlabelled data samples are relatively much easier to obtain than labelled data samples. Active learning is a special type of the semi-supervised learning, which will be further discussed in Section 5.2, future works.

This thesis will focus on the first two main paradigms, supervised learning and unsupervised learning.

1.1.1.1 Supervised Learning

Supervised learning is the most widely used machine learning paradigm [4]. Many real-world applications rely on supervised learning paradigm, such as sound event detection [2][15][16][17], speech recognition [18][19][20], action recognition [21][22][23], and machine translation [24][25][26]. Accordingly, various supervised learning approaches have been developed, such as deep learning [27], decision tree [28], nearest neighbour [29] and support vector machine [30]. In the past few years, due to the increasing data volume and decreasing hardware cost, deep learning has shown great advances and becomes the dominating approach in many applications [31].

Deep learning is a subfield of machine learning, which utilises multiple processing layers to compose computational models [31]. Depending on the design of processing layers, many deep learning architectures have been proposed, such as Deep Belief Networks [73][93], Deep Auto-Encoder [74][98], Deep Stack Networks [76], Convolutional Neural Networks (CNNs) [75][94][97], and Recurrent Neural Networks [77][78][79][80].

Among all these deep learning architectures, Recurrent Neural Network (RNN) has been proved that it is effective in processing sequential data and identifying long-term dependencies [11][31]. This is because RNN has a specially designed recurrent operation inside its hidden layers, which makes them capable of passing historical information from previous inputs and hidden layers to their successors [31]. However, in multi-class classification tasks, most of the current RNNs employ the cross-entropy loss function [27][31], which does not fully benefit from the information provided by data labels. This is because the cross-entropy loss function only considers the target category without considering the competing categories during training processes [81]. To solve this problem, a Large Margin Recurrent Neural Network (LMRNN) is developed in Chapter 2, which employs a large margin discriminative principle as a heuristic term to navigate the convergence process during training.

1.1.1.2 Unsupervised Learning

Different from supervised learning, unsupervised learning is processed without guidance from data labels. In real-world applications, data labels are not always available or easy to acquire. In some cases, the underly structures of data samples are required to be analysed [4]. Many applications rely on unsupervised learning, such as clustering [32][33][34], dimensionality reduction [35][36][37], feature extraction [38][39][40][41], and representation learning [42][43][44]. Accordingly, various unsupervised learning approaches have been developed, such as *k*-means [45], Gaussian mixture model [46][47], Auto-encoder [48][49] and Generative Adversarial Networks (GANs) [50][51][52].

Clustering is one of the basic approaches for statistical data analysis, which aims to group data samples into subsets according to some defined measures [13][45][110]. Many clustering approaches have been developed, such as *k*-means [53], *k*-medoids [54], hierarchical clustering [55], and density-based clustering [56].

In recent years, multi-view clustering has attracted arising attentions, as data samples are often in the form of multiple views in real-world applications [33][111][112]. A view is a distinctive perspective of the data samples. Data from different domains or features extracted by different feature extractors are considered as different views of the data samples [113][114]. Multi-view clustering aims to leverage complementary information among multiple views to improve the clustering accuracy and generalisation ability [116]. One primary approach for multi-view clustering is the subspace approach, which aims to learn a common latent subspace from all views and perform clustering in this subspace [116][117]. However, most existing multi-view subspace clustering approaches are based on *k*-means or spectral clustering, in which the number of clusters *k* and the weights of different views are required to be pre-set manually [113][119]. This may limit the further advancement of multi-view subspace clustering. To solve this problem, a novel Robust Multi-View Continuous Subspace Clustering (RMVCSC) approach is developed in Chapter 4, which exploits the advantage of recent published Robust Continuous Clustering (RCC) with the multi-view clustering setting.

1.1.2 Audio Processing

Many real-world applications rely on machine learning. One of the important applications of machine learning is audio processing [4]. Audio processing aims to extract meaningful information (descriptions or explanations) from audio [57], such as the type of a sound event [58], the content of a speech [59] or the artist of music [60].

From a historical perspective, audio processing started growing rapidly after the introduction of the Compact Disc (CD) in 1982 [61]. Since then, high-quality audio recordings can be easily accessed. In the past few decades, with the fast development of the Internet, the analysis demands of audio recordings significantly increased [66], which resulted in a lively audio processing research area.

Audio processing covers a vast of diverse research fields, such as sound event detection [63], speech processing [64] and music information retrieval [62], as audio signals may come from various sound sources. Each of these research fields also has its subfields to focus. For example, speech processing can be roughly categorised into speech recognition and speaker recognition [65]. Speech recognition focuses on recognising the content of a speech, whereas speaker recognition focuses on recognising the speaker of a speech and does not care much about the content [64].

One important practical application in audio processing is sound event detection [66]. Sound event detection aims to identify sound events from sound signals [176], which provides essential information about the context in the environment [156][157]. Thus, sound event detection has been widely applied into many context-aware tasks, such as acoustic surveillance [220][221], healthcare systems [222][223] and remote wildlife monitoring [224][225]. sound event detection can be broadly classified to monophonic sound event detection and polyphonic sound event detection [176][177]. Monophonic sound event detection is to recognise the most dominant sound events in a sound recording, whereas polyphonic sound event detection recognises all sound events (not only the most dominant sound event) in a sound recording [170][175][176]. In realistic scenarios, multiple sound events are very likely to overlap with each other in time. Thus, polyphonic sound event detection is more useful and challenging.

A large number of deep learning based approaches have been developed in recent years, which are considered as the cutting-edge approaches for polyphonic sound event detection, such as Convolutional Neural Network (CNN) based approaches [177][178][179] and RNN based approaches [180][181][182][183]. However, a major drawback in these approaches is that all the historical information is mixed together and packed into one single hidden memory vector, which will limit the ability of these approaches to untangle overlapping sound events. To solve this problem, a novel Relational Recurrent Neural Network (RRNN) based approach for polyphonic sound event detection is developed in Chapter 4, called RRNN-SED, which exploits the strength of Relational Recurrent Neural Network in long-term temporal context extraction and relational reasoning across a polyphonic sound signal.

1.1.3 Summary

To sum up, machine learning is a booming research area, which covers a vast of subfields and applications. This thesis addresses two crucial theoretical issues in the two most popular subfields (deep learning and clustering). Also, a novel approach is developed for polyphonic sound event detection, which is one of the most widely used applications in the audio processing area. Three developed novel approaches are: (i) the Large Margin Recurrent Neural Network in Chapter

2; (ii) the Robust Multi-View Continuous Subspace Clustering approach in Chapter 3; (iii) the Relational Recurrent Neural Network (RRNN) based polyphonic sound event detection approach in Chapter 4.

1.2 Research Objectives

In the previous section, the motivations for the developed approaches are discussed. In this section, three research objectives are presented correspondingly.

Objective 1 involves developing a novel approach to improve the discriminative ability of original Recurrent Neural Networks (RNNs).

As discussed in Section 1.1.1.1, most of RNNs utilise the cross-entropy loss function in multi-class classification tasks. However, the information provided by data labels is not fully used, as the cross-entropy loss function only considers target category without considering competing categories during training processes. To solve this problem, a novel Large Margin Recurrent Neural Network is proposed, which improves the discriminative ability of original Recurrent Neural Networks by introducing a large margin term into the widely used cross-entropy loss function.

Objective 2 involves developing a novel approach for multi-view subspace clustering, which aims to utilise comprehensive information from multiple views without manually pre-setting the number of clusters and the weight of each view.

As discussed in Section 1.1.1.2, most existing multi-view subspace clustering approaches are based on *k*-means or spectral clustering, in which the number of clusters *k* and the weights of different views are required to be pre-set manually. This may limit the further advancement of multi-view subspace clustering. To solve this problem, a Robust Multi-View Continuous Subspace Clustering (RMVCSC) approach is proposed, which utilises a robust estimator to automatically clip specious inter-cluster connections while maintaining convincing intra-cluster correspondences.

Objective 3 involves developing a novel approach for polyphonic sound event detection, which aims to utilise the context information to untangle overlapping sound events.

As discussed in Section 1.1.2, most of existing polyphonic sound event detection approaches rely heavily on CNNs or RNNs, in which all historical information is mixed together and packed into a single hidden memory vector. This may limit the ability to untangle overlapping sound events. To solve this problem, a novel Relational Recurrent Neural Network (RRNN) based polyphonic sound event detection approach is proposed, which utilises the relational reasoning ability of RRNNs to untangle the overlapping sound events across audio recordings.

1.3 Major Contributions

Corresponding to the three research objectives proposed in the previous section, the contributions of this thesis are summarized as the following:

(i) Developed a Large Margin Recurrent Neural Network (LMRNN), which utilises the large margin criteria to improve the discriminative ability of the original RNN.

The proposed LMRNN improves the discriminative ability of the original RNNs while maintaining the capability of handling sequential data. The proposed large margin term is mathematically analysed and tested with two typical tasks. The experimental results demonstrate that the proposed LMRNN outperforms current RNNs, such as the specially Initialized Recurrent Neural Networks (IRNNs) and the bi-directional Long Short Term Memory networks in terms of accuracy and perplexity without increasing the depth. The details about the proposed LMRNN model can be found in Chapter 2

(ii) Developed a Robust Multi-View Continuous Subspace Clustering (RMVCSC) approach, which can untangle heavily mixed clusters by optimising a single continuous objective function.

The proposed objective function of RMVCSC uses robust estimators to automatically clip specious inter-cluster connections while maintaining convincing intra-cluster correspondences in the common representation subspace learned from multiple views. RMVCSC is optimised in an alternating minimisation scheme, in which the clustering results and the common representation subspace are simultaneously optimised. Since different views can describe distinct perspectives of multi-view data, RMVCSC can achieve more accurate clustering performance than conventional approaches by exploring information among multi-view data. In other words, RMVCSC optimises a novel continuous objective function in the common representation subspace that is simultaneously learned across multiple views. By using robust redescending estimators, RMVCSC is not prone to stick into bad local minima even with outliers in data. This kind of robust continuous clustering approach has never been used for multi-view clustering before. Moreover, the convergence of the proposed approach is theoretically proved, and the experimental results show that the proposed RMVCSC can outperform several very recently proposed approaches in terms of clustering accuracy. The details about the proposed RMVCSC can be found in Chapter 3.

(iii) Developed a novel approach for polyphonic sound event detection, which utilises Relational Recurrent Neural Network (RRNN) for polyphonic sound event detection, named RRNN-SED.

The proposed RRNN-SED exploits the strength of RRNN in long-term temporal context extraction and relational reasoning across a polyphonic sound signal. Different from previous sound event detection approaches, which rely heavily on CNNs or RNNs, the proposed RRNN-SED approach can solve long-lasting and overlapping problems in polyphonic sound event detection. Specifically, since the historical information memorised inside RRNNs is capable of interacting with each other across a polyphonic sound signal, the proposed RRNN-SED approach is effective and efficient in extracting temporal context information and reasoning the unique relational characteristic of the target sound events. Experimental results on two public datasets show that the proposed approach achieved better sound event detection results in terms of segment-based *F*-score and segment-based error rate. The details about the proposed RRNN-SED can be found in Chapter 4.

To sum up, three novel approaches are developed to correspondingly fulfil the three objectives proposed in Section 1.2. These three novel approaches are: (i) the Large Margin Recurrent Neural Networks in Chapter 2; (ii) the Robust Multi-View Continuous Subspace Clustering approach in Chapter 3; (iii) the Relational Recurrent Neural Network (RRNN) based polyphonic sound event detection approach in Chapter 4. All these three developed novel approaches have already been submitted to or published on top journals.

1.4 Organisation of the Thesis

The rest of this thesis is organised as follows.

Literature reviews of the most relevant fields are presented in each chapter corresponding to the developed three approaches.

Chapter 2 presents the Large Margin Recurrent Neural Network. Chapter 3 presents the Robust Multi-View Continuous Subspace Clustering approach. Chapter 4 presents the Relational Recurrent Neural Network (RRNN) based polyphonic sound event detection approach. Chapter 5 concludes this thesis and discusses future works.

Chapter 2 Large Margin Recurrent Neural Networks

This chapter presents the first developed approach, named Large Margin Recurrent Neural Network (LMRNN), which fulfils the first research objective.

Recurrent neural networks (RNNs) have proved to be one of the most successful deep neural network architectures for processing sequential data. However, in multi-class classification tasks, most of the current RNNs employ the cross-entropy loss function, which does not fully benefit from the information provided by data labels, because it only considers target category without considering competing categories during training processes. To solve this problem, a Large Margin Recurrent Neural Network (LMRNN) is proposed in this chapter, which employs a large margin discriminative principle as a heuristic term to navigate the convergence process during training. The proposed LMRNN has improved the discriminative ability of the original RNN while maintaining the capability when generating sequential data. The proposed large margin term has been tested on both the original RNN and on the Long Short-Term Memory network with two typical tasks. The experimental results demonstrate that the proposed LMRNN outperforms current RNNs, such as the specially Initialized Recurrent Neural Network (IRNN) and the bi-directional Long Short Term Memory network in terms of accuracy and perplexity without increasing the depth.

This chapter is organised as follows. Section 2.1 introduces the deep learning research area and presents the motivation of the proposed LMRNN. Section 2.2 reviews the most relevant works and presents the preliminaries. Section 2.3 introduces the proposed LMRNN in details, together with the geometric interpretation and mathematic derivation. Section 2.4 presents the experiments and discusses experimental results. At the end of this chapter, conclusions are presented in Section 2.5.

2.1 Introduction

Deep learning has shown significant capabilities in many machine learning tasks in recent years [31], including image processing [67][68][92][96], video processing [99][100], speech recognition [69], natural language processing [70], information retrieval [71] and healthcare [72]. Various architectures of Deep Neural Networks (DNNs) have been proposed in the last few years such as Deep Belief Network [73][93], Deep Auto-Encoder [74][98], Convolutional Neural Networks (CNNs) [75][94][97], and Deep Stack Networks [76]. However, most of these architectures have difficulties in dealing with sequential data and identifying long-term dependencies embedded in sequences, mainly because the input and output sequences normally have to be subdivided by a fixed size of window in these architectures, and the input sub-sequences are assumed to be independent and identically distributed from each other [27].

Among the prominent DNN architectures, Recurrent Neural Networks (RNNs) have shown its effectiveness in processing sequential data and identifying long-term dependencies [27]. This is because RNNs have a specially designed recurrent operation inside hidden layers, which makes them capable of passing historical information from previous inputs and hidden layers to their successors [31]. Due to this characteristic, RNNs have been widely applied in many scenarios where input data are naturally in the form of sequences such as speech recognition [77], machine translation [78], language modelling [79] and video processing [80][92][95][99].

Similar to other deep learning architectures, the cross-entropy loss function with a softmax layer has been widely used by RNNs in multi-class classification tasks [27]. A softmax layer is a fully connected layer employed as the last layer in DNNs to squash the output of its previous layers into probabilities of each category. Then, the cross-entropy loss function is used to ensure that the output probability of the target category is as large as possible. The combination of the softmax layer and the cross-entropy loss function helps DNNs to maintain their generative ability in characterising the joint distributions of the inputs and their relevant outputs [27].

However, the cross-entropy loss function does not fully utilise the information provided by data labels as it only considers the target category without taking the competing categories into account during training processes [81], whereas the ignored potentially useful information provided by the labels of competing categories is crucial. This is especially significant when there are a large number of classification categories. For example, the target categories can be ten thousand or more in language modelling tasks, depending on the size of the vocabulary, in which proper use of all discriminative information may significantly increase the performance of the language models.

In order to address this problem, a large margin penalty term was introduced into the cross-entropy loss function recently [82]. In this work, the authors first derived an upper bound of Rademacher Average for DNNs in the view of the margin bound and then investigated the influence of the depth to the empirical margin error in DNNs. From their theoretical derivation, they concluded that a large margin penalty term can be added into the cross-entropy loss function to reduce the empirical margin error of a DNN without increasing the depth of it [82]. Although this work provides a solid theoretical foundation to the combination of the penalty term and the cross-entropy loss function, the margin terms they proposed do not have a clear geometric intuition. Furthermore, their work focused exclusively on CNNs, whereas our work will focus on RNNs.

Another work that utilised the large margin criteria was proposed in [106], which introduced a cosine similarity into the softmax layer and proposed a large-margin softmax loss function for CNNs. Different from this work, our work leaves the softmax layer untouched and focuses on the cross-entropy loss function and RNNs. Additionally, another work published recently [107] derived the generalisation error bounds of DNNs based on the Jacobian matrix in terms of the classification margin and proposed a Jacobian regularizer based on their derivation. However, their work did not address the problem of the cross-entropy loss function.

In this chapter, a constrained binary-optimisation sub-problem is introduced into the cross-entropy loss function to optimise both the target category and the competing categories jointly. Then, a Large Margin Recurrent Neural Network (LMRNN) is proposed by employing the large margin discriminative principle as a heuristic term to navigate the convergence process in the training procedure to enhance the discriminative ability of the original RNNs, while maintaining their capability when dealing with sequential data such as generating a sentence from a given context vector. The proposed model has a very clear geometric intuition, and experimental results show that the proposed margin term works well on both original RNN and Long Short-Term Memory (LSTM) network.

The rest of this chapter is organised as follows: Section 2.2 introduces the research background and presents the preliminaries; Section 2.3 introduces the proposed LMRNN and discusses the geometric interpretation and mathematic derivation; Section 2.4 presents and discusses the experimental results; finally, Section 2.5 presents the conclusions.

2.2 Backgrounds and Preliminaries

In this section, we will first review the original Recurrent Neural Networks (RNNs) and provide an overview of its architecture. Then, we introduce two RNN variations considered in our experiments. They are LSTM networks [83][102] and the special Initialized Recurrent Neural Networks (IRNNs) [84]. LSTM is one of the most widely used RNN architectures, whereas IRNN is designed to achieve comparable performance but consume much less computational resources than a complex LSTM, despite having similar architecture to the original RNN. At the end of this section, we will set up the preliminaries of this chapter by introducing the softmax layer combining with the cross-entropy loss function.

2.2.1 Recurrent Neural Networks

Recurrent Neural Networks (RNNs) are known by a specially designed recurrent operation inside hidden layers, which can pass historical information from previous inputs and hidden layers to their successors [31]. This operation gives RNNs the ability to memorise the historical information inside hidden layers and identify long-term dependencies embedded in the sequence [27].

Figure 2.1 illustrates a typical RNN and its unrolling, in which $x = \{x_0, \ldots, x_t, \ldots\}$ is the input sequence, h is the output hypothesis of the relevant input, A denotes the hidden cell, s is the hidden state and U, V, W are the weight matrices between layers. At time step t, the RNN will take the input x_t and the state of previous hidden layer *s_{t-1}* to compute the current hypothesis h_t and the current hidden layer's state s_t with the following equations, where ϕ_s and ϕ_h are arbitrary activation functions:

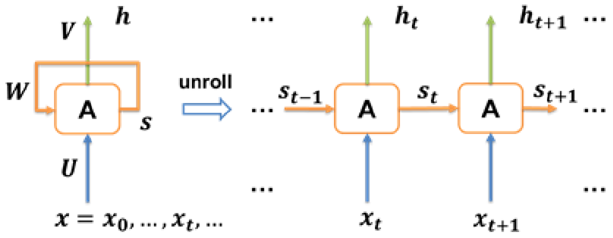


Figure 2.1 - Recurrent Neural Network (RNN) and its unrolling.

$$s_t = \phi_s(Ux_t + Ws_{t-1} + b) \tag{2.1}$$

$$h_t = \phi_h(Vs_t + c) \tag{2.2}$$

Equations (2.1) and (2.2) show that the historical information is passed throughout the whole sequence and the output of every time step is affected by the historical information, which means RNNs are capable of making use of the historical information embedded in the sequences. Another advantage of RNNs is that the size of the input and output vectors are not fixed, which means that the input sequence x and the output sequence h can have arbitrary lengths. This flexibility makes RNNs a natural choice for handling sequential data with variable lengths.

2.2.2 Long Short Term Memory Networks

LSTM network is a variant of RNNs proposed by Hochreiter & Schmidhuber in 1997 [83]. LSTM augments the classic RNN with a specially designed hidden unit called the recurrent gate. The recurrent gate controls the scale of information to be remembered or forgotten by the networks and helps LSTM to overcome the vanishing or exploding gradient problem, which prevents the classic RNNs from learning very long-term dependencies in practice [85]. Subsequently, LSTM networks have been proved to be more effective than the classic RNNs, especially in the tasks that require very deep structures [78][100][101].

Figure 2.2 illustrates a typical LSTM structure at timestamp *t*. The red circles *f*, *i* and *o* denote the forget-gate, the input-gate and the output-gate respectively, and they are calculated with Equations (2.3), (2.4) and (2.5). The symbols W_* and R_* are the relevant weight matrices between layers. The symbol σ denotes the sigmoid function, which maps its input value into the interval *(0, 1)*. Thus, these gates can be described as the proportion of information that can get through them. In other words, these gates can control what to be remembered or forgotten inside the sequence. The symbol *c* stands for the candidate value, which will renew the hidden states with current input x_t, and it is calculated by Equation (2.6). The equations for the current hidden state s_t and the current output hypothesis h_t are (2.7) and (2.8), respectively.

In contrast to classical RNNs, LSTM replaces the hidden layer with a memory cell that can store and process historical values throughout the input sequence. The memory cell has an input gate, an output gate and a forget gate. The input gate controls how much input can affect the stored memory value; the output gate controls how much this stored memory value is allowed to affect the output; the forget gate controls how much the stored memory value fades in each time step. All these three gates have a value in the range of *(0, 1)* and they are connected with the current input and the previous memory cell output. Each connection has its own weights that can be learnt during the training process.

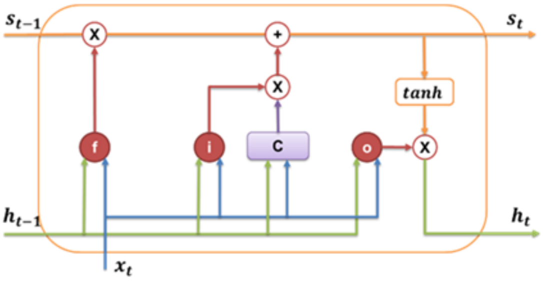


Figure 2.2 - A memory cell of Long Short-Term Memory (LSTM) network

$$f_t = \sigma\left(W_f x_t + R_f h_{t-1} + b_f\right) \tag{2.3}$$

$$i_t = \sigma(W_i x_t + R_i h_{t-1} + b_i) \tag{2.4}$$

$$o_t = \sigma(W_o x_t + R_o h_{t-1} + b_o) \tag{2.5}$$

$$c_t = tanh(W_c x_t + R_c h_{t-1} + b_c) \tag{2.6}$$

$$s_t = f_t s_{t-1} + i_t c_t \tag{2.7}$$

$$h_t = o_t * tanh(s_t) \tag{2.8}$$

This structure gives LSTM networks a significant property: the gradient error can be back-propagated all the way through the sequence [80]. In a special case, when the input gate and the output gate are turned off, and the forget gate does not fade any memory, which means the value of input gate and output gate are set to zero while the value of forget gate is set to one, every memory cell constantly maintains the same memory state throughout the sequence. In the meantime, the gradient error of each memory cell also remains unchanged during the back-propagating operation. Thus, LSTM networks have the ability to propagate the error backwards throughout the whole sequence and overcome the vanishing or exploding gradient problem [85].

2.2.3 Special Initialized Recurrent Neural Network (IRNN)

Proposed by Quoc V. Le [84], IRNN has the same structure as the original RNNs; however, the weight matrix of the recurrent layer is initialised to an identity matrix with all zero biases, and the tanh activation function is replaced by a Rectified Linear Unit (ReLU). This means that the hidden unit will just copy the value of the hidden state from the previous time step, and then add the effects of the current input to produce the current hidden state. The ReLU units will then preserve all the positive values of the current hidden state and replace all the negative values with 0. In a special situation, when the input effects are ignored, all the hidden states throughout the sequence will stay the same indefinitely.

Due to this special initialisation, IRNN has a similar expected property that gradient error can be back-propagated all the way through the sequence. Thus, IRNN has the ability to overcome the vanishing or exploding gradient problem and learn extremely long-term dependencies while achieving comparable performance with a complex LSTM network but remains less resource consuming, as demonstrated in [84].

2.2.4 Preliminaries

In this section, we will set up the preliminaries of this chapter by introducing the softmax layer combining with the cross-entropy loss function.

In a typical multi-class classification task, the input space is usually defined as $X = \mathbb{R}^d$, where d stands for the dimension of the input space. The output space is then defined as $Y = \{1, \ldots, K\}$, where K stands for the total number of categories. The joint distribution over X, Y is defined as $P(X, Y)$, and the training set sampled from the distribution $P(X, Y)$ is denoted as $S = \{(x^{(1)}, y^{(1)}), \ldots, (x^{(n)}, y^{(n)})\}$, where $x \in X$ and $y \in Y$, the superscript denotes the number of the sample. The goal of this classification task is to learn a prediction model F from the training set S, which takes each $x \in S$ as input and predicts the probabilities of the input x belonging to each category k, where $k \in Y$. If $P(k \mid x)$ is defined as the probability which input x belongs to category k, then $F(x, k) = P(k \mid x)$. The final prediction decision is made by $\text{argmax}_k P(k \mid x)$, where $k \in Y$, which means we choose the category in which the maximum probability belongs to as the final decision of the prediction model F.

In a deep neural network based prediction task, a softmax output layer can be used to properly present the outputs as probability distributions. In particular, a softmax layer is a fully connected layer, in which the number of nodes is the same as the number of categories *K*. Each node will first produce a value that maps the outputs from previous layers into each category. Then, these values will be normalised by the softmax function to produce the final outputs of the softmax layer. More specifically, if z_k is defined as the intermediate output of the softmax layer belonging to category *k*, then, the softmax function is applied to each z_k to produce the output probabilities of each category. The softmax function is defined as:

$$soft\,max_k(x) = P(k|x) = \frac{exp(\,z_k)}{\sum_{j\in Y} exp(\,z_j)}, \tag{2.9}$$

where $softmax_k(x)$ denotes the output of the softmax layer for category *k* taking sample *x* as input. From the definition of $softmax_k(x)$ and $P(k|x)$, it can be comprehended that $softmax_k(x) = P(k|x)$.

The output of the softmax layer is a vector, in which each element illustrates the probability of given input belonging to each corresponding category. The purpose of the training process is to make sure that the output probability of the target category illustrated by the sample label *y* is the largest one in this vector. This can be achieved by putting the vector into a loss function to evaluate it. In terms of DNNs, the most commonly used loss function is the cross-entropy loss function, which is:

$$CE(x, y) = -\sum_{k=1}^{K} 1_k \, ln\, P\,(k|x) \tag{2.10}$$

where $1_k = 1$ if $k = y$, and $1_k = 0$, otherwise.

Equation (2.10) depicts that minimising the loss function leads to increasing the output probability value corresponding to the correct category as expected. Nonetheless, since $1_k = 0$ when $k \neq y$, the sign function 1_k enforces the cross-entropy loss function only taking the output probability corresponding to the target category into account and ignoring the remaining probabilities of all other competing categories. It can be argued that the cross-entropy loss function implicitly uses the output probabilities of competing categories as the sum of all the outputs constantly equals to 1, when the output probability corresponding to the target category increases, all other outputs decrease accordingly. However, this is not exactly true, and we will discuss this in-depth in the next section.

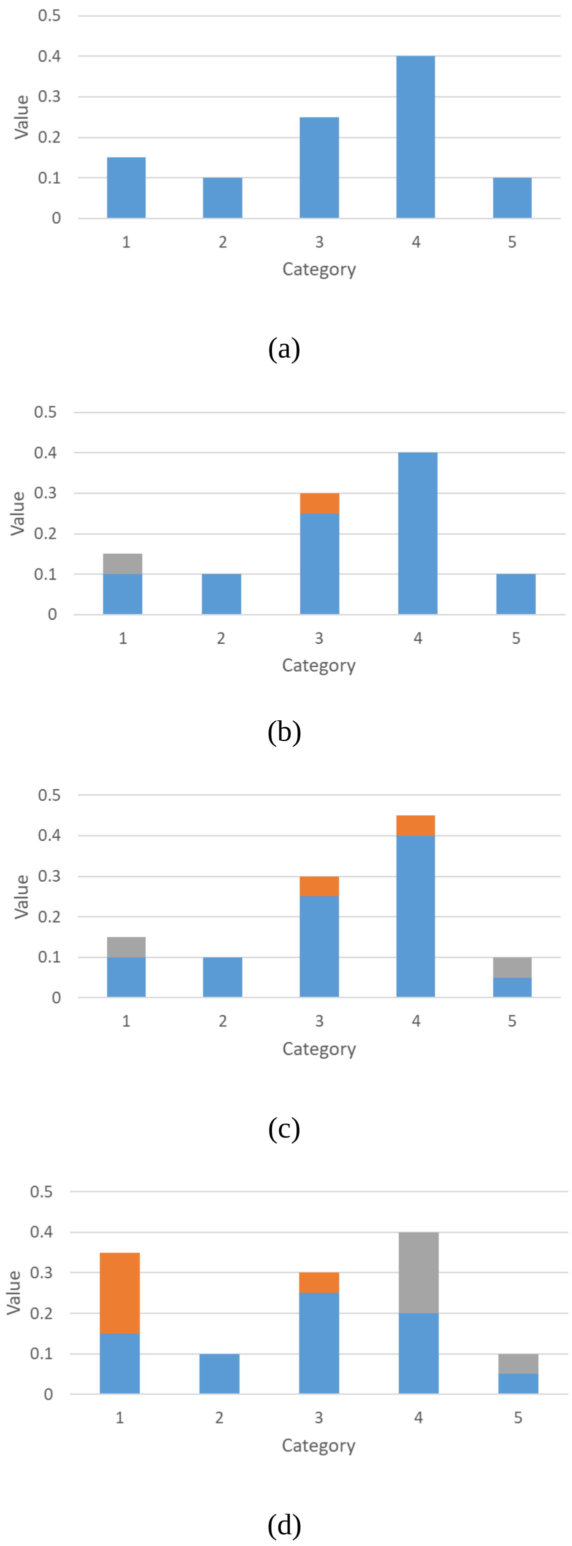
2.3 Large Margin Recurrent Neural Networks

This section further explains the drawback of the cross-entropy loss function and then discusses the proposed margin term and its geometric interpretation. At the end of this section, the mathematic derivation for optimisation is also presented.

2.3.1 The drawback of Cross-Entropy Loss Function

As discussed in the previous section, the cross-entropy loss function does not fully utilise the information provided by the competing categories since it only takes the output probability corresponding to the target category into account and ignores all other probabilities from competing categories. Although the sum of all the output probabilities from competing categories decreases when the output probability of target category increases, it does not necessarily mean that the probabilities from competing categories go down equally. This may decrease the performance of DNNs when increasing the probability of the target category does not lead to modification, or even increment of the maximum output probability values of competing categories, which may result in the prediction model not being able to make a correct prediction. This problem can happen in all multi-class classification problems that have more than two categories.

For example, Figure 2.3 illustrates a classification problem with five categories. Each bar illustrates the output probability produced by a prediction model for each category. Suppose the target category of the current input sample is 3. Hence the other categories are competing categories. In Figure 2.3(a), the output probability for the target category is 0.3, whereas the maximum probability from competing categories is 0.4, which means the prediction model does not produce an accurate prediction. Applying the cross-entropy loss function to enlarge the output probability for the target category could result in Figure 2.3(b): the output probability for target category increases by 0.05 to 0.35, whereas only the first category decreases by 0.05 and the maximum output probability of all categories is still category 4, which means the prediction model still cannot produce a correct answer.



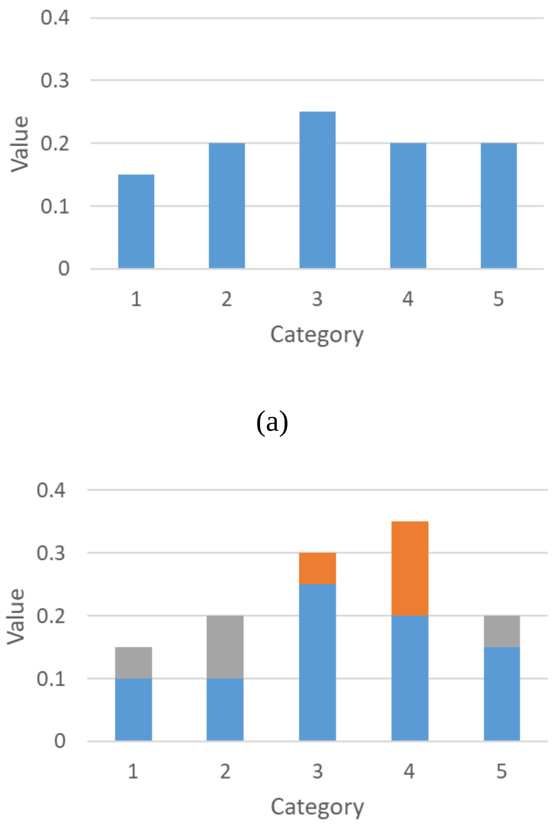
(a)

(b)

(c)

(d)

Figure 2.3 - Example 1 for classification problem with 5 categories.



(a)

(b)

Figure 2.4 - Example 2 for classification problem with 5 categories

The worst-case scenario is that the predication model successfully makes the classification and produces the correct answer initially; however, after the next step of training with the cross-entropy loss function, one of the competing output probabilities suddenly jumps to be the top one due to the changes to the weights of DNNs, and becomes the new maximum output probability of all categories despite increasing the output probability of the target category simultaneously. This will make the DNN produce a wrong answer regardless of the correct answer produced previously. Figure 2.4 illustrates this scenario: during the training process, the output probability of target category 3 was the maximum one of all categories, which means the DNN produced a correct answer, as illustrated in Figure 2.4(a). However, as shown in Figure 2.4(b), after training another epoch, the cross-entropy loss function increases the output probability of category 3 as expected, but the output probability of category 4 increases simultaneously and becomes the maximum one of all categories, which means the DNN will produce a wrong prediction instead of the previous correct one.

Such problems are caused by ignoring the output probabilities of competing categories during the training process. In order to solve this problem, we have developed a mechanism that utilises the information provided by the competing categories to constrain the cross-entropy loss function to converge towards the expected direction.

2.3.2 Margin Term and Geometric Interpretation

From the discussion of the previous section, we can see that the output probability of the target category and the maximum output probability of all competing categories are the two main factors that affect the prediction model producing a correct answer. The output probability of the target category is expected to be as large as possible, while the maximum output probability of all the competing categories is expected to be as small as possible. This will maximise the distance between correct and incorrect predictions, in order that the prediction model can easily distinguish the correct category from any other competing category.

The problem is that the maximum output probability of all the competing categories may vary from one to another. Nevertheless, they are all competing categories, which enables us to introduce a binary optimisation sub-problem here.

In this binary optimisation sub-problem, all the output probabilities produced by softmax layer and evaluated by the cross-entropy loss function are treated as samples of this sub-problem. The output probability of the target category is considered as a positive sample, while all other output probabilities from competing categories are considered as negative samples. The purpose of this problem is to distinguish these two sample groups. In order to do so, a margin term can be defined as:

$$\begin{gathered} M(x,y) = P(y|x) - \max_{k \in Y, k \neq y} P(k|x) \\ s.t. P(y|x), P(k|x) \in (0,1), \sum_{k \in Y} P(k|x) = 1 \end{gathered} \tag{2.11}$$

where $P(y|x)$ is the output probability of target category y provided by the label of the input sample, $P(k|x)$ is the output probability of category k with input training sample x, as defined in the preliminaries section.

Then, the new loss function is defined as:

$$L(x,y) = CE(x,y) - \lambda M(x,y) \tag{2.12}$$

where λ is the margin parameter to balance these two terms.

It is clear that the sample space of this sub-problem has only one dimension, and the discriminative relation between these two sample groups is straightforward. Thus, a complex margin term may not be necessary. Additionally, in this binary classification sub-problem, the relationship between the samples is not independent, because they are all probabilities of corresponding categories, and the sum of them constantly equals to 1. If one of them increases its value, some of the others must decrease. Hence, if we take all the output probabilities into account, there must be some redundancy information in the margin term, which may decrease the performance. Furthermore, according to a recent review of the margin-based approaches [87], a simple margin term is usually adequate to achieve competing performance to a complex one.

When minimising the loss function in Equation (2.12), the cross-entropy term enlarges the output probabilities corresponding to the sample labels to achieve high classification accuracy, while the margin term magnifies the gaps between the correct predictions and the maximum incorrect predictions in order to make the correct predictions more distinguishable. λ in Equation (2.12) is a weight parameter to balance these two terms.

The geometric interpretation of this margin term is simple and clear. Figure 2.5 illustrates how the margin term works during training. The margin term faces two main situations during training. Figure 2.5(a) is the ideal situation in which the output probability of the target category is the maximum one of all the categories.

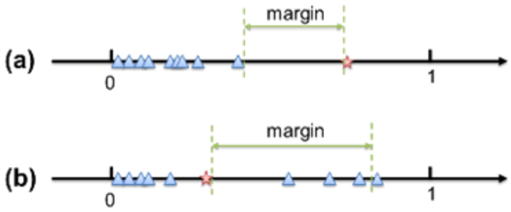


Figure 2.5 - Geometric interpretation of the proposed margin term.

Thus, the value of the margin term is positive, and when Equation (2.12) is minimised, the optimiser enlarges the margin between these two classes, which makes the target category more distinguishable from the competing categories. Figure 2.5(b) shows the situation we are trying to optimise. In this situation, the output probability of the target category is smaller than some output probabilities of the competing categories. Thus, the margin value is negative when Equation (2.12) is minimised so that the optimiser reduces the margin between them. This guides the training process towards the expected direction, as depicted in Figure 2.5(a).

In summary, the margin term here can be considered as additional navigation or constraint for the loss function. Since the output probabilities generated by DNNs are left untouched, this loss function properly maintains their generative ability

2.3.3 Optimisation

From the geometric interpretation, we can intuitively see how the proposed margin term works in the training process. In this section, the mathematical derivation of the margin term is provided to explain how backpropagation algorithm with gradient descent benefits from it.

Since the original network structure of the prediction model is untouched, we will only focus on the gradient descent aspect of the the loss function in respect to the output of the last layer, which is the intermediate output of softmax layer before the softmax function is applied. This intermediate output of the softmax layer is denoted as z_k in the preliminaries section, where $k \in Y$ denotes the corresponding category. When this gradient is determined, it can be easily propagated backwards to every weight in the previous layers as the softmax layer is fully connected to the previous layer of deep neural network. Then, the network can be adjusted to produce a better prediction with the gradient descent algorithm.

In the previous section, we have defined the new loss function in Equation (2.12). The gradient of the proposed loss function in respect of z_k is derived as follows:

$$\frac{\partial L(x,y)}{\partial z_k} = \frac{\partial CE(x,y)}{\partial z_k} - \lambda \frac{\partial M(x,y)}{\partial z_k} \tag{2.13}$$

The partial derivative of the cross-entropy term is:

$$\frac{\partial CE(x,y)}{\partial z_k} = -\sum_{k=1}^{K} 1_k \frac{\partial \, ln \, P\,(k|x)}{\partial z_k} = P(k|x) - 1_k \tag{2.14}$$

where $1_k = 1$ if $k = y$, and $1_k = 0$, otherwise.

The partial derivative of the margin term is:

$$\frac{\partial M(x,y)}{\partial z_k} = \frac{\partial P(y|x)}{\partial z_k} - \frac{\partial P(k_{max}|x)}{\partial z_k} \tag{2.15}$$

where $P(k_{max}|x)$ is the short denotation of $\max_k P(k \mid x)$, $k \in Y$ and $k \neq y$; k_{max} denotes the corresponding category. Thus, $P(k_{max}|x)$ is the maximum output probability of the competing categories.

According to the definitions of softmax function in Equation (2.9) in the previous section, thus,

$$\frac{\partial M(x,y)}{\partial z_k} = \begin{cases} P(y|x)\big(1 - P(y|x) + P(k_{max}\,|x)\big), & if\ k = y \\ -P(k_{max}\,|x)\big(1 - P(k_{max}\,|x) + P(y|x)\big), & if\ k = \ k_{max} \\ P(k|x)\big(P(k_{max}\,|x) - P(y|x)\big), & if\ k \neq \{y, k_{max}\} \end{cases} \tag{2.16}$$

Then, by putting the partial derivative of the cross-entropy term and the margin term together, the final partial derivative of the proposed loss function is:

If $k = y$, then

$$\begin{aligned} \frac{\partial L(x,y)}{\partial z_k} &= P(y|x)\left(1 - \lambda\big(P(y|x) - P(k_{max}|x)\big)\right) - 1 \\ &= P(y|x)\left(1 - \lambda\big(P(y|x) - M(x,y)\big)\right) - 1 \end{aligned} \tag{2.17}$$

If $k = k_{max}$, then

$$\begin{aligned} \frac{\partial L(x,y)}{\partial z_k} &= P(k_{max}\,|x)\left(1 + \lambda\left(1 + \big(P(y|x) - P(k_{max}\,|x)\big)\right)\right) \\ &= P(k_{max}\,|x)\big(1 + \lambda(1 + M(x,y))\big) \end{aligned} \tag{2.18}$$

If $k \neq \{y, k_{max}\}$, then

$$\begin{aligned}\frac{\partial L(x,y)}{\partial z_k} &= P(k\,|x)\left(1+\lambda\big(P(y|x)-P(k_{max}\,|x)\big)\right)\\ &= P(y\,|x)\big(1+\lambda M(x,y)\big)\end{aligned} \tag{2.19}$$

Comparing with the partial derivative of the original cross-entropy loss function (2.10), the partial derivative of the new loss function can be considered as multiplying a coefficient to constrain its value in respect to the margin between the target category and the competing categories. Thus, by adding the proposed margin term to the cross-entropy loss function, the original prediction model will be redirected towards the direction of enlarging the margin between the target category and the maximum competing category.

From the geometric interpretation of the proposed margin term, we can clearly see that the margin term constantly pushes the cross-entropy loss function towards the direction where the margin between the target category and the competing categories will be enlarged. From the mathematical derivation, it can be seen that the same effect of the margin term constrains the optimisation process towards the expected direction. Thus, the additional margin term can bring more discriminative power into the original prediction model without changing anything else. For example, we usually manipulate the hidden layers to expect the original neural network to achieve a better test performance.

We also derived the two margin terms proposed in [82], since they did not provide a clear geometric interpretation or a mathematical derivation to explain how the margin terms work. The loss functions corresponding to the two margin terms are called C1 and C2 in their paper [82]. C1 is defined as the following:

$$L_{c1}(x,y) = CE(x,y) + \lambda\big(1 - M(x,y)\big)^2 \tag{2.20}$$

Similarly, C2 is defined as:

$$L_{c2}(x,y) = CE(x,y) + \frac{\lambda}{K-1}\sum_{k\neq y}^{K}\Big(1 - \big(P(x,y) - P(k|x)\big)\Big)^2 \tag{2.21}$$

The margin term in $L_{c1}(x, y)$ is similar to our proposed margin term. However, it lacks the geometric interpretation. Furthermore, we derived the partial derivative of $L_{c1}(x, y)$ with respect to z_k as following:

If $k = y$, then

$$\frac{\partial L_{c1}(x,y)}{\partial z_k} = P(y|x)\Big(1 - 2\big(1 - M(x,y)\big)\lambda\big(1 - M(x,y)\big)\Big) - 1 \tag{2.22}$$

If $k = k_{max}$, then

$$\frac{\partial L_{c1}(x,y)}{\partial z_k} = P(k_{max}|x)\Big(1 + 2\big(1 - M(x,y)\big)\lambda\big(1 + M(x,y)\big)\Big) \tag{2.23}$$

If $k \neq \{y, k_{max}\}$, then

$$\frac{\partial L_{c1}(x,y)}{\partial z_k} = P(k|x)\Big(1 + 2\big(1 - M(x,y)\big)\lambda M(x,y)\Big) \tag{2.24}$$

Comparing with the partial derivative of the proposed loss function $L(x, y)$, the $L_{c1}(x, y)$ can be considered as multiplying a dynamic coefficient $2(1\text{-}M(x, y))$ by our proposed loss function. The $L_{c1}(x, y)$ might work well on some datasets since it pushes harder when $M(x, y)$ is negative. In such a case, the prediction model produces a wrong answer with the current input sample. However, when $M(x, y)$ increases, the dynamic coefficient $2(1\text{-}M(x, y))$ is decreased to zero. Thus, the ability to maintain the margin constraint is weak in $L_{c1}(x, y)$. Furthermore, we can choose a larger margin parameter λ for the proposed margin term to achieve comparable performance as the $L_{c1}(x, y)$ when dealing with the wrong predicted training samples, since the maximum value of the dynamic coefficient $2(1\text{-}M(x, y))$ is 4.

The $L_{c2}(x, y)$ can be considered as the mean squared value of 1 minus the margin between each competing category and the target category. Since it uses all the competing categories, there must be some redundancy information in this loss function, because the inputs of the loss function are all output probabilities and the sum of them equals 1. If one of them increases its value, some of the others must decrease respectively. Thus, $L_{c2}(x, y)$ cannot achieve an overall better performance than $L_{c1}(x, y)$, as explained in the original work [82].

In summary, from the view of mathematical derivation, the proposed loss function $L(x, y)$ can achieve a comparable performance to $L_{c1}(x, y)$, and both $L(x, y)$ and $L_{c1}(x, y)$ can achieve a better performance than $L_{c2}(x, y)$. However, both $L_{c1}(x, y)$ and $L_{c2}(x, y)$ suffer from a lack of geometric interpretation, whereas our proposed loss function $L(x, y)$ has a simple and clear geometric interpretation.

An RNN that employs the proposed loss function $L(x, y)$ is called LMRNN. The next section describes the experimental studies conducted to evaluate the proposed LMRNN on two different tasks and presents the performance of these three margin terms in practice.

2.4 Experiments

In order to verify the ability of the proposed LMRNN, two different experiments were conducted with two typical tasks that RNNs have shown to be successful.

Furthermore, in order to explore the differences between the original RNN model and the model with an additional margin term, all the configurations of the models in our experiments were kept the same as the baseline model except for the margin term.

We also tested the two models reported in [82], which also introduced additional large margin penalty terms into the cross-entropy loss function except for the authors only tested them on CNNs. Here, the RNN using the $L_{c1}(x, y)$ loss function is called LMRNN-C1, and the RNN using the $L_{c2}(x, y)$ loss function is called LMRNN-C2.

All the experiments were conducted on an HP-Z840 workstation with an NVIDIA GeForce GTX 1080 graphic card, and all the neural networks were impended using Theano deep learning framework [86].

2.4.1 MNIST dataset

The first experiment was conducted using the MNIST dataset [108]. This dataset is a well-known handwriting digit recognition dataset and has been widely used in machine learning studies. The standard dataset has ten categories and contains 60,000 images in the training set and 10,000 images in the test set. Each category has an equal number of images in both training and test sets. To find the best hyper-parameter, 10,000 images from the training set were randomly selected from each category to form a validation set, which means 1000 images per category.

Each image in this dataset is presented as a 28×28 matrix of pixels. Thus, in this experiment, we reshape the image into a long sequence with 28 time-steps and each time step has a vector of 28 pixels. At each time step, we input one column vector of the original image, which contains 28 pixels, and after 28 time-steps, the prediction model was given the whole image data and asked to predict which category the input image should be. Hence, this is a "many to one" task.

The prediction model in this experiment was IRNN. The configuration of IRNN in this experiment was similar to the network reported in [84]. The hidden recurrent node was 100 in the hidden layer. The recurrent weights were initialised to be an identity matrix with all zero biases. Nonetheless, the non-recurrent weights were initialised to be random matrices in Gaussian distribution with a mean of zero and the standard deviation of 0.001. The output vectors of the recurrent network's hidden states were fed into a softmax layer with ten nodes to generate the predicted probabilities of each category.

A typical mini-batch stochastic gradient descent optimisation method with a fixed learning rate 1e-8, a 0.9 momentum, and gradient clipping [-1, 1] was used in the training process. The size of the batch was set to 16. The training process concluded after it converged or when it reached 10,000 iterations and the network that achieved the highest validation accuracy was saved for testing on the test set.

We tested the margin parameter from one to ten with step-size one and chose the one that achieved the highest accuracy on the validation set as the best margin parameter. The best validation accuracy and the corresponding test accuracy are reported in Table 2.1.

Table 2.1 - Best results on MNIST dataset

Model	Margin Parameter	Validation Accuracy	Test Accuracy at max Valid
IRNN baseline	N/A	97.11	96.45
LMRNN	10	97.47	97.28
LMRNN-C1	2	97.61	97.02
LMRNN-C2	5	97.39	96.96

From Table 2.1, we can see that all the margin terms have achieved better results than the original IRNN baseline, and the margin term proposed in this thesis achieved the highest test accuracy on the test set. In particular, by applying the proposed margin term, the test error reduced from 3.55 to 2.72, which is about 23.4% error reduction. The LMRNN-C1 achieved the highest validation accuracy. However, it did not achieve the same success on the test set. The LMRNN-C2 achieved a better result than the original IRNN baseline, but it did not achieve the same success as LMRNN and LMRNN-C1.

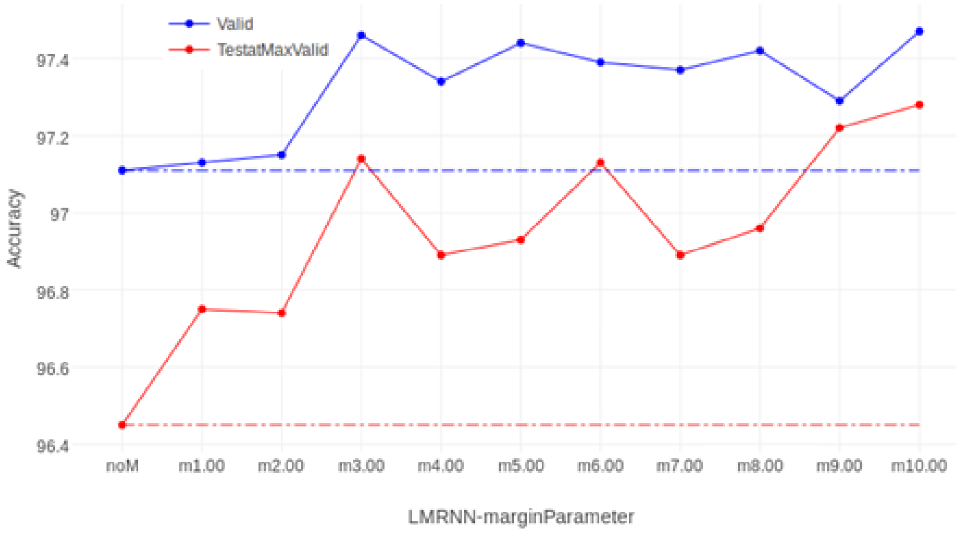


Figure 2.6 - LMRNN accuracy with different margin parameters

The best validation accuracy for each margin parameter and corresponding test accuracy are depicted in Figure 2.6, in which the LMRNN consistently outperformed the original IRNN with all the tested margin parameters. There was a wide range of the margin parameter to choose from for the LMRNN to achieve acceptable validation accuracy, which means the proposed LMRNN had stable performance.

We also noticed that by adding the margin term, the training process converged faster to the highest validation accuracy. The average iteration to achieve the highest validation accuracy for the original IRNN was about 7852 iterations on average; in contrast, by adding the margin term, that average iteration decreased to 2778 iterations.

The experimental results prove that the proposed margin term enables the original IRNN to achieve superior performance. Particularly, by adding the proposed margin term, the original IRNN had a better generalisation ability on the test set and converged faster to the highest validation accuracy.

2.4.2 Penn Treebank Corpus

In the second set of experiments, we applied the proposed model to a language-modelling task in which each sample was a sentence containing a vector of words. At each time step, the prediction model took one word from the sentence as input and produced an output that indicated the upcoming word of the sentence. Thus, both the input and output of the task were a sequence of words. This kind of tasks is usually referred to as “many to many” or “sequence to sequence” tasks. The purpose of the training process was to produce the same sentence as the input sentence but predicting one word ahead with respect to the current given sequence of words. Thus, at each time step, it is still a classification task, which is required to produce a vector of output probabilities corresponding to each word in the vocabulary, and the word with the maximum probability in the vector is chosen to be the predicted word.

Table 2.2 - Configurations of each model

Models	Hidden cells	Uniform range	Time steps	Training epochs	Decay rate	Start epoch
Small-LSTMs	200	±0.1	20	15	1/2	6
Medium-LSTMs	650	±0.05	35	40	1/1.2	6
Large-LSTMs	1500	±0.04	35	55	1/1.15	15

The corpus used in this experiment was the standard split of the Penn Treebank corpus [88], which contains 929k words in the training set, 73k words in the validation set and 82k words in the test set. The vocabulary of this corpus consists of 10k unique words. Thus, the classification task has 10k categories in each time step.

We employed the same network configuration and implementation as reported in [89] but changed the loss function.

The prediction model employed in this experiment was the LSTM network. Three different sizes of LSTMs were used in this experiment, and each model consisted of two stacked LSTM layers, and a summary of the configurations is presented in

Table 2.2. Each size of the models was constructed with the given number of hidden cells, and all the parameters were initialised with a uniform distribution in the given range in Table 2.2.

All the networks were trained with mini-batch stochastic gradient descent optimisation method with the mini-batch size of 32. Each of the models back-propagated with the given time steps and trained with the given epochs. Likewise, in the medium and large LSTMs, the dropout technique [90] with a probability of 50% was also applied between all layers. The learning rate in each of the models was decayed with the given decay rates in every epoch after the given starting epochs.

Initially, we tested the margin parameter from 1 to 5 with step-size 0.5 and found that all the models achieved the best result when the margin parameter was set to 1. Hence, we proceeded with testing the margin parameter from 0.1 to 1 with step-size 0.1. We used the perplexity as the criteria to evaluate the performance of the prediction model, which is one of the popular criteria for language modelling tasks. The lower the perplexity is, the better the model is. We chose the margin parameter that achieved the lowest perplexity on the validation set as the best margin parameter.

The margin parameters and the corresponding perplexities on the training set and validation set are reported in Figure 2.7. Furthermore, the best perplexities on the training and validation sets for each model are reported in Table 2.3.

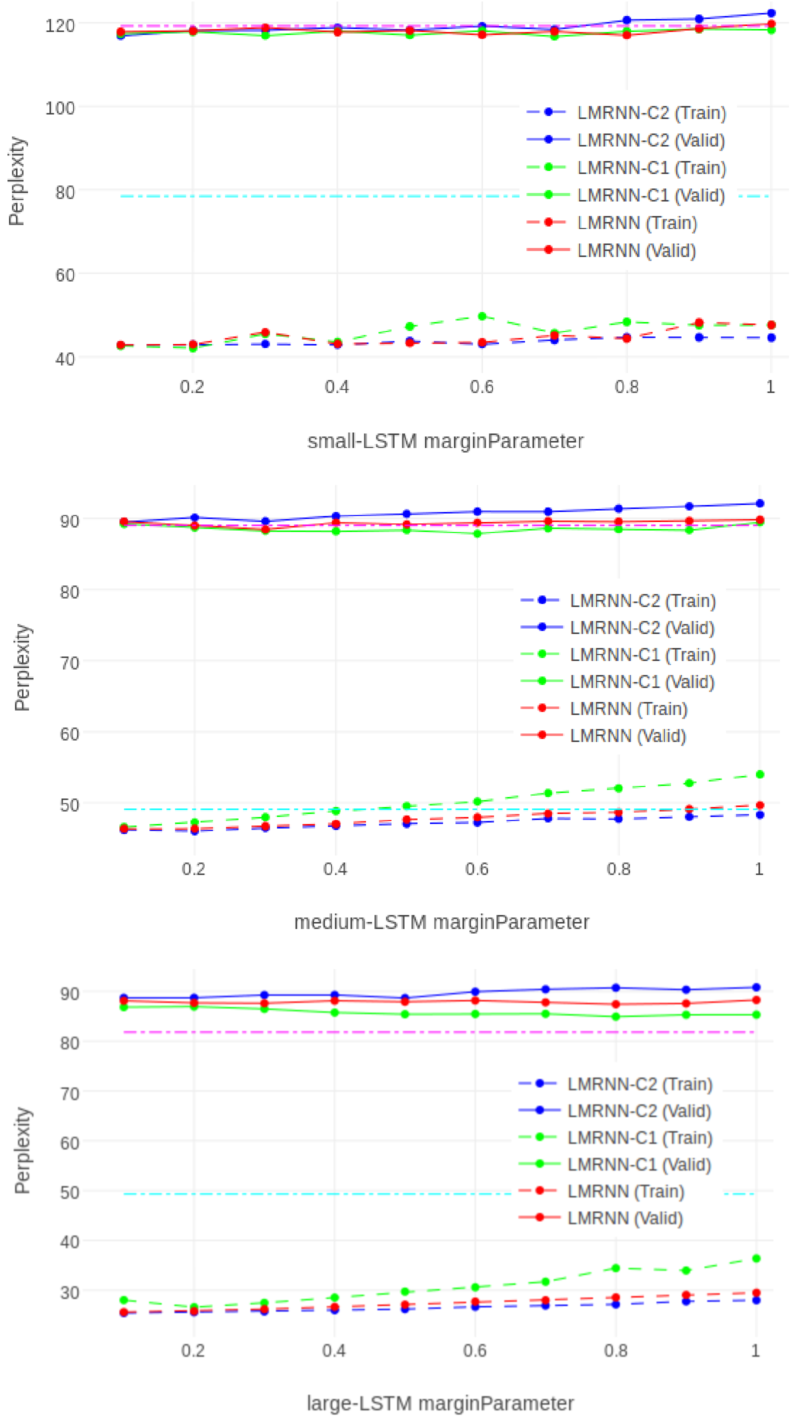


Figure 2.7 - Margin parameter and corresponding training and validation perplexities

Table 2.3 - Best perplexities of each model

Models	Margin Parameter	Training Perplexity	Validation Perplexity
Small-baseline	N/A	78.5	119.2
Small-LMRNN	0.8	44.37	117.01
Small-LMRNN-C1	0.7	45.69	116.71
Small-LMRNN-C2	0.1	42.74	116.87
Medium-baseline	N/A	49.1	89.0
Medium-LMRNN	0.3	46.72	88.46
Medium-LMRNN-C1	0.6	50.17	87.87
Medium-LMRNN-C2	0.1	46.18	89.5
Large-baseline	N/A	49.3	81.8
Large-LMRNN	0.1	28.56	84.78
Large-LMRNN-C1	0.8	34.43	84.92
Large-LMRNN-C2	0.5	26.19	88.66

From Table 2.3, we can see that the three margin terms delivered superior performances over the original baselines on the training set. However, they did not achieve the same success on the validation set because the original models already over fitted the dataset, especially the medium and large LSTMs.

Comparing the models based on medium LSTM and the models using large LSTM, it is obvious that the model that achieved the lowest perplexity on the training set achieved the highest perplexity on the validation set. Nevertheless, by only adding a margin term, the original baseline models achieved significantly lower perplexities, which means that the margin term can bring additional discriminative power to the original model. In addition, the three margin terms achieved comparable performance.

2.4.3 Discussion

In order to further investigate the effectiveness of the margin term, we recorded the mean value of the cross-entropy term and the margin term on every iteration in the MNIST experiments during the training process for all three margin enhanced loss functions.

Figure 2.8 shows the value of the cross-entropy term and the margin term on each iteration in the MNIST experiments. As can be seen, the proposed margin term value in $L(x, y)$ followed the same trend as the margin term in $L_{c1}(x, y)$ and $L_{c2}(x, y)$, whereas its value shifted down below 0. The value of the cross-entropy term in $L_{c1}(x, y)$ dropped faster than the other two at the beginning of the training process. This can demonstrate our mathematical interpretation presented in the optimization section that the $L_{c1}(x, y)$ pushes harder when the predication produces wrong answers due to the dynamic coefficient $2(1\text{-}M(x, y))$ making the value of the partial derivation up to nearly four times larger than that in $L(x, y)$.

However, at the end of the training process, the $L(x, y)$ delivered a lower value of the cross-entropy term because the dynamic coefficient $2(1-M(x, y))$ decreased to zero when the predication model produced correct answers, which made the $L_{c1}(x, y)$ fail to keep the training process in the correct direction.

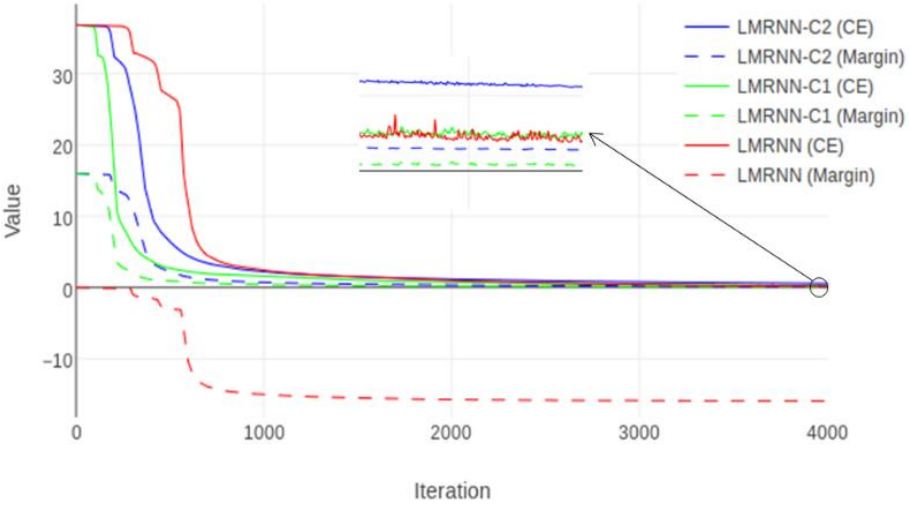


Figure 2.8 - Values of cross-entropy term and margin term in each iteration during training

Both $L(x, y)$ and $L_{c1}(x, y)$ produced comparable low values of the cross-entropy term, whereas $L_{c2}(x, y)$ did not. This confirms what we discussed in Section 2.3.3.

All three margin-enhanced loss functions tend to achieve performance at a similar degree. This is because all these additional margin terms utilise the same information provided by the competing categories. This information is very limited, since only the cross-entropy term can achieve a good result already, and the margin term itself does not have proper optimisation property as the cross-entropy term. Thus, the additional margin term can only help the cross-entropy term in achieving better results by bringing in additional discriminative information, but it cannot replace the cross-entropy term to perform the primary optimisation during the training process.

In summary, the experimental results demonstrate our mathematic interpretation and prove that the additional margin terms can help RNNs to achieve superior generalisation results.

2.5 Conclusions

In this chapter, a novel Large Margin Recurrent Neural Network (LMRNN) is developed, which maintains the generative ability of the original RNN model and extends its discriminative ability by introducing the large margin principle into the cross-entropy loss function.

As discussed in detail in Section 2.3.1 with an example of 5-categories classification task, the cross-entropy loss function clearly shows its drawback on such a task. This drawback is that the information provided by the competing categories is not fully exploited by the cross-entropy loss function. This is because the cross-entropy loss function only takes the output probability corresponding to the target category into account and ignores all other probabilities from competing categories.

To solve this issue, a large margin term is introduced into the cross-entropy loss function. The geometric interpretation of the developed large margin term is discussed in Section 2.3.2, together with the mathematic derivation in Section

2.3.3. Both of the geometric interpretation and mathematic derivation clearly show that the developed large margin term can help the cross-entropy loss function to exploit the information from competing categories during the training process.

Experimental results on the MNIST dataset and the Penn Treebank corpus demonstrate that the proposed margin term cooperates well with RNNs and the proposed model achieves significant performance improvement over the original model. From further discussion on the training process in Section 2.4.3, the developed LMRNN clearly shows its advantages on the MNIST dataset and demonstrates the mathematic interpretation.

The limitation of the developed large margin term is that it can only be effective on the classification tasks with more than two categories. This is because the large margin cross-entropy loss function will degenerate into a normal cross-entropy loss function on the binary classification tasks, as the probability of the competing category equals to one minus the probability of the target category. In other words, the information on the competing category has already been implicitly used.

It is worth mentioning that the margin term can help the model converge during the training process, which will reduce the computational costs and accelerate the training process. This is extremely helpful for Neural Architecture Search (NAS) tasks. A detailed discussion can be found in Section 5.2 Future works.

Chapter 3 Robust Multi-View Continuous Subspace Clustering

This chapter presents the second developed approach, named Robust Multi-View Continuous Subspace Clustering (RMVCSC), which fulfils the second research objective.

A novel Robust Multi-View Continuous Subspace Clustering (RMVCSC) approach is developed in this chapter, which can untangle heavily mixed clusters by optimising a single continuous objective. The proposed objective uses robust estimators to automatically clip specious inter-cluster connections while maintaining convincing intra-cluster correspondences in the common representation subspace learned from multiple views. The common representation subspace can reveal the underlying cluster structure in data. RMVCSC is optimised in an alternating minimisation scheme, in which the clustering results and the common representation subspace are simultaneously optimised. Since different views can describe distinct perspectives of input data, the proposed approach has more accurate clustering performance than conventional approaches by exploring information among multi-view data. In other words, the proposed approach optimises a novel continuous objective in the simultaneously learned common representation subspace across multiple views. By using robust redescending estimators, the proposed approach is not prone to stick into bad local minima even with outliers in data. This kind of robust continuous clustering approaches has never been used for multi-view clustering before. Moreover, the convergence of the proposed approach is theoretically proved, and the experimental results show that the proposed RMVCSC can outperform several very recently proposed approaches in terms of clustering accuracy.

This chapter is organised as follows. Section 3.1 introduces the motivation of the developed approach. Two most relevant topics are reviewed in Section 3.2. The model formulation is presented in Section 3.3, followed by the optimisation process in Section 3.4. The convergence of the proposed approach is analysed in Section 3.5. Section 3.6 presents the experiments, and Section 3.7 summarises the conclusion.

3.1 Introduction

Clustering is one of the main approaches for data mining and statistical analysis, which is the process of partitioning a data set into different subsets according to some defined measures [110]. In real-world applications, multi-view data are obtained naturally, since data are often collected across multiple domains or extracted by different feature extractors [111]. Each view can be considered as a distinctive perspective of the data [112]. For example, a webpage can be described according to the contents of this webpage, the webpage contents linked to this webpage and the link structures used by this webpage, while an image can be described according to its colours, textures, shapes and so on. Thus, exploring information among multiple views to create accurate multi-view clustering approaches is beneficial for big data analysis [113][114].

Multi-view clustering is a machine learning paradigm, which aims to leverage the complementary information among multiple views to improve the clustering accuracy and generalisation ability [111][113][114]. There are mainly two approaches to do the multi-view clustering [114]. The first one is the fusion approach, which fuses similarity measurements from multiple views to construct a graph for clustering [115][116]. The other one is the subspace-clustering approach, which aims to learn a common latent subspace for all the multiple views [116][117][118][119]. Since the subspace approach can reveal the underlying cluster structure in multi-view data and achieves the state-of-the-art performance [116], multi-view subspace clustering has attracted arising attention in the past years.

Multi-view subspace clustering performs clustering on a common subspace representation of all the views simultaneously with the assumption that all the views are generated from this latent subspace [120][121]. Many multi-view subspace clustering approaches have been developed in recent years [118][127][128][129][130], such as iteration based approaches [120][121], factorization based approaches [122][123], statistical approaches [124], and spectral clustering based approaches [125][126].

Although multi-view subspace clustering has permeated into many fields and has made a great performance, there are still some limitations. Especially, since most existing multi-view subspace clustering approaches are based on *k*-means or spectral clustering, the number of clusters *k* and the weights of different views are required to be pre-set manually. This may limit the further advancement of multi-view subspace clustering.

More recently, Shah *et al.* [131] proposed a Robust Continuous Clustering (RCC) algorithm, which does not need to know the number of clusters in advance and has the ability to achieve high accuracy efficiently even the data is in high-dimension. RCC optimised a clear continuous objective by using standard numerical methods and has the ability to be integrated into a dimensionality reduction system. However, RCC has not been integrated into a multi-view subspace clustering system yet.

In this chapter, a novel Robust Multi-View Continuous Subspace Clustering (RMVCSC) approach is proposed to untangle heavily mixed clusters by optimising a single continuous objective. We use the self-expressiveness property of multi-view data, which is proposed in [132] to learn a common representation subspace across multiple views. By using robust redescending estimators, the proposed approach is optimised in an alternating minimisation scheme, in which the clustering results and the common representation subspace are simultaneously optimised. Without the requirement of the number of clusters given in advance, RMVCSC is not prone to stick into bad local minima even with outliers in data and is insensitive to initialisation.

Over the iteration of the proposed approach, the representatives will move and merge into several discrete clusters. This kind of robust continuous clustering approaches has never been used for multi-view clustering before. The proposed RMVCSC approach can outperform several very recently proposed approaches in terms of clustering accuracy.

The rest of this chapter is organised as follows: Section 3.2 reviews the most relevant topics. Section 3.3 formulates our proposed RMVCSC approach; Section 3.4 introduces the optimisation process; Section 3.5 analyses the convergence behaviour; experimental results and conclusions are presented in Section 3.6 and Section 3.7.

3.2 Related Work

In this section, we review the approaches in multi-view subspace clustering and continuous clustering, which are the two most relevant topics to our developed approach.

3.2.1 Multi-view Subspace Clustering

Multi-view subspace clustering aims to find the shared latent subspace for all the views of a data set and obtain the segments of the data set in this subspace [133]. As this subspace is jointly learned from all the views using the self-expression property of the data set, it can represent the data set and unveil the underlying cluster structure of the data set.

Currently, many multi-view subspace clustering approaches have been proposed. Gao *et al.* [134] proposed a multi-view subspace clustering model which performs subspace clustering on each view and guarantees the consistency of the clustering structure among different views. Zhang *et al.* [135] performed clustering on multi-views simultaneously with a low-rank tensor constraint, which is constructed by subspace representation matrices. Ding *et al.* [136] proposed a multi-view subspace clustering approach via dual low-rank decompositions, which expects to find a low-dimensional view-invariant subspace for multi-view data. Fan *et al.* [137] proposed a localised multi-view subspace clustering model by fusing noiseless structures among views and samples. Zhuge *et al.* [116] proposed an auto-weighted multi-view subspace clustering approach based on a common subspace representation matrix.

However, most of these works are based on the *k*-means and its variants. Thus, their performance are sensitive to the choice of the cluster numbers.

3.2.2 Continuous Clustering

Continuous clustering is another topic related to our proposed approach in Chapter 3. The main idea of continuous clustering is to transform the clustering problem into a continuous optimisation problem [115]. Lindsten *et al.* [138] proposed a formulation, which can relax *k*-means clustering to convex optimisation problems. Hocking *et al.* [139] proposed a convex relaxation of hierarchical clustering in calculating continuous regularisation. Chi *et al.* [140]

proposed a splitting method for solving convex clustering problem. Chi *et al.* [141] proposed a convex bi-clustering algorithm Convex Bicluste Ring Algorithm (COBRA), which settles on a graph-based representation of both samples and features. All the works mentioned above are regularised by convex function (l_2-norm).

In addition, Shah *et al.* [131] proposed Robust Continuous Clustering (RCC) algorithm, regularisation using a non-convex function (Geman-McClure). RCC does not have the prior knowledge of the number of clusters, and it has the ability to achieve high accuracy efficiently, even the data is in high-dimension. He *et al.* [115] proposed an optimisation method for Robust Continuous Co-Clustering (ROCCO), which formulated a co-clustering problem as a continuous non-convex optimisation problem. ROCCO learns the representation regularised on both samples and feature graphs.

3.3 Model Formulation

In this section, we introduce the subspace representation and formulation of RMVCSC.

In multi-view clustering, the clustering results of different views should be consistent, which means the clustering assignments of all the views should be the same [117]. As multi-view data is collected across different domains, different views may show a considerable divergence when learning a consensus representation. Thus, multi-view subspace representation is used in the proposed approach to learn a common view-invariant subspace while reducing the influence of view-variance [113].

Consider the problem of spectral-based subspace representation. The *i*th row, *j*th column and *ij*th element in a matrix *M* can be denoted as $m_{i:}$, $m_{:j}$, and m_{ij} respectively. Suppose single-view data matrix is $X = [x_{:1}, x_{:2}, \ldots, x_{:n}] \in \mathbb{R}^{d \times n}$ which includes *n* data points in *d* dimensions. If *X* presents multi-view data, the data matrix of *X* in the *v*th view can be denoted as $X^v \in \mathbb{R}^{dv \times n}$.

Thus, based on the self-expressiveness property, the data matrix of *X* in the *v*th view is represented as:

$$X^v = X^v Z + E^v \tag{3.1}$$

where $Z = [z_{:1}, z_{:2}, \ldots, z_{:n}] \in \mathbb{R}^{n \times n}$ is the self-representation matrix, in which each $z_{:i}$ is the representative of data point $x_{:i}$, and E^v is an error matrix. The nonzero elements of $z_{:i}$ correspond to the data points from the same subspace.

Since the input data is denoted as $X^v \in \mathbb{R}^{dv \times n}$ and *Z* is the self-representation matrix of X^v, the new representatives are initialised based on *X*. After that, all steps of RMVCSC are operated by the new representatives. Over times of iteration, the new representatives will migrate and merge into several discrete clusters.

The objective function of RMVCSC is defined as follows:

$$\Phi(Z) = \|X^v - X^v Z\|_{2,p}^{p} + \lambda \Omega^v(Z) \tag{3.2}$$

where $\|\ \|_{2,p}^{p}$ is the sparsity-inducing norm with $0 \le p \le 1$; λ is a trade-off factor, and $\Omega^v(Z)$ is a smooth regularizer on Z.

Given the representation error matrix $E^v \in \mathbb{R}^{dv \times n}$ of the vth view as:

$$E^v = X^v - X^v Z \tag{3.3}$$

the $\|\ \|_{2,p}^{p}$-norm of the representation error matrix E^v can be defined as:

$$\|E^v\|_{2,p}^{p} = \sum_{i=1}^{d^v}\left(\sum_{j=1}^{n}\left|e_{ij}^{v}\right|^2\right)^{\frac{p}{2}} = \sum_{i=1}^{d^v}(\|e_{i:}^{v}\|_2)^p \tag{3.4}$$

where $e^v{}_{i:}$ is the ith row of E^v, and $\|E^v\|_{2,p}^{p}$ is a $\ell_{2,p}$-norm [142].

$\Omega^v(Z)$ aims to smooth the distribution of the common representation Z on the vth view. The common subspace representation matrix Z will be enforced to meet the grouping effect using $\Omega^v(Z)$.

Based on the original data X and the new representatives Z, a graph is constructed automatically by using *m*-*k*NN graphs, a variant of the standard *k*NN graphs [143]. Compared with standard *k*NN graphs, all vertices in an *m*-*k*NN graph have a *k*-upper bound, which helps the graph not to produce vertices (hub vertices) with an extremely high degree and is more robust for utilising.

Specifically, for $v = 1, 2, ..., m$, each regularised term $\Omega^v(Z)$ in our proposed approach is defined as:

$$\Omega^v(Z) = \frac{1}{2}\sum_{(s,t)\in\varepsilon} w_{s,t}^v \rho\left(\|z_s - z_t\|_{2,p}^p\right) \tag{3.5}$$

Thus, the objective function of RMVCSC can be rewritten as:

$$\Phi(Z) = \sum_{v=1}^{m}\left(\sum_{i=1}^{n}\|x_i^v - x_i^v z_i\|_{2,p}^p + \frac{\lambda}{2}\sum_{(s,t)\in\varepsilon} w_{s,t}^v \rho\left(\|z_s - z_t\|_{2,p}^p\right)\right) \tag{3.6}$$

where (*s*, *t*) means there is a connection between data x_s and data x_t, and *ε* is the edge set of this graph; weights $w_{s,t}^v$ measure the strength of each data to the pairwise terms that exist, and *λ* is used to measure the proportion of each objective term to the whole; function *ρ*(·) is a penalty on the regularisation terms.

Since our approach is based on the duality between robust estimation and line processes [131], an auxiliary variable $h_{s,t}^v$ is introduced for each connection (*s*, *t*) ∈ *ε*. Thus, a joint objective over the representatives *Z* and the line process *H* $=\{h_{s,t}^v\}$ is proposed:

$$\Phi(Z,H) = \sum_{v=1}^{m}\left(\sum_{i=1}^{n}\|x_i^v - x_i^v z_i\|_{2,p}^p \right. \\ \left. + \frac{\lambda}{2}\sum_{(s,t)\in\varepsilon} w_{s,t}^v \left(h_{s,t}^v\|z_s - z_t\|_{2,p}^p + \Psi\left(h_{s,t}^v\right)\right)\right) \tag{3.7}$$

where $\Psi(h_{s,t}^v)$ is a penalty on ignoring a connection (*s*,*t*), i.e., when the connection is active (i.e., $h_{s,t}^v \to 1$) then $\Psi(h_{s,t}^v)$ tends to zero; when the connection is disabled (i.e., $h_{s,t}^v \to 0$) then $\Psi(h_{s,t}^v)$ tends to one. Each robust estimator *ρ*(·) has its own corresponding penalty function $\Psi(\cdot)$ so that Equation (3.6) and Equation (3.7) are equivalent with respect to the representatives of X. In other words, the same set of Z will be produced by optimising either of these two objectives. Equation (3.7) is based on the iteratively reweighted least squares. However, it is more flexible because of the explicit variables H and the additional terms defined by these variables. Although many different gradient-based methods can be used to optimise Equation (3.7), the iterative solution of linear least-squares systems can achieve more efficient and scalable optimisation [144].

Although RMVCSC can accommodate different estimators within the same computational efficiency framework, our presentations and experiments are all based on a well-known estimator: Geman-McClure estimator [145],

$$\rho(y) = \frac{\mu y^2}{\mu + y^2} \tag{3.8}$$

where μ is a scale parameter. The corresponding penalty function that makes Equation (3.6) and Equation (3.7) equivalent with respect to the representatives is:

$$\psi(h_{s,t}) = \mu\left(\sqrt{h_{s,t}} - 1\right)^2 \tag{3.9}$$

3.4 Optimisation

Based on Equation (3.7) we observe that (i) when the variable H is fixed, Equation (3.7) will transform into a linear least-squares problem; (ii) when the variable Z is fixed, the decoupling of individual pairwise terms and the optimal value of each connection $h_{s,t}^{v}$ can be calculated independently. Based on these two conditions, the objective can be optimised by updating the variable sets Z and H alternatingly. As a block coordinate descent approach, this alternating minimisation scheme provably converges.

The first step is fixing the variable H, updating the common subspace representation Z. When the variable H is fixed, Equation (3.2) can be rewritten in a matrix form to obtain a simplified expression by solving the variable Z:

$$\min_{Z} \sum_{v=1}^{m} \|X^{v} - X^{v}Z\|_{2,p}^{p} + \lambda\,\Omega^{v}(Z) \tag{3.10}$$

Intuitively, there are no weight factors explicitly defined in Equation (3.10), so that all different views are treated equally. Thus, the Lagrange function of Equation (3.10) can be written as:

$$\sum_{v=1}^{m} \|X^{v} - X^{v}Z\|_{2,p}^{p} + \lambda\,Tr(ZL^{v}Z^{T}) \tag{3.11}$$

$L^{v} = D^{v}$-Q^{v} is the Laplacian matrix, in which D^{v} is a diagonal matrix; Q^{v} measures the spatial closeness of the data points on vth view. $Q^{v} = [q_{11}, q_{12}, \ldots, q_{nn}] \in \mathbb{R}^{n\times n}$, where $q_{st} = q_{ts} = w_{s,t}^{v} \cdot h_{s,t}^{v}$ when $w_{s,t}^{v}$ and $h_{s,t}^{v}$ is nonzero, otherwise $q_{st} = q_{ts} = 0$.

Taking the derivative of Equation (3.11) with respect to Z and setting the derivative to zero, we have:

$$\sum_{v=1}^{m} \frac{\partial \left(\|X^v - X^v Z\|_{2,p}^{p} + \lambda Tr(Z L^v Z^T) \right)}{\partial Z} = 0 \tag{3.12}$$

In order to solve Equation (3.12), we consider the following problem to tackle a non-smooth norm problem:

$$\min_{Z,U^v} \sum_{v=1}^{m} H^v + \lambda\, Tr(Z L^v Z^T) \tag{3.13}$$

where

$$H^v = Tr\left((X^v - X^v Z)^T U^v (X^v - X^v Z) \right) \tag{3.14}$$

$U^v \in \mathbb{R}^{dv \times dv}$ is a diagonal matrix corresponding to the vth view. The ith entry on the diagonal is defined as:

$$u_{ii}^{v} = \frac{p}{2} \|e_{i:}^{v}\|_2^{p-2}, \forall i = 1,2, \dots, d^v \tag{3.15}$$

Then differentiating the objective function with respect to Z and setting it to zero:

$$AZ + ZB + C = 0 \tag{3.16}$$

where

$$A = \sum_{v=1}^{m} X^{v^T} U^v X^v \tag{3.17}$$

$$B = \lambda \sum_{v=1}^{m} L^v \tag{3.18}$$

$$C = -\sum_{v=1}^{m} X^{v^T} U^v X^v \tag{3.19}$$

Equation (3.16) is a standard Sylvester equation, which has a unique optimal solution [109].

As discussed in Section 3.2, the proposed objective function Equation (3.7) is a joint objective over the representatives *Z* and the line process *H* =$\{h_{s,t}^{v}\}$. Thus, the second step is fixing the variable *Z*, then the optimal value of each connection $h_{s,t}^{v}$ is calculated by:

$$h_{s,t}^{v} = \left(\frac{\mu}{\mu + \|z_s - z_t\|_2}\right)^2 \tag{3.20}$$

According to the above two steps, we alternatively update *Z* and *H*, and repeat the process iteratively.

Algorithm 1 shows the whole process of RMVCSC. Note that all updates to *Z* and *H* optimise the same continuous global objective in Equation (3.7). Step I and II are the input and output statements of the proposed approach. Step III to VI are the initialisation steps, which are discussed in Section 3.2 above Equation (3.5). Step VII to XII are the main optimisation steps, which are discussed in detail in this section. Step XIII and XIV are the output steps of the final clustering results.

Algorithm 1. RMVCSC

```
Input: Data for $m$ views $\{X^1, \ldots, X^m\}$ and $X^v \in \mathbb{R}^{dv \times n}$.
Output: Cluster assignment $\{\phi_i\}_{i=1}^{n}$.
Construct connectivity structure $\varepsilon$.
Precompute $\chi^v = \|X^v\|_2$, $w_{s,t}^v$, $\delta$.
Initialise $h_{s,t}^v = 1$, $\mu >> \max\|x_s - x_t\|_2^2$, $\lambda = \Sigma^m(\chi^v / \|L^v\|_2)$.
Initialise the feature weight matrix $U^v = I^v$ for each view, where $I^v \in \mathbb{R}^{dv \times dv}$ is the identity matrix.
While $|\phi^t - \phi^{t-1}| < \varepsilon$ or $t <$ max-iterations do:
    Compute the common representation $Z$ by solving Equation (3.16) with Equations (3.17), (3.18) and (3.19).
    Update the diagonal feature weight matrix $U^v$ for each view by Equation (3.15).
    Update $h_{s,t}^v$ with Equation (3.20) and $L^v = D^v - Q^v$.
    Every four itrations, update $\lambda = \Sigma^m(\chi^v / \|L^v\|_2)$ and $\mu = max(\mu/2, \delta/2)$.
End while
Construct graph $G = (V, F)$ with $f_{s,t} = 1$ if $\|z_s^* - z_t^*\|_2 < \delta$.
Output clusters are given by the connected components of $G$.
```

3.5 Convergence Analysis

In order to prove the convergence of our proposed approach can reach at least a locally optimal solution, we first introduce the following lemma [146].

Lemma 1: When *0 < p ≤ 2*, for any positive number *a* and *b*, the inequality holds:

$$a^p - \frac{p}{2}\frac{a^2}{b^{2-p}} \leq b^p - \frac{p}{2}\frac{a^2}{b^{2-p}} \tag{3.21}$$

The first step is fixing the variable *H*, updating the common subspace representation *Z*.

Theorem 1: Each updated *Z* in Algorithm 1 will monotonically decrease the objective in Equation (3.13) in each iteration.

Proof: Denote $\tilde{Z}$ as the updated Z in each iteration and $\tilde{E}^v = X^v - X^v\tilde{Z}$ is the *v*th representation error matrix calculated by $\tilde{Z}$. According to the optimisation to $\tilde{Z}$ in Algorithm 1, $\tilde{Z}$ reaches the unique optimal solution of Equation (3.10) when *U*^{v} are fixed. Thus,

$$\begin{aligned}&\sum_{v=1}^{m}\left(Tr\left(\tilde{E}^{vT}U^v\tilde{E}^v\right) + \lambda Tr\left(\tilde{Z}L^v\tilde{Z}^T\right)\right)\\ &\qquad\leq \sum_{v=1}^{m}\left(Tr\left(E^{vT}U^vE^v\right) + \lambda Tr\left(ZL^vZ^T\right)\right)\end{aligned} \tag{3.22}$$

Combining weight matrix *U*^{v} which

$$u_{ii}^{v} = \frac{p}{2} \| e_{i:}^{v} \|_2^{p-2} \tag{3.23}$$

the inequation can be rewritten as:

$$\sum_{v=1}^{m} \left(\sum_{i=1}^{d^v} \frac{p}{2} \frac{\| \tilde{e}_{i:}^{v} \|_2^2}{\left\| e_{i:}^{v} \right\|_2^{2-p}} + \lambda Tr\left(\tilde{Z} L^v \tilde{Z}^T\right) \right) \leq \sum_{v=1}^{m} \left(\sum_{i=1}^{d^v} \frac{p}{2} \frac{\| e_{i:}^{v} \|_2^2}{\left\| e_{i:}^{v} \right\|_2^{2-p}} + \lambda Tr(Z L^v Z^T) \right) \tag{3.24}$$

Generally, $\| e_{i:}^{v} \|_2 > 0$ and $\| \tilde{e}_{i:}^{v} \|_2 > 0$, the regularised *l2,p*-norm can be used to guarantee it. According to Lemma 1, we can derive

$$\| \tilde{e}_{i:}^{v} \|_2^p - \frac{p}{2} \frac{\| \tilde{e}_{i:}^{v} \|_2^2}{\left\| e_{i:}^{v} \right\|_2^{2-p}} \leq \| e_{i:}^{v} \|_2^p - \frac{p}{2} \frac{\| e_{i:}^{v} \|_2^2}{\left\| e_{i:}^{v} \right\|_2^{2-p}} \tag{3.25}$$

Thus, the following inequality holds

$$\sum_{v=1}^{m} \sum_{i=1}^{d^v} \| \tilde{e}_{i:}^{v} \|_2^p - \sum_{v=1}^{m} \sum_{i=1}^{d^v} \frac{p}{2} \frac{\| \tilde{e}_{i:}^{v} \|_2^2}{\left\| e_{i:}^{v} \right\|_2^{2-p}} \leq \sum_{v=1}^{m} \sum_{i=1}^{d^v} \| e_{i:}^{v} \|_2^p - \sum_{v=1}^{m} \sum_{i=1}^{d^v} \frac{p}{2} \frac{\| e_{i:}^{v} \|_2^2}{\left\| e_{i:}^{v} \right\|_2^{2-p}} \tag{3.26}$$

Summing Equation (3.24) and Equation (3.26), we have

$$\sum_{v=1}^{m} \left(\left\| X^v - X^v \tilde{Z} \right\|_{2,p}^{p} + \lambda Tr\left(\tilde{Z} L^v \tilde{Z}^T\right) \right) \leq \sum_{v=1}^{m} \left(\| X^v - X^v Z \|_{2,p}^{p} + \lambda Tr(Z L^v Z^T) \right) \tag{3.27}$$

Thus, each updated Z will monotonically decrease *Φ(Z, H)* in each iteration, which means the following inequality holds:

$$min\,\Phi\,(\tilde{Z},H) \leq min\,\Phi\,(Z,H) \tag{3.28}$$

The second step is fixing the variable *Z*, updating the variable *H*.

Since Equation (3.20) is the optimal solution of *H*, this step will have a unique optimal solution *H* decreasing our objective function *Φ(Z, H)*. Moreover, the second-order partial derivatives of *Φ(Z, H)* with respect to *H* is greater than zero. Thus, it is obvious that each updated *H* will monotonically decrease *Φ(Z, H)* in each iteration, which means the following inequality holds:

$$min\,\Phi\,(\tilde{Z},\widetilde{H}) \leq min\,\Phi\,(\tilde{Z},H) \tag{3.29}$$

To sum up, the alternately updated *Z* and *H* in Algorithm 1 can monotonically decrease the objective function in the iteration process.

$$min\,\Phi\,(\tilde{Z},\widetilde{H}) \leq min\,\Phi\,(\tilde{Z},H) \leq min\,\Phi\,(Z,H) \tag{3.30}$$

3.6 Experiments

In this section, we evaluate the RMVCSC approach and several reference approaches on three widely used datasets. Experimental results show their convergence behaviour.

3.6.1 Dataset Descriptions

Three multi-view benchmark datasets, which are commonly used for multi-view learning, have been used to validate the effectiveness of RMVCSC. They are Caltech101 [147], Handwritten Dutch Digit Recognition (Digit) [148], and Web Knowledge Base (WebKB) [149]. The statistics information about these three datasets is concluded in Table 3.1.

Table 3.1 - Details of the multi-view datasets

View type	WebKB	Caltech101-7	Digit
1	Fulltext (2949)	LBP (256)	FOU(76)
2	Inlinks (334)	PHOG (680)	FAC(216)
3	-	GIST (512)	KAR(64)
4	-	Gabor (32)	PIX(240)
5	-	SURF (200)	ZER(47)
6	-	SIFT (200)	MOR(6)
Data points	1051	441	2000
Classes	2	7	10

All the experiments are following the 5-fold cross-validation scheme. Each dataset is randomly split into five subsets equally. Each clustering approach is tested on a selected subset and trained on the rest of the subsets. The final results are reported as the average of these 5 clustering results.

The Caltech101-7 dataset is composed of 8677 objective images, which belong to 101 categories. Following [150], we selected seven widely used classes, including DollaBill, Faces, Garfield, Motorbikes, Snoopy, Stop-Sign and Windsor-Chair, which have 441 images in total. In order to obtain different views, we extract 256 LBP, 100 PyramidHOG (PHOG), 512 GIST, 32 Gabor textures, 200 SURF and 200 SIFT features.

The Digit dataset contains 2,000 data points for 0 to 9 ten-digit classes, and each class has 200 data points [148]. Six published features can be used for multi-view clustering: 76 Fourier coefficients of the character shapes (FOU), 216 profile correlations (FAC), 64 Karhunen-love coefficients (KAR), 240 pixel averages in 2 × 3 windows (PIX), 47 Zernike moment (ZER) and 6 morphological (MOR) features.

The WebKB dataset is a subset of web documents from four universities [149]. This dataset consists of 1051 pages, which are classified into 2 classes: 230 Course pages and 821 Non-Course pages. Each page has two views: Fulltext view contains 2949 features representing the textual content on the web pages, and Inlinks view consists of 334 features recording that the anchor text on the hyperlinks pointing to the pages.

3.6.2 Experimental Setup

In order to evaluate the performance of RMVCSC, we compared RMVCSC with several state-of-the-arts approaches, which includes robust multi-view *k*-means clustering (RMKMC) [152], pair-wised co-regularized multi-modal spectral clustering (PC-SPC) [153], centroid co-regularized multi-modal spectral clustering (CC-SPC) [154], multi-view subspace clustering (MVSC) [134], diversity induced multi-view subspace clustering (DiMSC) [155], and auto-weighted multi-view subspace clustering (RAMSC) [116].

RMKMC: RMKMC obtains common cluster indicators across multiple views by minimising the linear combination of the relaxed *k*-means on each view with learned weight factors [152].

PC-SPC: PC-SPC enforces the corresponding point in different modality to have the same cluster membership by a pair-wise co-regularization term, which makes different views be the same as each other [153].

CC-SPC: Similar to PC-SPC, CC-SPC makes different views be the same as a common one based on a centroid-based co-regularization term [154].

MVSC: MVSC performs subspace clustering on individual modality, respectively and then unify them with a common indicator matrix [134].

DiMSC: DiMSC learns subspace representations and employs the Hilbert-Schmidt Independence Criterion to enhance complementary information [155].

RAMSC: RAMSC is an auto-weighted multi-view subspace clustering approach based on common subspace representation matrix [116].

The source codes of these reference approaches are downloaded from the Internet. The best performances of these approaches are achieved according to these reference papers' setting. All the experiments are written in Matlab R2017a and processed on an HP Elite Desk 800 workstation with Intel i7-4790 CPU and 16GB RAM.

We normalise each view of the multi-view data firstly. All values of these input data will be in the range [-1, 1] before clustering. The threshold δ is set to be the mean of the lengths of the shortest 1% of the edges in the edge set ε. The parameter μ is initially set to $\mu = 3r^2$, where r is the maximal edge length in the edge set ε. All experiments are repeated five times independently. We reported the mean and standard deviation of them as experimental results.

Three standard clustering evaluation metrics are utilised to measure the multi-view clustering performance, i.e., Clustering Accuracy (ACC), Normalized Mutual Information (NMI) and Purity.

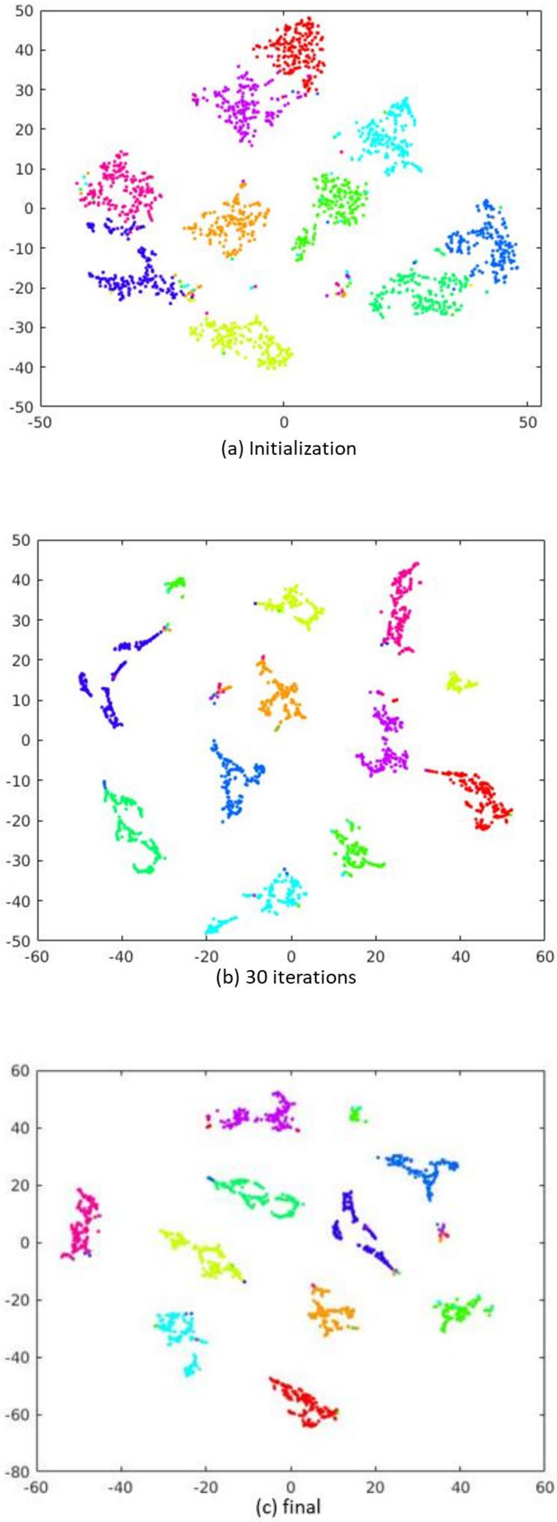
3.6.3 Experimental Results

The experimental results on three datasets with three metrics are shown in Table 3.2, Table 3.3 and Table 3.4, respectively. The final representation produced by RMVCSC on the Digit dataset is shown in Figure 3.1. The convergence behaviours of RMVCSC on three datasets are illustrated in Figure 3.2.

In terms of clustering accuracy, we have the following conclusions. From Table 3.2, Table 3.3 and Table 3.4, RMVCSC outperforms the reference approaches on all benchmark datasets. According to three different evaluation metrics: ACC, NMI, and Purity, our proposed approach can achieve a better or at least comparable performance.

As shown in Table 3.2, Table 3.3 and Table 3.4, the previous multi-view clustering approaches cannot always achieve better performances. This may be because the previous approaches characterise the structure of each view data separately and combine them with naïve addition operations, which makes the final clustering result affected by these inaccurate structures. RMVCSC can produce better results in most cases since our proposed approach assigns small weight factors to the inaccurate view and learns a common self-expressiveness matrix Z among different views.

Table 3.2, Table 3.3 and Table 3.4 also show the robustness of RMVCSC. Our proposed approach learns view weight factors without an extra parameter and uses the $\ell_{2,p}$-norm to eliminate the effects of inaccurate functions.

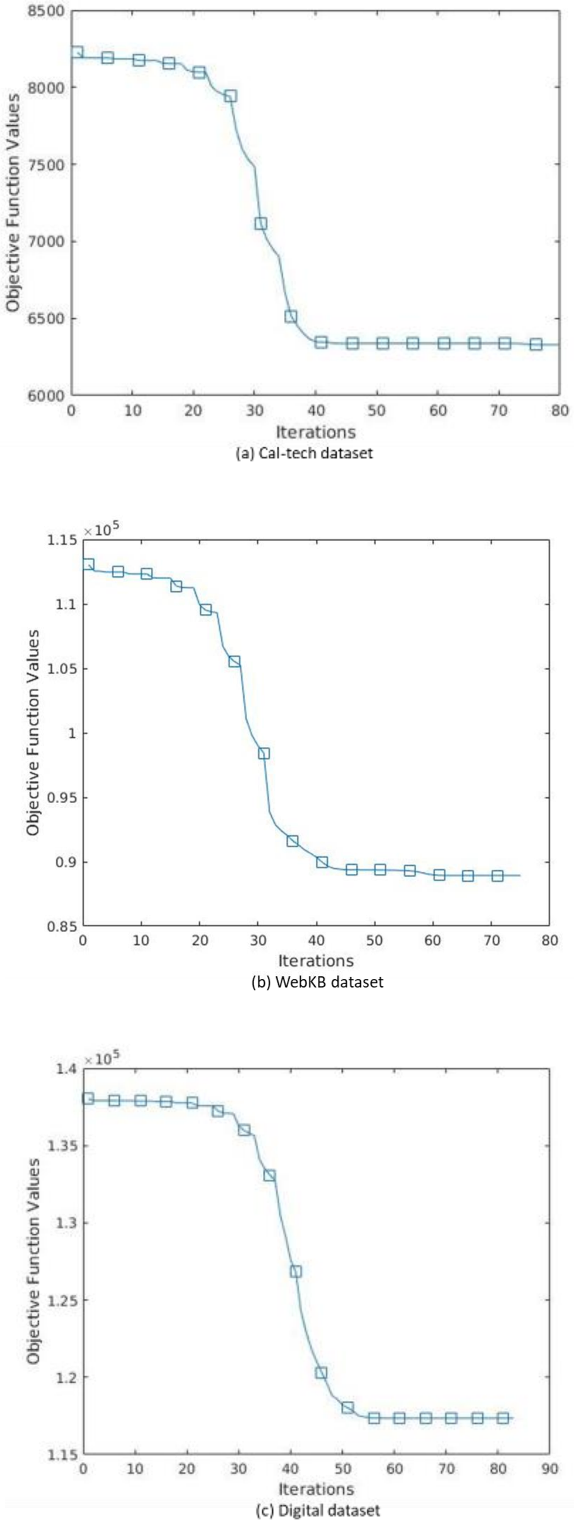


(a) Initialization

(b) 30 iterations

(c) final

Figure 3.1 - The common representations produced by RMVCSC on Digit dataset



(a) Cal-tech dataset

(b) WebKB dataset

(c) Digital dataset

Figure 3.2 - Convergence behaviours of RMVCSC on three datasets.

Table 3.2 - Clustering results of different approaches on the Cal-tech101-7 dataset.

Approach	ACC	NMI	Purity
RMKMC	0.6034 (±0.0680)	0.5488 (±0.0482)	0.6846 (±0.0541)
PC-SPC	0.6975 (±0.0499)	0.6547 (±0.0262)	0.7581 (±0.0288)
CC-SPC	0.7047 (±0.0654)	0.6879 (±0.0378)	0.7972 (±0.0389)
MVSC	0.6034 (±0.0309)	0.4766 (±0.0373)	0.6559 (±0.0314)
DiMSC	0.7312 (±0.0244)	0.6458 (±0.0179)	0.7698 (±0.0268)
RAMSC	0.7384 (±0.0082)	0.7276 (±0.0080)	0.8258 (±0.0115)
RMVCSC	0.7360 (±0.0078)	0.7631 (±0.0075)	0.8912 (±0.0112)

Table 3.3 - Clustering results of different approaches on the WebKB dataset.

Approach	ACC	NMI	Purity
RMKMC	0.8049 (±0.0000)	0.1592 (±0.0000)	0.8159 (±0.0000)
PC-SPC	0.7659 (±0.0000)	0.0991 (±0.0000)	0.7812 (±0.0000)
CC-SPC	0.5785 (±0.0000)	0.0019 (±0.0000)	0.7812 (±0.0000)
MVSC	0.7802 (±0.0000)	0.0041 (±0.0000)	0.7812 (±0.0000)
DiMSC	0.6147 (±0.0000)	0.0006 (±0.0000)	0.7812 (±0.0000)
RAMSC	0.9401 (±0.0000)	0.5689 (±0.0000)	0.9401(±0.0000)
RMVCSC	0.9402 (±0.0000)	0.5694 (±0.0000)	0.9420 (±0.0000)

Table 3.4 - Clustering results of different approaches on the Digit dataset.

Approach	ACC	NMI	Purity
RMKMC	0.7853 (±0.0800)	0.8125 (±0.0384)	0.8190 (±0.0614)
PC-SPC	0.8682 (±0.0604)	0.8267 (±0.0303)	0.8759 (±0.0500)
CC-SPC	0.8768 (±0.0605)	0.8234 (±0.0338)	0.8855 (±0.0471)
MVSC	0.8242 (±0.0686)	0.8399 (±0.0355)	0.8286 (±0.0664)
DiMSC	0.8400 (±0.0569)	0.8076 (±0.0347)	0.8465 (±0.0518)
RAMSC	0.9299 (±0.0439)	0.8864 (±0.0199)	0.9343 (±0.0333)
RMVCSC	0.9312 (±0.0245)	0.8867 (±0.0123)	0.9962 (±0.0323)

3.7 Conclusion

In this chapter, a novel Robust Multi-view Continuous Subspace Clustering (RMVCSC) approach is developed, which utilises a single continuous objective function to untangle heavily mixed clusters for multiple views data. In RMVCSC, the self-expressiveness is used to learn a common representation subspace across multiple views, in which the underlying cluster structure is revealed. The common representation subspace and the clustering result are simultaneously optimised in an alternating minimisation scheme by a robust redescending estimator. Thus, RMVCSC is not prone to stick into bad local minima even with outliers in data. As equipped with the recent developed robust continuous clustering algorithm, the developed RMVCSC is insensitive to initialisation, which means it does not require to pre-set the number of clusters. Due to these advantages, RMVCSC can achieve higher clustering accuracy across multiple views and is more robust for utilising.

A detailed optimisation process is discussed in Section 3.4. The convergence of the proposed approach is proved rigorously in Section 3.5. Compared with several very recently approaches, RMVCSC has more accurate clustering performance without pre-setting the number of clusters.

One limitation of this research is worth to mention. Although RMVCSC released the requirement of pre-setting the number of clusters, it still needs to pre-set the upper bound of the connections in the *m*-*k*NN graphs. A proper upper bound of connections can help the model to converge faster and lead to a better clustering result. However, it has much less effect than the number of clusters, as the pre-set number of clusters has a high potential to ruin the underlying clustering structure and is sensitive to the outliers in data.

As RMVCSC has great advantages in untangling heavily mixed clusters for multiple-view data, it can be widely extended into the active learning area. A detailed discussion about this can be found in Section 5.2 Future works.

Chapter 4 Polyphonic Sound Event Detection

This chapter presents the third developed approach, which fulfils the third research objective.

A smart environment is one of the application scenarios of the Internet of Things (IoT). In order to provide a ubiquitous smart environment for humans, a variety of technologies are developed. In a smart environment system, sound event detection is one of the fundamental technologies, which can automatically sense sound changes in the environment and detect sound events that cause the changes. In this chapter, we propose the use of Relational Recurrent Neural Network (RRNN) for polyphonic sound event detection, called RRNN-SED, which utilised the strength of RRNN in long-term temporal context extraction and relational reasoning across a polyphonic sound signal. Different from previous sound event detection approaches, which rely heavily on convolutional neural networks or recurrent neural networks, the proposed RRNN-SED can solve long-lasting and overlapping problems in polyphonic sound event detection. Specifically, since the historical information memorised inside RRNNs is capable of interacting with each other across a polyphonic sound signal, the proposed RRNN-SED is effective and efficient in extracting temporal context information and reasoning the unique relational characteristic of the target sound events. Experimental results on two public datasets show that the proposed RRNN-SED achieved better sound event detection results than other approaches, in terms of segment-based *F*-score and segment-based error rate.

This chapter is organised as follows. Section 4.1 introduces the motivation of the developed approach. Section 4.2 presents a literature review for sound event detection approaches. Section 4.3 firstly introduces the architecture of Relational Recurrent Neural Networks (RRNN), then a detailed description of the developed approach is presented in Section 4.3.2. Section 4.4 presents the evaluation framework and discusses the experimental results.

4.1 Introduction

In recent years, the Internet of Things (IoT) has received much attention and increasingly affects how we live [156][157][158][159]. Based on the interactions with multiple sensor devices, mobile computing devices, and advanced communications technologies, IoT provides smart and ubiquitous services for humans [159][160]. A smart environment is one of the application scenarios of IoT. The primary goal of the smart environment is to sense changes in the environment and to automatically adapt and act based on the changes, especially sensing and responding to human activities[156].

Among all the technologies used in a smart environment (e.g., infra-red sensors, contact doors and video cameras), sound event detection plays an important role. Specifically, since our daily life is filled with a rich variety of environmental sound events, such as dog barks, footsteps, baby crying and thunder, an effective sound event detection approach is beneficial to sense changes in the environment [161][162][163]. For example, if the sound of thunder is detected, it means that it is raining outside; if the sound of footsteps is detected, it means that someone is moving. Due to sounds are physically intangible, sound event detection does not specify the operation that produces a sound event to be physically at a particular place [164][165][166].

Sound event detection is to identify sound events in a continuous sound signal, which can be broadly classified to monophonic sound event detection and polyphonic sound event detection [174][176]. Monophonic sound event detection is to recognise the most dominant of sound events in a sound segment, whereas polyphonic sound event detection recognises all sound events (not only the most dominant sound event) in a sound segment [168][169][170][175]. In realistic scenarios, multiple sound events are very likely to overlap in time. For example, in Figure 4.1, car horn, footsteps, dog barking, and speech are interwoven in an urban sound segment. Thus, polyphonic sound event detection is more suitable for smart environment applications.

Moreover, since sound events occur in unstructured environments, there are a large amount of environmental noise and overlapping sounds existing in sound segments. In addition, the acoustic characteristics of realistic sound events are variable. For example, although the barks of Labrador and Samoyed are different, they all belong to the sound event named dog bark. The complexity of real-life environments reflects the great challenges of polyphonic sound event detection.

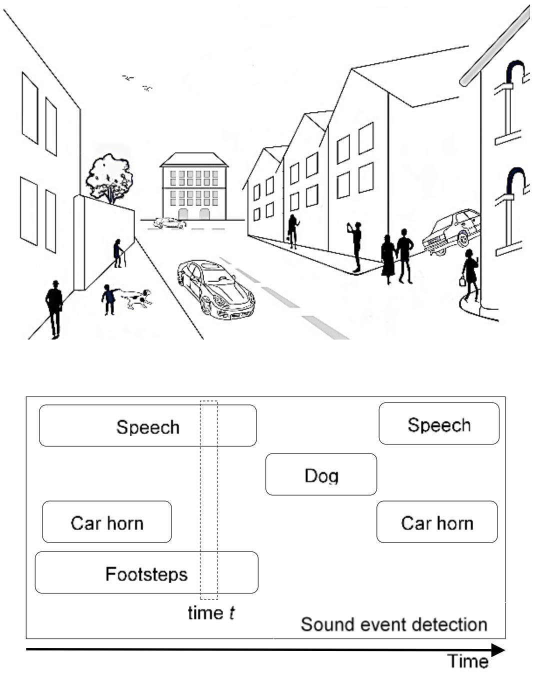


Figure 4.1 - Sound events in a real-life scenario can occur in isolation or overlap.

Due to the complexity of living environments, conventional classifiers (e.g., support vector machines), which are suitable for monophonic sound event detection, are not successful in polyphonic sound event detection [161]. Specifically, when monophonic sound event detection approaches are applied to polyphonic data, only one prominent sound event is detected. This will result in a loss of information in realistic environments [162]. Previous work for polyphonic sound event detection has chosen Mel Frequency Cepstrum Coefficient (MFCC) to characterise sound segments and uses Gaussian Mixture Model-Hidden Markov Models (GMM-HMMs) as classifiers with consecutive passes of the Viterbi algorithm [169][170][172][174][176].

Recently, Deep Neural Networks (DNNs, such as Convolutional Neural Networks (CNNs) and Recurrent Neural Networks (RNNs)) have dramatically improved the accuracy when solving complex problems (e.g., image recognition, speech recognition, and machine translation) [27][31]. Thus, numerous DNN-based approaches that are considered as the cutting-edge approaches have been developed for the sound event detection. CNN-based sound event detection approaches [177][178][179] have the ability to learn both time and frequency invariances using convolutional filters. However, this kind of approaches has difficulties in modelling long-lasting sound events (e.g., raining) [176]. RNN-based sound event detection approaches [180][181][182][183] have the ability to learn long-term context information by integrating historical information in memory units. However, standard RNNs cannot easily capture the invariance in the frequency domain.

In order to combine the strengths of both RNNs and CNNs, Convolutional Recurrent Neural Networks (CRNNs), which implements convolutional layers followed by recurrent layers, is developed [184][185]. However, all information in CRNNs is packed into a common hidden memory vector, in which all the historical information is mixed together. This results in the lack of the ability to solve the two critical issues in polyphonic sound event detection: (i) long-lasting sound event detection, and (ii) overlapping sound event detection. These two problems demand a network to have the abilities to capture the long-term temporal context information and to reason the unique relational characteristic of the reference sound events.

More recently, a new memory-based neural network called Relational Recurrent Neural Network (RRNN) is proposed in [171]. In RRNN, a new memory module called Relational Memory Core (RMC) is developed to memorise information for a long-term and perform complex relational reasoning with the information stored inside the memory module. Based on this novel memory module, the history information in RMC will be memorised for a long time and interacted with other context information in the history [171]. Thus, RRNN can be used to solve the aforementioned two issues in polyphonic sound event detection.

In this chapter, we propose the use of RRNN for polyphonic sound event detection in real-life environments, called RRNN-SED. Since RRNN has the capacity to allow memories to interact across a sound segment, it is more capable of extracting temporal context information in the segment. The proposed RRNN-SED can improve the detection accuracy of complex and varied sound events in real-life environments. The proposed RRNN-SED is evaluated on two datasets, and the experimental results show that the proposed RRNN-SED outperforms previous sound event detection approaches in terms of segment-based *F*-score and segment-based error rate.

The rest of this chapter is organized as follows. Section 4.2 presents a literature review for sound event detection approaches. Section 4.3 introduces the architecture of RRNN, which is then utilized for polyphonic sound event detection. Section 4.4 presents the evaluation framework used to measure the performance of different DNN architectures and discusses the experimental results. Section 4.5 presents conclusions.

4.2 Related Work

Polyphonic sound event detection involves detecting sound events in a sound segment and assigning them to the known labels [184]. There are no fixed patterns of sound events in reality. Different sound events may occur independently or overlap with other sound events. A polyphonic sound event detection approach aims to correctly and simultaneously detect all the overlapping events. Although identifying isolated sound events can be done with an appreciable accuracy in recent years, detecting a series of overlapping sound events is still a challenging task. An ideal sound event detection approach should be able to deal with such overlapping sound events.

In the past decades, different approaches have been developed for polyphonic sound event detection. Early approaches relied on the combination of standard features (e.g., MFCC) and standard machine learning algorithms, such as support vector machines or GMM-HMMs. Annamaria *et al.* [172] presented an acoustic event detection approach based on HMMs. The size and topology of the proposed approach are chosen based on a study of isolated events recognition. Toni *et al.* [173] proposed a GMM-based sound event detection approach, which utilises both context-dependent acoustic models and count-based event priors to improve the detection accuracy. By using the coupled matrix factorisation of spectral representations and class activity annotations, Annamaria *et al.* [174] developed a sound event detection approach to detect sound events in real-life recordings. However, these approaches are suitable for monophonic sound event detection but not for polyphonic sound event detection.

Recently, since DNN has achieved great success in image and speech domain, numerous deep learning approaches have been proposed for sound event detection. In [175], a multi-label DNN is proposed for detection of temporally overlapping sound events in realistic environments. However, there are two limitations exist in this proposed DNN structure: (i) the lack of ability to extract time and frequency invariance, (ii) the lack of ability to memorise long-term context information [176]. Thus, two types of powerful neural networks (CNNs and RNNs) are introduced to sound event detection.

Multiple convolutional filters are utilised by the CNN structure so that they can learn both time and frequency invariance. Il-Young *et al.* [177] used both short- and long-term audio signals simultaneously as the input of CNNs to maximise detection performance. Yukun *et al.* [178] calculated Mel-band energy for the merged multi-channel audio signals and train the sound event detection model using a CNN. A CNN-based sound event detection approach coupled with two loss functions (the weighted loss function and multi-task loss function) is proposed in [179], where the weighted loss function is used to solve the problem of imbalanced data (in background/foreground classification) and the multi-task loss function is designed to model class distribution and temporal structures of sound events simultaneously. However, CNN structures have difficulties in modelling long-lasting sound events (e.g., raining).

On the contrary, by integrating early time information in memory units, RNNs can learn long-term temporal context information from the input signals. Sharath *et al.* [180] proposed the use of spatial and harmonic features in combination with Long Short Term Memory (LSTM) neural network for automatic sound event detection. Toan *et al.* [181] presented a sound event detection approach for real-life audio using log Mel-band energy features and bi-directional RNN structure. Giambattista *et al.* [182] presented a Bi-directional Long Short-Term Memory (BLSTM) based approach for the polyphonic sound event detection. Since real-life sound segments consist of multiple sound events, the acoustic features will be mapped to binary activity indicators of each sound event class by using a single multilabel BLSTM. A polyphonic sound event detection approach for stereo (multichannel) audio signals is proposed in [183], which uses log Mel-band energy features with LSTM. Besides the left channel and right channel, it also constructs two more channels (mean channel and different channel) for feature extraction. Then the detection is based on the fusion results of these channels. However, standard RNNs do not easily capture the invariance in the frequency domain.

In recent years, a combination of CNNs and RNNs, named CRNN and integrate the strengths of both RNNs and CNNs, has shown to outperform previous sound event detection approaches. Giambattista *et al.* [184] proposed to use low-level spatial features extracted from stereo audios and use Gated Recurrent Units (GRUs) for CRNN-based sound event detection. Sharath *et al.* [185] proposed to use low-level spatial features extracted from multichannel audio for CRNN-based sound event detection approach. However, all information in the previous CNN or RNN structures is packed into a common hidden memory vector, which potentially makes compartmentalisation and relational reasoning more difficult.

More recently, a memory-based neural network RRNN is proposed in [171]. RRNN employs multi-head attention for memory interaction so that the network has the ability to perform complex relational reasoning with the information 'remembered'. Based on its novel memory structure and relational reasoning ability across sequence information, RRNN can be utilized for polyphonic sound event detection. For long-lasting or overlapping sound events, RRNN can learn the unique characteristics of different environmental sound events and distinguish them by using its RMC structure [171].

This chapter proposed a novel RRNN-based approach for polyphonic environmental sound event detection. Since RRNN has the capacity to allow memories to interact across a sound segment, it is more capable of extracting temporal context information in the sound segments. The proposed RRNN-SED can improve the detection accuracy of polyphonic sound events in real-life environments.

4.3 Models

In this section, the Relational Recurrent Neural Network (RRNN) is introduced first. Then, the proposed RRNN-SED approach is presented in detail.

4.3.1 Relational Recurrent Neural Networks

In order to enable the network to perform complex relational reasoning with the information 'remembered', RRNN employs multi-head attention for memory interaction. As shown in Figure 4.2, the Relational Memory Core (RMC) in RRNN modifies the structure of memory cell in a standard Long Short Term Memory network (LSTM) [171].

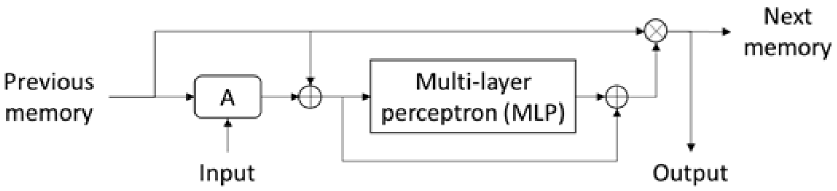


Figure 4.2 - Relational Memory Core

Attention mechanisms have become an integral part of sequence modelling tasks, which is to select the most pertinent piece of information rather than all available information to the current state [186]. Thus, the attention mechanism can be described as the process of mapping a query and a set of key-value pairs to an output, where the query, key, value, and output are all vectors [186]. The output is calculated as a weighted sum of the value, where the weight assigned to each value is calculated by querying a compatibility function with the corresponding key.

The multi-head dot product attention (MHDPA) is a self-attention algorithm proposed in [186]. Since the attention function is calculated on a set of queries simultaneously, all queries, keys, and values will be stored respectively in matrices *Q*, *K*, and *V*. The dot products of the query (i.e., dot-product attention) is calculated with all keys *K*, the dimensionality of the key vectors d_k, and a softmax function, which is used to obtain the weights on the values. Equivalently,

$$A(Q,K,V) = softmax\left(\frac{QK^T}{\sqrt{d_k}}\right)V \tag{4.1}$$

Using a linear projection, all queries, keys, and values can be constructed with $Q = MW_q$, $K = MW_k$, and $V = MW_v$, where W_* denotes the weight matrix, and M is a matrix of memories, which is randomly initialized. The dot-product attention can be calculated with the following equation:

$$A_\theta(M) = softmax\left(\frac{MW_q(MW_k)^T}{\sqrt{d_k}}\right)MW_v, \text{where } \theta = \left(W_q, W_k, W_v\right) \tag{4.2}$$

The output of $A_\theta(M)$, which can be presented as M', is a matrix with the same dimensionality as M. M' can be considered as the update of M, where each element m'_e in M' is consisted of information from the matrix of memories M. Thus, in one step of the attention, each memory is updated with information from other memories, and learning how to shuttle information from memory to memory via the parameters W_q, W_k and W_v.

RRNN is based on the multi-head attention mechanism, where there are l heads exist in the input, and the attention mechanism is processed on each head. In other words, the algorithm will generate l sets of queries, keys, and values for l heads, and then calculates a linear projection from the original memory for each head using unique parameters. For example, M is a $N_{height} \times N_{width}$ memory matrix, and the number of the attention head is 2, hence $M_1' = A_\theta(M)$ and $M_2' = A_\vartheta(M)$, where M_1' and M_2' are $N_{height}/2 \times N_{width}$ matrices, θ and ϑ denote unique parameters for linear projections to process the queries, keys, and values. $M' = [M_1'; M_2']$ is a row-wise concatenation for M'. In general, the multi-head attention allows memory to share different information and focus on the same information from different perspectives (i.e., heads).

Suppose there is a temporal dimension with new observations at each time step *t*. Since M and M' have the same dimensionality, the recurrence will randomly initialise M, and then update it with M' at each time step. Thus, Equation (4.2) can be rewritten as:

$$M' = softmax\left(\frac{MW_q([M;\ x]W_k)^T}{\sqrt{d_k}}\right)[M; x]W_v \tag{4.3}$$

In RRNN, the memory matrix M is considered as a matrix of cell states C for a standard LSTM. Specifically, the operations on each $m_{e,t}$ in RRNN replace the operations on each $c_{e,t}$ in a standard LSTM. Equivalently,

$$s_{e,t} = \left(h_{e,t-1}, m_{e,t-1}\right) \tag{4.4}$$

$$f_{e,t} = \sigma\left(W_f x_t + U_f h_{e,t-1} + b_f\right) \tag{4.5}$$

$$i_{e,t} = \sigma\left(W_i x_t + U_i h_{e,t-1} + b_i\right) \tag{4.6}$$

$$o_{e,t} = \sigma\left(W_o x_t + U_o h_{e,t-1} + b_o\right) \tag{4.7}$$

$$m_{e,t} = f_{e,t} m_{e,t-1} + i_{e,t} g_\varphi\left(m'_{e,t}\right) \tag{4.8}$$

$$h_{e,t} = o_{e,t} * tanh\left(m_{e,t}\right) \tag{4.9}$$

$$s_{e,t+1} = \left(m_{e,t}, h_{e,t}\right) \tag{4.10}$$

where $m_{e,t}$ is the e^{th} row in M at time step t, and g_φ denotes a post-attention processor (i.e., the output of Equation (4.3)). Since the parameters W_*, U_* and φ are shared for each m_e, the modifications of the number of memories do not affect the number of parameters. In other words, it just modifies the number of memories (i.e., the total number of elements in the memory matrix M) and the size of each memory (i.e., the dimensionality of m_e). Comparing with the structure of LSTM introduced in Section 2.2.2, the main change in RRNN is that it replaced the candidate value c in LSTM with the new memory cell states m_e.

4.3.2 Proposed Approach

The proposed RRNN-SED approach follows the state-of-the-art CRNN architecture in the sound event detection area. The proposed RRNN-SED consists of four modules: (i) a time-frequency representation module to convert a sound segment into Mel-spectrograms; (ii) a CNNs module to extract the time and frequency invariant features; (iii) a RRNNs module to extract long-term dependency and overlapping sound event information across a sound segment; (iv) a prediction module to estimate the probabilities of each sound event and output the final prediction by binarizing these probabilities over a constant threshold. Figure 4.3 illustrates the framework of proposed RRNN-SED approach.

In the first module, a raw stereo sound segment is firstly split into frames. Then, Mel-filterbanks are applied to these frames to generate the Mel-spectrogram for each sound channel. After this, an input sound segment is represented into a set of $MF \times T \times CH$ Mel-spectrograms, where MF is the number of Mel-filterbanks, T is the number of frames of the sound segment, CH is the number of sound channels.

The Mel-spectrograms generated by the first module are then fed into the CNNs module to extract the time and frequency invariant features. Each CNN layer utilises a certain number of convolutional filters to generate the feature maps. Then, max pooling is used to reduce the dimensionality of the data and provide time and frequency invariance. To preserve the time dimension, the max-pooling operation is calculated with zero-padding along the frequency axis. The output of the CNNs module from the last CNN layer is a $F \times MF' \times T$ tensor, where F is the number of convolutional filters in the last CNN layer, MF' is the number of Mel-filterbanks remaining after the max pooling operation in the last CNN layer. This output tensor is then reshaped into a $(F \cdot MF')$ $\times T$ feature sequence by concatenating along the frequency axis to produce the final output.

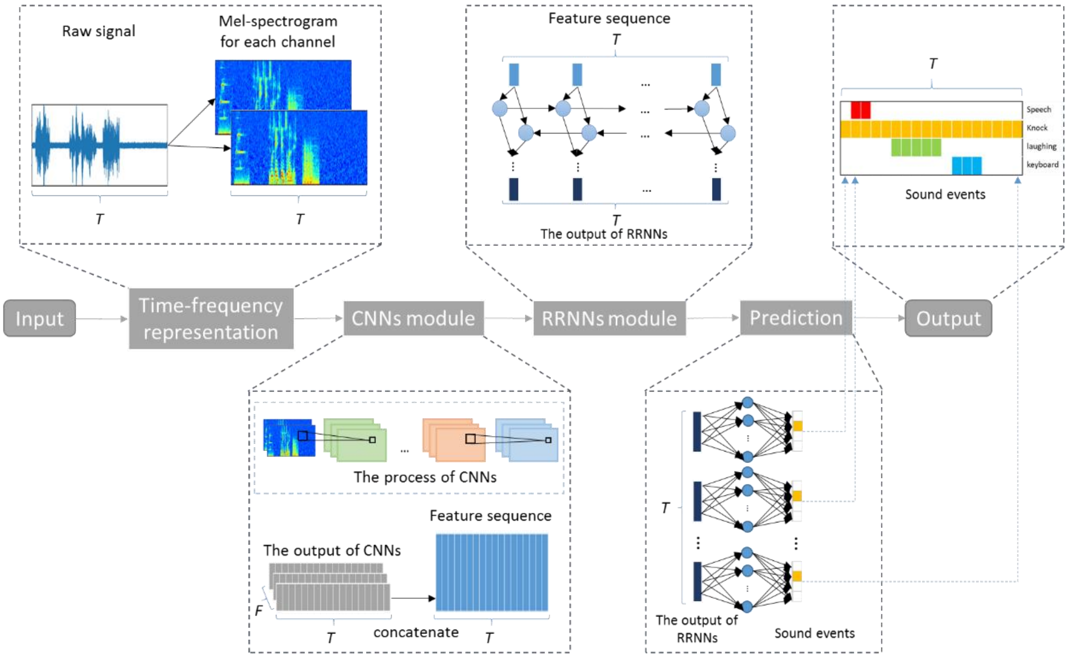


Figure 4.3 - Relational recurrent neural networks for polyphonic sound event detection

The concatenated $(F \cdot MF')$ $\times T$ tensor is then fed into the RRNNs module as a sequence of features for each frame to extract the long-term context information for sound events. The bi-directional technology is applied to extract the context information along both forward and backward time dimension. The output of the RRNNs module from the last RRNN layer is an $R \times T$ sequence, where R is the size of the output in each frame.

The prediction module consists of a fully-connected time-distributed layer, which takes the output sequence from the RRNNs module as input. At each frame, N_l sigmoid units are applied to estimate the probabilities for each sound event, where N_l is the total number of reference labels in the dataset. These probabilities are then binarised by a constant threshold to predict whether a reference sound event happened at this frame or not. A binary cross-entropy loss function is applied at each frame for each reference sound event to measure the predictions. The output of the prediction module is an $N_l \times T$ sequence with binary values. For a sound event, if it happened at the frame, it will be denoted as “1”; otherwise, it will be denoted as “0”.

The prediction results are evaluated by two widely used metrics for polyphonic sound event detection, the segment-based *F*-score and segment-based error rate, which will be introduced in detail in Section 4.4.2 Evaluation Metrics. Batch normalisation and drop out technologies are also applied after every layer of the proposed RRNN-SED approach.

The configuration used in the proposed network has several points of similarity with the networks presented in [162][184], as all of them are based on the CRNN architecture. The main difference is that instead of GRU or LSTM, RRNNs are used to improve the relational reasoning ability of the proposed network, which significantly improved the performance of sound event detection for complex and varied sound events in terms of segment-based *F*-score and segment-based error rate.

4.4 Evaluation

In order to test the proposed approach, a series of experiments have been done on two widely used polyphonic sound event detection datasets: TUT Sound Events 2016 dataset (TUT-SED 2016) [187] and TUT Sound Events 2017 dataset (TUT-SED 2017) [187]. The experimental results are evaluated by two widely used metrics: segment-based *F*-score [188] and segment-based error rate [189].

4.4.1 Datasets

The proposed approach is tested on two polyphonic sound event detection datasets: Tampere University of Technology Sound Events 2016 dataset (TUT-SED 2016) [187] and Tampere University of Technology Sound Events 2017 dataset (TUT-SED 2017) [187]. Both datasets are used as the competition datasets in an IEEE Audio and Acoustic Signal Processing Challenge event, named Detection and Classification of Acoustic Scenes and Events (DCASE) challenges [190].

The TUT-SED 2016 dataset is captured in real-life environments, which mean the number of overlapping sound events at each time is uncontrolled, neither in training nor in test recordings. This dataset consists of recordings from two acoustic scenes: home (indoor) and residential area (outdoor). Each recording was captured in a different location (e.g. different homes and different streets) with 3-5 minutes long, 44.1 kHz sampling rate and 24-bit resolution. Each recording is annotated with 18 different sound events (e.g. object impact, people walking and birds singing) together with the start time and end time of each event.

The TUT-SED 2016 dataset is officially split into two subsets: development dataset and evaluation dataset. The four-fold cross-validation setup published along with the dataset is used in the experiments. 25% of the training recordings are assigned for validation in the training stage of the proposed approach. Since the proposed approach discards the information about the scene, we train a single model for both scenes, rather than train two separate models for each scene.

The TUT-SED 2017 dataset is also captured in real-life environments, but only in the street acoustic scenes. Each scene contains various levels of traffic and human activity. The recordings were captured in different streets with 3-5 minutes long, 44.1 kHz sampling rate and 24-bit resolution. Each recording is annotated with six sound events (e.g. car, children and people walking) together with the start time and end time of each event. The activities annotated with the same sound event are variable. For example, “car passing by”, “car engine running” and “car idling” are all annotated with “car” label, which means the acoustic characteristics of the same annotated sound events may vary. An official five-fold cross-validation setup is also published along with the TUT-SED 2017 dataset, which is used in the experiments.

4.4.2 Evaluation Metrics

In this section, two segment-based evaluation metrics are used to evaluate the performance of experimental results: segment-based F-score [188] and segment-based error rate [189]. Both metrics use segments of one-second length to compare the system output with the ground truth. They are also the official metrics used in the DCASE challenges.

The segment-based F-score is used as the primary evaluation metric. For each segment in the test dataset, Precision (P) and Recall (R) are calculated from the accumulated intermediate statistics (i.e., the number of true positive (*TP*), false positive (*FP*) and false-negative entries (*FN*)). In each sound segment k, *TP* presents the events indicated as active by both the ground truth and the output, *FP* presents the events indicated as active by the output but inactive by the ground truth, and *FN* presents the events indicated as inactive by the output but active by the ground truth. Thus, F-score can be formulated as:

$$F = \frac{2 \cdot P \cdot R}{P + R}, \qquad (4.11)$$

$$where\, P = \frac{\sum TP(k)}{\sum TP(k) + \sum FP(k)}, \qquad R = \frac{\sum TP(k)}{\sum TP(k) + \sum FN(k)}$$

The other evaluation metric is segment-based error rate (*ER*), in which four intermediate statistics (i.e., the number of substitutions (*S*), insertions (*I*), deletions (*D*) and reference events (*N*)) are calculated per segment. In each sound segment k, a substitution (S) is the system output indicating as activing a wrong label event, which means the system did not detect the correct event (false negative for the correct class) but detected something (false positive for another class). Insertions (I) are the false-positive events after subtracting the substitutions. Deletions (D) are the false-negative events after subtracting the substitutions, and referent events (N) are the total number of events in the ground truth. Equivalently,

$$ER = \frac{\sum S(k) + \sum I(k) + \sum D(k)}{\sum N(k)} \tag{4.12}$$

A more detailed explanation of the segment-based F-score and the segment-based error rate in multi-label setting can be found in [188].

4.4.3 Experiments

The experimental setups and the evaluation results are presented in this section. All the experiments are following the official cross-validation setup published along with the dataset. All the reported results are calculated on the test datasets. The experiments are executed on a deep learning workstation with four GTX 1080 Ti GPUs, 128GB RAM, and Intel Core i9 CPU. The proposed RRNN-SED approach is implemented in Python with the DeepMind Sonnet library [191], which is a deep learning library built on top of TensorFlow [192].

All the sound segments in both TUT-SED 2016 and TUT-SED 2017 datasets are preprocessed with the same procedure in the time-frequency representation module of the proposed RRNN-SED approach. The raw sound recordings are firstly split by a 50% overlapping window. Each window contains 2048 samples (approximate 46 milliseconds per frame with 44.1 kHz sample-rate). Then, the sound recordings are split into sound segments with 256 frames each as the training samples. After that, standard 40 Mel-filterbanks are applied on each frame to calculate log Mel-band energy and generate the Mel-spectrogram as the acoustic features. At the end of the preprocessing, the sound segments are represented as a set of Mel-spectrograms from each sound channel.

As introduced in Section 4.3.2, the CNNs module will take the Mel-spectrograms as input to extract the time and frequency invariant features. These features will be fed into the RRNNs module to extract temporal context information and to reason the unique relational characteristic of the target sound event. The prediction module will take the output sequence from the RRNNs module to predict which sound events happened in each frame.

The proposed RRNN-SED is trained with Adam [193] optimiser and binary cross-entropy loss function. Batch normalisation [194] and drop out [195] technologies are also applied in the proposed RRNN-SED approach to improving the generalisation ability. Training is stopped if the value of loss function does not decrease for 100 epochs. The binary threshold is set to 0.5 during the testing.

Table 4.1 - The configurations of RRNN-SED for each dataset

	TUT-SED 2016		TUT-SED 2017	
	CNN	RRNN	CNN	RRNN
CNN layers	3	-	3	-
pool size	(5,2,2)	-	(5,2,2)	-
RRNN layers	-	2	-	2
feature maps/ hidden units	96	32	128	32
memory slots	-	2	-	4
head size	-	16	-	16
number of heads	-	2	-	2
frame size (ms)	46		46	
frame overlap	50%		50%	
Mel-filterbanks	40		40	
sequence length (frames)	128		256	
batch size	64		128	
sigmoid layer	32		32	
dropout rate	0.5		0.5	

Table 4.1 summarises the configurations of the proposed RRNN-SED approach for each dataset. The inputs are stereo sound segments from datasets, which contains two sound channels. The raw sound segments are firstly split into 46 milliseconds long (2048 sample points of a 44100Hz sample-rate recording) frames with 50% overlap. Then, the acoustic features are calculated by using

standard 40 Mel-filterbanks to generate the Mel-spectrogram for each sound channel. (3×3) kernels are used for the CNN layers. The max-pooling of each CNN layer is five, two, and two, respectively. After that, two bi-directional RRNNs are utilised for extracting temporal information through the sound segments and a fully-connected time-distributed layer is used to produce final detection results.

Table 4.2 - Experimental results on TUT-SED 2016 dataset

	TUT-SED 2016	
	segment-based *F*-score	segment-based ER
Official baseline	0.343	0.8773
Adavanne *et al.* [169]	0.478	0.8051
Lai *et al.* [198]	0.345	0.9287
Vu and Wang [200]	0.419	0.9124
RRNN-SED	0.475	0.7459

The proposed RRNN-SED approach is compared with the official baselines and the top three competition results from the corresponding DCASE challenge for each dataset. The official baselines are published together with the datasets, which is based on a multilayer perceptron architecture with two dense layers of 50 hidden units per layer. The details about the baselines can be found here [190]. The top three competition results are selected from the DCASE challenge 2016 and 2017 official evaluation servers, which report all the results during the competition. The technical reports about each selected approach can be found here [196][197].

Table 4.2 and Table 4.3 list the experimental results of the best performing (based on the validation data) of the proposed approach, and other reference approaches. As shown in the tables, considering both metrics, the proposed approach consistently outperforms other reference approaches on the two datasets in terms of the segment-based *F*-score and the segment-based error rate.

Table 4.3 - Experimental results on TUT-SED 2017 dataset

	TUT-SED 2017	
	segment-based *F*-score	segment-based ER
Official baseline	0.428	0.9358
Kroos and Plumbley [199]	0.449	0.8979
Adavanne and Virtanen [161]	0.417	0.7914
Dang *et al.* [200]	0.442	1.0318
RRNN-SED	0.461	0.7973

The experimental results shown in Table 4.2 and Table 4.3 indicate that the proposed RRNN-SED approach has a better performance on distinguishing overlapped sound events. When different sound events overlap with each other, the acoustic characteristics of them may heavily mix together within the raw sound signals. Thus, it is hard to untangle these acoustic characteristics of different sound events and detect them accurately. In addition, as the datasets are recorded in real-life environments, background noises and irrelevant sounds are also mixed inside the raw sound signals, which makes the polyphonic sound event detection problem more complicated.

The advantage of RRNN is that it enables historical information to interact with each other within the Relational Memory Core (RMC). Inside the RMC, the Multi-Head Dot Product Attention (MHDPA) mechanism is applied. Thus, during the interaction of historical information inside the RMC, the features that are more discriminatively reflecting the acoustic characteristics of reference sound events will be highlighted, whereas the irrelevant features will be ignored. This special designed RMC helps the proposed RRNN-SED gain better performance when facing overlapped sound events.

4.5 Conclusions

In this chapter, the RRNN-SED approach is developed, which utilises Relational Recurrent Neural Network (RRNN) for polyphonic sound event detection in real-life environments. Since RRNN employs multi-head self-attention inside Relational Memory Core (RMC) for memory interaction, it is more capable of performing complex relational reasoning with the information ‘remembered’. Because of its novel memory structure and relational reasoning ability across a sequence, RRNN is utilised for polyphonic sound event detection to achieve better results. Specifically, for long-lasting or overlapping sound events, RRNN can learn the unique acoustic characteristics of different sound events and distinguish them inside the relational memory core, in which the historical information can interact with each other with the self-attention mechanism. For the acoustic characteristics of sound events, the information that can distinguish a sound event from others will be highlighted by the self-attention mechanism, whereas other irrelevant information will be ignored. Thus, the proposed RRNN-SED can improve the detection accuracy of complex and varied sound events in real-life environments.

The developed RRNN-SED is evaluated on two public datasets recorded from real-life environments. The performance of RRNN-SED is compared with the approaches presented in the DCASE Challenge. The experimental results show that the proposed approach outperforms previous polyphonic sound event detection approaches in terms of segment-based error rate and segment-based *F*-score.

As the relational memory core is much complex than the original memory cell in Long Short Term Memory (LSTM) networks, RRNN-SED will consume more computational resources than LSTM, which may limit the deployment of RRNN-SED in real-world scenarios.

Since polyphonic sound event detection can provide important information about the changes in the real-world environment, the proposed RRNN-SED can be widely extended into many other tasks such as the video caption. A detailed discussion about RRNN-SED for video caption can be found in Section 5.2 Future works.

Chapter 5 Conclusions and Future Works

This final chapter firstly summarises each previous chapter and highlights the major contributions of this thesis. At the end of this chapter, future works are discussed for continuous research.

5.1 Research Overview and Summary

Machine learning is a blooming and fast-growing research area, which covers a vast of diverse sub-fields and real-world applications. Two novel approaches are developed to address two crucial theoretical issues in deep neural networks and clustering, which are the two most popular subfields in the machine learning area. In addition, another novel approach is developed for polyphonic sound event detection, which is one of the most important applications in the audio processing area. Each developed approach is explicitly presented in the corresponding chapter.

Chapter 2 presents the developed Large Margin Recurrent Neural Networks (LMRNNs), which fulfils the first research objective. Since RNNs is one of the most successful approaches for processing sequential data, most of the state-of-the-art audio processing models are based on RNNs. However, in the most common multi-class classification tasks, most of the current RNNs employ the cross-entropy loss function, which does not fully benefit from the information provided by the data labels. This is because the cross-entropy loss function only considers the target category without considering the competing categories during the training processes.

To solve this problem, LMRNNs is developed, which utilises a large margin discriminative principle as a heuristic term to improve the discriminative ability of original RNNs. A detailed explanation about the drawback of the cross-entropy loss function can be found in Section 2.3.1. Section 2.3.2 presents the proposed margin term. Section 2.3.3 mathematically discusses the behaviour of the proposed margin term. Section 2.4 tests the proposed LMRNNs with two widely used datasets and further discussed the behaviour of margin term in the experiments.

Chapter 3 presents the developed, Robust Multi-View Continuous Subspace Clustering (RMVCSC) approach, which fulfils the second research objective. Multi-view clustering is one of the most efficient ways to analyse multi-view data. Multi-view subspace clustering is the most promising approach to do so, as this approach can reveal the underlying cluster structure in multi-view data from a learned representation inside a common subspace. However, most existing multi-view subspace clustering algorithms are based on k-means or spectral clustering, which requires to manually set the number of clusters beforehand. This limited the further advancement of multi-view subspace clustering.

To solve this problem, a Robust Multi-View Continuous Subspace Clustering (RMVCSC) approach is developed. RMVCSC utilises the recently developed Robust Continuous Clustering (RCC), which does not need to know the number of clusters in advance and can efficiently achieve high accuracy even the data is in high-dimension. The proposed RMVCSC extends RCC into multi-view setting, which optimises a novel continuous objective in the simultaneously learned common representation subspace across multiple views. RMVCSC is optimized in an alternating minimisation scheme, in which the clustering result and the common representation subspace are simultaneously optimised. By using robust redescending estimators, the proposed RMVCSC is not prone to stick into bad local minima even with outliers in data. Section 3.3 introduces the proposed RMVCSC. Section 3.4 gives a detailed optimisation process. Section 3.5 analyses the convergence behaviour and Section 3.6 tests the proposed RMVCSC on three widely used multi-view datasets.

Chapter 4 presents the developed polyphonic Sound Event Detection approach based on the recent developed Relational Recurrent Neural Network (RRNN), named RRNN-SED. As reviewed in Section 4.2, the current state-of-the-art approaches for polyphonic Sound Event Detection are based on Convolutional Recurrent Neural Network (CRNN). However, one limitation of CRNN based approaches is that all the historical information is mixed together into the hidden state vector. This results in the disadvantage of reasoning the unique relational characteristic of overlapping sound events.

To solve this problem, RRNN-SED is developed, which exploit the strength of RRNN in complex relational reasoning with the information stored inside the memory module, called Relational Memory Core (RMC). The history information in RMC will be memorised for a long time and interacted with other context information in history. Thus, the proposed RRNN-SED is effective and efficient in extracting temporal context information and reasoning the unique relational characteristic of the target sound events. Section 4.3.1 introduces RRNN and explains the mechanism of RMC. Section 4.3.2 presents the proposed RRNN-SED approach in detail. Section 4.4.3 tests RRNN-SED on two competition datasets and evaluate RRNN-SED performance in terms of segment-based *F*-score and segment-based error rate.

To sum up, this thesis developed three novel approaches in machine learning and audio processing research areas. These novel approaches are (i) Large Margin Recurrent Neural Networks in Chapter 2; (ii) Robust Multi-View Continuous Subspace Clustering in Chapter 3; (iii) Relational Recurrent Neural Network based polyphonic sound event detection (RRNN-SED) approach in Chapter 4. All these three developed novel approaches have already been submitted to or published on top journals.

5.2 Future Works

Three novel approaches have been developed in machine learning and audio processing areas in this thesis. However, the potential of these developed approaches has not been fully explored. In this section, several research directions are discussed to extend the potential of these developed approaches.

The first research direction is Neural Architecture Search (NAS). NAS aims to automatically design the architecture of Deep Neural Networks, which is one of the critical aspects that affect DNN's performance [201]. Although DNNs are dominating many research areas, such as audio processing, video processing, and machine translation, most of the current DNN architectures are designed manually by human experts [202]. Such designing process is extremely error-prone and time-consuming [201]. NAS can automate this process and achieve comparable performance with human experts [203]. One of the key problems in NAS is how to accelerate the evaluation process of the potential architectures, as training a DNN is usually time-consuming [201][204][205]. As discussed in Chapter 2, the proposed large margin term in Large Margin Recurrent Neural Network can navigate the convergence process during training, which may be extended to NAS to help to accelerate the evaluation process.

The second research direction is Active Learning, in which the proposed Robust Multi-View Continuous Subspace Clustering (RMVCSC) approach can be extended. Active Learning is a special case of semi-supervised learning [4][206]. Different from other semi-supervised learning approaches, Active Learning aims to interactively select the most informative and representative samples for the users or experts to label [207]. This is a very efficient way to label a large amount of data while reducing the labelling cost [208]. Active Learning usually consists of two stages [209]. The first stage is to select the most informative and representative samples from the unlabelled dataset, which is usually done by clustering approaches [168][210]. Clustering approaches can group the unlabelled data into clusters based on given similarity criteria. The cluster centres are naturally the most representative samples of the corresponding cluster [211]. The second stage is to query the users or experts to label the selected data. Then, the data in the same cluster will automatically get the same label as the cluster centre. It is obvious that the clustering step is the most critical step in this process, in which the proposed RMVCSC can help to improve the clustering accuracy by comprehensively exploit information from multiple views of data.

The third research direction is the video caption. Video caption aims to generate natural language descriptions for video recordings [212], which can be widely used in video information retrieval [213] and video understanding [214]. Current research on video caption is mainly focusing on visual information while ignoring the audio information [215][216][217]. However, audios in a video recording is an important information source about what is happening in this video [218]. Sometimes, audios can provide additional information for Video Caption. For example, footsteps may mean a person is approaching, while this person may not necessarily appear in the visual stream. Thus, how to exploit the audio information is one of the key problems in video caption[219]. Sound event detection is one of the solutions. The proposed Relational Recurrent Neural Network based polyphonic Sound Event Detection (RRNN-SED) approach can help to detect all the overlapping sound events in a video, which will help to improve the generation of video captions.

In summary, the proposed approaches in this thesis can be further extended to many research directions, such as Active Learning, Neural Architecture Search, and Video Caption.

References

[1] Junbo Ma, Ruili Wang, Wanting Ji, Jiawei Zhao, Ming Zong, and Andrew Gilman. "Robust multi-view continuous subspace clustering." *Pattern Recognition Letters* (2018).

[2] Junbo Ma, Ruili Wang, Wanting Ji, Hao Zheng, En Zhu, and Jianping Yin. "Relational recurrent neural networks for polyphonic sound event detection." *Multimedia Tools and Applications* (2019): 1-19.

[3] Christopher M. Bishop, "*Pattern Recognition and Machine Learning*". *Springer*, (2006).

[4] Michael I. Jordan, and Tom M. Mitchell. "Machine learning: Trends, perspectives, and prospects." *Science* 349, no. 6245 (2015): 255-260.

[5] Jaime G. Carbonell, Ryszard S. Michalski, and Tom M. Mitchell. "An overview of machine learning." In *Machine Learning*, pp. 3-23. Morgan Kaufmann, 1983.

[6] Arthur L. Samuel. "Some studies in machine learning using the game of checkers. II—recent progress." In *Computer Games I*, pp. 366-400. Springer, New York, NY, 1988.

[7] Tom M. Mitchell, Jaime G. Carbonell, and Ryszard S. Michalski, eds. "Machine Learning: A Guide to Current Research". Vol. 12. *Springer Science & Business Media*, 1986.

[8] Tom M. Mitchell. "Machine Learning". *McGraw Hill*. (1997).

[9] Robert Hecht-Nielsen. "Theory of the backpropagation neural network." In *Neural Networks for Perception*, pp. 65-93. Academic Press, 1992.

[10] Theodoros Evgeniou, and Massimiliano Pontil. "Support vector machines: Theory and applications." In *Advanced Course on Artificial Intelligence*, pp. 249-257. Springer, Berlin, Heidelberg, 1999..

[11] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. "Deep learning". *MIT press*, 2016.

[12] Sotiris B. Kotsiantis, I. Zaharakis, and P. Pintelas. "Supervised machine learning: A review of classification techniques." *Emerging Artificial Intelligence Applications in Computer Engineering* 160 (2007): 3-24.

[13] Steven W. Knox. "Machine learning: a concise introduction". Vol. 285. *John Wiley & Sons*, 2018.

[14] Leslie Pack Kaelbling, Michael L. Littman, and Andrew W. Moore. "Reinforcement learning: A survey." *Journal of Artificial Intelligence Research* 4 (1996): 237-285.

[15] Irene Martín-Morató, Annamaria Mesaros, Toni Heittola, Tuomas Virtanen, Maximo Cobos, and Francesc J. Ferri. "Sound Event Envelope Estimation in Polyphonic Mixtures." In *2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 935-939. IEEE, 2019.

[16] Fabio Vesperini, Leonardo Gabrielli, Emanuele Principi, and Stefano Squartini. "Polyphonic Sound Event Detection by using Capsule Neural Networks." *IEEE Journal of Selected Topics in Signal Processing* (2019).

[17] Yun Wang, and Florian Metze. "Connectionist temporal localization for sound event detection with sequential labeling." In *2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 745-749. IEEE, 2019.

[18] Chung-Cheng Chiu, Tara N. Sainath, Yonghui Wu, Rohit Prabhavalkar, Patrick Nguyen, Zhifeng Chen, Anjuli Kannan *et al.* "State-of-the-art speech recognition with sequence-to-sequence models." In *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 4774-4778. IEEE, 2018.

[19] Stavros Petridis, Themos Stafylakis, Pingehuan Ma, Feipeng Cai, Georgios Tzimiropoulos, and Maja Pantic. "End-to-end audio visual speech recognition." In *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 6548-6552. IEEE, 2018.

[20] Chao Weng, Jia Cui, Guangsen Wang, Jun Wang, Chengzhu Yu, Dan Su, and Dong Yu. "Improving attention based sequence-tosequence models for end-to-end English conversational speech recognition." In *Interspeech 2018*. 2018.

[21] Gül Varol, Ivan Laptev, and Cordelia Schmid. "Long-term temporal convolutions for action recognition." *IEEE Transactions on Pattern Analysis and Machine Intelligence* 40, no. 6 (2018): 1510-1517.

[22] Hakan Bilen, Basura Fernando, Efstratios Gavves, and Andrea Vedaldi. "Action recognition with dynamic image networks." *IEEE Transactions on Pattern Analysis and Machine Intelligence* 40, no. 12 (2018): 2799-2813.

[23] Hossein Rahmani, Ajmal Mian, and Mubarak Shah. "Learning a deep model for human action recognition from novel viewpoints." *IEEE Transactions on Pattern Analysis and Machine Intelligence* 40, no. 3 (2018): 667-681.

[24] Heeyoul Choi, Kyunghyun Cho, and Yoshua Bengio. "Fine-grained attention mechanism for neural machine translation." *Neurocomputing* 284 (2018): 171-176.

[25] Di He, Yingce Xia, Tao Qin, Liwei Wang, Nenghai Yu, Tie-Yan Liu, and Wei-Ying Ma. "Dual learning for machine translation." In *Advances in Neural Information Processing Systems (NIPS)*, pp. 820-828. 2016.

[26] Jason Lee, Kyunghyun Cho, and Thomas Hofmann. "Fully character-level neural machine translation without explicit segmentation." *Transactions of the Association for Computational Linguistics* 5 (2017): 365-378.

[27] Jürgen Schmidhuber. "Deep learning in neural networks: An overview." *Neural Networks* 61 (2015): 85-117.

[28] Yan-Yan Song, and L. U. Ying. "Decision tree methods: applications for classification and prediction." *Shanghai archives of psychiatry* 27, no. 2 (2015): 130.

[29] Yury A. Malkov, and Dmitry A. Yashunin. "Efficient and robust approximate nearest neighbor search using hierarchical navigable small world graphs." *IEEE Transactions on Pattern Analysis and Machine Intelligence* (2018).

[30] Jingjing Tang, Yingjie Tian, Peng Zhang, and Xiaohui Liu. "Multiview privileged support vector machines." *IEEE Transactions on Neural Networks and Learning Systems* 29, no. 8 (2018): 3463-3477.

[31] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. "Deep learning." *Nature* 521, no. 7553 (2015): 436.

[32] Dmitrii Marin, Meng Tang, Ismail Ben Ayed, and Yuri Boykov. "Kernel clustering: density biases and solutions." *IEEE Transactions on Pattern Analysis and Machine Intelligence* 41, no. 1 (2019): 136-147.

[33] Ling Huang, Hong-Yang Chao, and Chang-Dong Wang. "Multi-view intact space clustering." *Pattern Recognition* 86 (2019): 344-353.

[34] Ioannis A. Maraziotis, Stavros Perantonis, Andrei Dragomir, and Dimitris Thanos. "K-Nets: Clustering through nearest neighbors networks." *Pattern Recognition* 88 (2019): 470-481.

[35] Yugen Yi, Jianzhong Wang, Wei Zhou, Yuming Fang, Jun Kong, and Yinghua Lu. "Joint Graph Optimization and Projection Learning for Dimensionality Reduction." *Pattern Recognition* (2019).

[36] Cem Örnek, and Elif Vural. "Nonlinear supervised dimensionality reduction via smooth regular embeddings." *Pattern Recognition* 87 (2019): 55-66.

[37] Mehrtash Harandi, Mathieu Salzmann, and Richard Hartley. "Dimensionality reduction on SPD manifolds: The emergence of geometry-aware methods." *IEEE Transactions on Pattern Analysis and Machine Intelligence* 40, no. 1 (2018): 48-62.

[38] Adriana Romero, Carlo Gatta, and Gustau Camps-Valls. "Unsupervised deep feature extraction for remote sensing image classification." *IEEE Transactions on Geoscience and Remote Sensing* 54, no. 3 (2016): 1349-1362.

[39] Patrick M. Sheridan, Chao Du, and Wei D. Lu. "Feature extraction using memristor networks." *IEEE Transactions on Neural Networks and Learning Systems* 27, no. 11 (2016): 2327-2336.

[40] Youness A. Ghassabeh, Frank Rudzicz, and Hamid Abrishami Moghaddam. "Fast incremental LDA feature extraction." *Pattern Recognition* 48, no. 6 (2015): 1999-2012.

[41] Aapo Hyvarinen, and Hiroshi Morioka. "Unsupervised feature extraction by time-contrastive learning and nonlinear ica." In *Advances in Neural Information Processing Systems (NIPS)*, pp. 3765-3773. 2016.

[42] Aaron van den Oord, and Oriol Vinyals. "Neural discrete representation learning." In *Advances in Neural Information Processing Systems (NIPS)*, pp. 6306-6315. 2017.

[43] Carl Doersch, Abhinav Gupta, and Alexei A. Efros. "Unsupervised visual representation learning by context prediction." In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pp. 1422-1430. 2015.

[44] Luan Q. Tran, Xi Yin, and Xiaoming Liu. "Representation learning by rotating your faces." *IEEE Transactions on Pattern Analysis and Machine Intelligence* (2018).

[45] Rui Xu, and I. I. Donald Wunsch. "Survey of Clustering Algorithms." *IEEE Transactions on Neural Networks* 16, no. 3 (2005): 645.

[46] Carl E. Rasmussen. "The infinite Gaussian mixture model." In *Advances in Neural Information Processing Systems (NIPS)*, pp. 554-560. 2000.

[47] Bing Jian, and Baba C. Vemuri. "Robust point set registration using gaussian mixture models." *IEEE Transactions on Pattern Analysis and Machine Intelligence* 33, no. 8 (2011): 1633-1645.

[48] Cheng-Yuan Liou, Wei-Chen Cheng, Jiun-Wei Liou, and Daw-Ran Liou. "Autoencoder for words." *Neurocomputing* 139 (2014): 84-96.

[49] AP Sarath Chandar, Stanislas Lauly, Hugo Larochelle, Mitesh Khapra, Balaraman Ravindran, Vikas C. Raykar, and Amrita Saha. "An autoencoder approach to learning bilingual word representations." In *Advances in Neural Information Processing Systems (NIPS)*, pp. 1853-1861. 2014.

[50] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. "Generative adversarial nets." In *Advances in Neural Information Processing Systems (NIPS)*, pp. 2672-2680. 2014.

[51] Arnab Ghosh, Viveka Kulharia, Vinay P. Namboodiri, Philip HS Torr, and Puneet K. Dokania. "Multi-agent diverse generative adversarial networks." In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 8513-8521. 2018.

[52] Antonia Creswell, Tom White, Vincent Dumoulin, Kai Arulkumaran, Biswa Sengupta, and Anil A. Bharath. "Generative adversarial networks: An overview." *IEEE Signal Processing Magazine* 35, no. 1 (2018): 53-65.

[53] Anil K. Jain. "Data clustering: 50 years beyond K-means." *Pattern Recognition Letters* 31, no. 8 (2010): 651-666.

[54] Hae-Sang Park, and Chi-Hyuck Jun. "A simple and fast algorithm for K-medoids clustering." *Expert Systems with Applications* 36, no. 2 (2009): 3336-3341.

[55] Ying Zhao, George Karypis, and Usama Fayyad. "Hierarchical clustering algorithms for document datasets." *Data mining and Knowledge Discovery* 10, no. 2 (2005): 141-168.

[56] Hans-Peter Kriegel, Peer Kröger, Jörg Sander, and Arthur Zimek. "Density-based clustering." *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery* 1, no. 3 (2011): 231-240.

[57] Alexander Lerch. "An introduction to audio content analysis: Applications in signal processing and music informatics." *Wiley-IEEE Press*, 2012.

[58] Tuomas Virtanen, Mark D. Plumbley, and Dan Ellis, eds. "Computational analysis of sound scenes and events." *Heidelberg: Springer*, 2018.

[59] Sadaoki Furui. "Digital speech processing: synthesis, and recognition." *CRC Press*, 2018.

[60] Jialie Shen, John Shepherd, Bin Cui, and Kian-Lee Tan. "A novel framework for efficient automated singer identification in large music databases." *ACM Transactions on Information Systems (TOIS)* 27, no. 3 (2009): 18.

[61] Zhouyu Fu, Guojun Lu, Kai Ming Ting, and Dengsheng Zhang. "A survey of audio-based music classification and annotation." *IEEE Transactions on Multimedia* 13, no. 2 (2011): 303-319.

[62] Stephen Downie. "Music information retrieval." *Annual review of Information Science and Technology* 37, no. 1 (2003): 295-340.

[63] Xiaodan Zhuang, Xi Zhou, Mark A. Hasegawa-Johnson, and Thomas S. Huang. "Real-world acoustic event detection." *Pattern Recognition Letters* 31, no. 12 (2010): 1543-1551.

[64] Li Deng, and Douglas O'Shaughnessy. "Speech processing: a dynamic and optimization-oriented approach." *CRC Press*, 2018.

[65] Ben Gold, Nelson Morgan, and Dan Ellis. "Speech and audio signal processing: processing and perception of speech and music." *John Wiley & Sons*, 2011.

[66] Udo Zölzer. "Digital audio signal processing". *John Wiley & Sons*, 2008.

[67] Dapeng Tao, Xu Lin, Lianwen Jin, and Xuelong Li. "Principal component 2-D long short-term memory for font recognition on single Chinese characters." *IEEE Transactions on Cybernetics* 46, no. 3 (2016): 756-765.

[68] Fayao Liu, Chunhua Shen, Guosheng Lin, and Ian Reid. "Learning depth from single monocular images using deep convolutional neural fields." *IEEE Transactions on Pattern Analysis and Machine Intelligence* 38, no. 10 (2016): 2024-2039.

[69] Tara N. Sainath, Brian Kingsbury, George Saon, Hagen Soltau, Abdel-rahman Mohamed, George Dahl, and Bhuvana Ramabhadran. "Deep convolutional neural networks for large-scale speech tasks." *Neural Networks* 64 (2015): 39-48.

[70] Jeffrey Donahue, Lisa Anne Hendricks, Sergio Guadarrama, Marcus Rohrbach, Subhashini Venugopalan, Kate Saenko, and Trevor Darrell. "Long-term recurrent convolutional networks for visual recognition and description." In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2625-2634. 2015.

[71] Hamid Palangi, Li Deng, Yelong Shen, Jianfeng Gao, Xiaodong He, Jianshu Chen, Xinying Song, and Rabab Ward. "Deep sentence embedding using long short-term memory networks: Analysis and application to information retrieval." *IEEE/ACM Transactions on Audio, Speech and Language Processing* 24, no. 4 (2016): 694-707.

[72] Qing Liao, Naiyang Guan, Chengkun Wu, and Qian Zhang. "Predicting unknown interactions between known drugs and targets via matrix completion." In *Pacific-Asia Conference on Knowledge Discovery and Data Mining*, pp. 591-604. Springer, Cham, 2016.

[73] Geoffrey E. Hinton, Simon Osindero, and Yee-Whye Teh. "A fast learning algorithm for deep belief nets." *Neural Computation* 18, no. 7 (2006): 1527-1554.

[74] Bo Du, Wei Xiong, Jia Wu, Lefei Zhang, Liangpei Zhang, and Dacheng Tao. "Stacked convolutional denoising auto-encoders for feature representation." *IEEE Transactions on Cybernetics* 47, no. 4 (2017): 1017-1027.

[75] George E. Dahl, Dong Yu, Li Deng, and Alex Acero. "Context-dependent pre-trained deep neural networks for large-vocabulary speech recognition." *IEEE Transactions on Audio, Speech and Language Processing* 20, no. 1 (2012): 30-42.

[76] Brian Hutchinson, Li Deng, and Dong Yu. "Tensor deep stacking networks." *IEEE Transactions on Pattern Analysis and Machine Intelligence* 35, no. 8 (2013): 1944-1957.

[77] Alex Graves, Abdel-rahman Mohamed, and Geoffrey Hinton. "Speech recognition with deep recurrent neural networks." In *2013 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 6645-6649. IEEE, 2013.

[78] Wim De Mulder, Steven Bethard, and Marie-Francine Moens. "A survey on the application of recurrent neural networks to statistical language modeling." *Computer Speech & Language*30, no. 1 (2015): 61-98.

[79] Martin Sundermeyer, Hermann Ney, and Ralf Schlüter. "From feedforward to recurrent LSTM neural networks for language modeling." *IEEE/ACM Transactions on Audio, Speech and Language Processing* 23, no. 3 (2015): 517-529.

[80] Jeffrey Donahue, Lisa Anne Hendricks, Sergio Guadarrama, Marcus Rohrbach, Subhashini Venugopalan, Kate Saenko, and Trevor Darrell. "Long-term recurrent convolutional networks for visual recognition and description." In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2625-2634. 2015.

[81] Georg Heigold, Hermann Ney, Ralph Schluter, and Simon Wiesler. "Discriminative training for automatic speech recognition: Modeling, criteria, optimization, implementation, and performance." *IEEE Signal Processing Magazine* 29, no. 6 (2012): 58-69.

[82] Shizhao Sun, Wei Chen, Liwei Wang, Xiaoguang Liu, and Tie-Yan Liu. "On the depth of deep neural networks: A theoretical view." In *Thirtieth AAAI Conference on Artificial Intelligence*. 2016.

[83] Sepp Hochreiter, and Jürgen Schmidhuber. "Long short-term memory." *Neural Computation* 9, no. 8 (1997): 1735-1780.

[84] Quoc V. Le, Navdeep Jaitly, and Geoffrey E. Hinton. "A simple way to initialize recurrent networks of rectified linear units." *arXiv preprint arXiv:1504.00941* (2015).

[85] Sepp Hochreiter. "The vanishing gradient problem during learning recurrent neural nets and problem solutions." *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems* 6, no. 02 (1998): 107-116.

[86] Rami Al-Rfou, Guillaume Alain, Amjad Almahairi, Christof Angermueller, Dzmitry Bahdanau, Nicolas Ballas, Frédéric Bastien *et al.* "Theano: A Python framework for fast computation of mathematical expressions." *arXiv preprint arXiv:1605.02688* (2016).

[87] Ibrahim Alabdulmohsin, Moustapha Cisse, Xin Gao, and Xiangliang Zhang. "Large margin classification with indefinite similarities." *Machine Learning* 103, no. 2 (2016): 215-237.

[88] Mitchell Marcus, Beatrice Santorini, and Mary Ann Marcinkiewicz. "Building a large annotated corpus of English: The Penn Treebank." *Computational linguistics-Association for Computational Linguistics* 19, no. 2: 313-330. (1993).

[89] César Laurent, Gabriel Pereyra, Philémon Brakel, Ying Zhang, and Yoshua Bengio. "Batch normalized recurrent neural networks." In *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 2657-2661. IEEE, 2016.

[90] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. "Dropout: a simple way to prevent neural networks from overfitting." *The Journal of Machine Learning Research* 15, no. 1 (2014): 1929-1958.

[91] Lituan Wang, Lei Zhang, and Zhang Yi. "Trajectory predictor by using recurrent neural networks in visual tracking." *IEEE Transactions on Cybernetics* 47, no. 10 (2017): 3172-3183.

[92] Andrej Karpathy, and Li Fei-Fei. "Deep visual-semantic alignments for generating image descriptions." In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 3128-3137. 2015.

[93] Xiaoyang Wang, and Qiang Ji. "Hierarchical context modeling for video event recognition." *IEEE Transactions on Pattern Analysis and Machine Intelligence* 39, no. 9 (2017): 1770-1782.

[94] Shaoqing Ren, Kaiming He, Ross Girshick, Xiangyu Zhang, and Jian Sun. "Object detection networks on convolutional feature maps." *IEEE Transactions on Pattern Analysis and Machine Intelligence* 39, no. 7 (2017): 1476-1481.

[95] Hao Liu, Jiwen Lu, Jianjiang Feng, and Jie Zhou. "Two-stream transformer networks for video-based face alignment." *IEEE Transactions on Pattern Analysis and Machine Intelligence* 40, no. 11 (2018): 2546-2554.

[96] Zecheng Xie, Zenghui Sun, Lianwen Jin, Hao Ni, and Terry Lyons. "Learning spatial-semantic context with fully convolutional recurrent network for online handwritten Chinese text recognition." *IEEE Transactions on Pattern Analysis and Machine Intelligence* 40, no. 8 (2018): 1903-1917.

[97] Lingyun Wu, Jie-Zhi Cheng, Shengli Li, Baiying Lei, Tianfu Wang, and Dong Ni. "FUIQA: Fetal ultrasound image quality assessment with deep convolutional networks." *IEEE Transactions on Cybernetics* 47, no. 5 (2017): 1336-1349.

[98] Kun Zeng, Jun Yu, Ruxin Wang, Cuihua Li, and Dacheng Tao. "Coupled deep autoencoder for single image super-resolution." *IEEE Transactions on Cybernetics* 47, no. 1 (2017): 27-37.

[99] Yan Huang, Wei Wang, and Liang Wang. "Video super-resolution via bidirectional recurrent convolutional networks." *IEEE Transactions on Pattern Analysis and Machine Intelligence* 40, no. 4 (2018): 1015-1028.

[100] Xu-Yao Zhang, Fei Yin, Yan-Ming Zhang, Cheng-Lin Liu, and Yoshua Bengio. "Drawing and recognizing chinese characters with recurrent neural network." *IEEE Transactions on Pattern Analysis and Machine Intelligence* 40, no. 4 (2018): 849-862.

[101] Pau Rodriguez, Guillem Cucurull, Jordi Gonzàlez, Josep M. Gonfaus, Kamal Nasrollahi, Thomas B. Moeslund, and F. Xavier Roca. "Deep pain: Exploiting long short-term memory networks for facial expression classification." *IEEE Transactions on Cybernetics* 99 (2017): 1-11.

[102] Herbert Jaeger. "Using conceptors to manage neural long-term memories for temporal patterns." *The Journal of Machine Learning Research* 18, no. 1 (2017): 387-429.

[103] Wanli Ouyang, Xingyu Zeng, Xiaogang Wang, Shi Qiu, Ping Luo, Yonglong Tian, Hongsheng Li *et al*. "DeepID-Net: Object detection with deformable part based convolutional neural networks." *IEEE Transactions on Pattern Analysis and Machine Intelligence* 39, no. 7 (2017): 1320-1334.

[104] Qiang Xiao, and Zhigang Zeng. "Scale-limited lagrange stability and finite-time synchronization for memristive recurrent neural networks on time scales." *IEEE Transactions on Cybernetics* 47, no. 10 (2017): 2984-2994.

[105] Sitian Qin, Xiudong Yang, Xiaoping Xue, and Jiahui Song. "A one-layer recurrent neural network for pseudoconvex optimization problems with equality and inequality constraints." *IEEE Transactions on Cybernetics* 47, no. 10 (2017): 3063-3074.

[106] Weiyang Liu, Yandong Wen, Zhiding Yu, and Meng Yang. "Large-Margin Softmax Loss for Convolutional Neural Networks." In *International Conference on Machine Learning (ICML),* vol. 2, no. 3, p. 7. 2016.

[107] Jure Sokolić, Raja Giryes, Guillermo Sapiro, and Miguel RD Rodrigues. "Robust large margin deep neural networks." *IEEE Transactions on Signal Processing* 65, no. 16 (2017): 4265-4280.

[108] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. "Gradient-based learning applied to document recognition." *Proceedings of the IEEE* 86, no. 11 (1998): 2278-2324.

[109] Sang-Gu Lee, and Quoc-Phong Vu. "Simultaneous solutions of Sylvester equations and idempotent matrices separating the joint spectrum." *Linear Algebra and its Applications* 435, no. 9 (2011): 2097-2109.

[110] Anil K. Jain, M. Narasimha Murty, and Patrick J. Flynn. "Data clustering: a review." *ACM computing surveys (CSUR)* 31, no. 3 (1999): 264-323.

[111] Chang Xu, Dacheng Tao, and Chao Xu. "A survey on multi-view learning." *arXiv preprint arXiv:1304.5634* (2013).

[112] Yingming Li, Ming Yang, and Zhongfei Mark Zhang. "A Survey of Multi-View Representation Learning." *IEEE Transactions on Knowledge and Data Engineering* (2018).

[113] Guoqing Chao, Shiliang Sun, and Jinbo Bi. "A survey on multi-view clustering." *arXiv preprint arXiv*:1712.06246 (2017).

[114] Chang Xu, Dacheng Tao, and Chao Xu. "Multi-view self-paced learning for clustering." In *Twenty-Fourth International Joint Conference on Artificial Intelligence.* 2015.

[115] Xiao He, and Luis Moreira-Matias. "Robust Continuous Co-Clustering." *arXiv preprint arXiv*:1802.05036 (2018).

[116] Wenzhang Zhuge, Chenping Hou, Yuanyuan Jiao, Jia Yue, Hong Tao, and Dongyun Yi. "Robust auto-weighted multi-view subspace clustering with common subspace representation matrix." *PloS One* 12, no. 5 (2017): e0176769.

[117] Changqing Zhang, Qinghua Hu, Huazhu Fu, Pengfei Zhu, and Xiaochun Cao. "Latent multi-view subspace clustering." In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 4279-4287. 2017.

[118] Xiaobo Wang, Zhen Lei, Xiaojie Guo, Changqing Zhang, Hailin Shi, and Stan Z. Li. "Multi-view subspace clustering with intactness-aware similarity." *Pattern Recognition* 88 (2019): 50-63.

[119] Yanbo Fan, Jian Liang, Ran He, Bao-Gang Hu, and Siwei Lyu. "Robust localized multi-view subspace clustering." *arXiv preprint arXiv*:1705.07777 (2017).

[120] Jeffrey Ho, Ming-Hsuan Yang, Jongwoo Lim, Kuang-Chih Lee, and David Kriegman. "Clustering appearances of objects under varying illumination conditions." In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 11-18. 2003.

[121] Paul Tseng. "Nearest q-flat to m points." *Journal of Optimization Theory and Applications* 105, no. 1 (2000): 249-252.

[122] João Paulo Costeira, and Takeo Kanade. "A multibody factorization method for independently moving objects." *International Journal of Computer Vision* 29, no. 3 (1998): 159-179.

[123] Ken-ichi Kanatani. "Motion segmentation by subspace separation and model selection." In *Proceedings Eighth IEEE International Conference on computer Vision. (ICCV)* 2001, vol. 2, pp. 586-591. IEEE, 2001.

[124] Michael E. Tipping, and Christopher M. Bishop. "Mixtures of probabilistic principal component analyzers." *Neural computation* 11, no. 2 (1999): 443-482.

[125] Jingyu Yan, and Marc Pollefeys. "A general framework for motion segmentation: Independent, articulated, rigid, non-rigid, degenerate and non-degenerate." In *European Conference on Computer Vision (ECCV)*, pp. 94-106. Springer, Berlin, Heidelberg, 2006.

[126] Alvina Goh, and René Vidal. "Segmenting motions of different types by unsupervised manifold clustering." In 2007 *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1-6. IEEE, 2007.

[127] Wei Zheng, Xiaofeng Zhu, Guoqiu Wen, Yonghua Zhu, Hao Yu, and Jiangzhang Gan. "Unsupervised feature selection by self-paced learning regularization." *Pattern Recognition Letters* (2018).

[128] Xiaofeng Zhu, Shichao Zhang, Yonggang Li, Jilian Zhang, Lifeng Yang, and Yue Fang. "Low-rank sparse subspace for spectral clustering." *IEEE Transactions on Knowledge and Data Engineering* (2018).

[129] Xiaofeng Zhu, Shichao Zhang, Rongyao Hu, and Yonghua Zhu. "Local and global structure preservation for robust unsupervised spectral feature selection." *IEEE Transactions on Knowledge and Data Engineering* 30, no. 3 (2018): 517-529.

[130] Wei Zheng, Xiaofeng Zhu, Yonghua Zhu, Rongyao Hu, and Cong Lei. "Dynamic graph learning for spectral feature selection." *Multimedia Tools and Applications* 77, no. 22 (2018): 29739-29755.

[131] Sohil A. Shah, and Vladlen Koltun. "Robust continuous clustering." *Proceedings of the National Academy of Sciences*114, no. 37 (2017): 9814-9819.

[132] Ehsan Elhamifar, and Rene Vidal. "Sparse subspace clustering: Algorithm, theory, and applications." *IEEE Transactions on Pattern Analysis and Machine Intelligence* 35, no. 11 (2013): 2765-2781.

[133] Maria Brbić, and Ivica Kopriva. "Multi-view low-rank sparse subspace clustering." *Pattern Recognition* 73 (2018): 247-258.

[134] Hongchang Gao, Feiping Nie, Xuelong Li, and Heng Huang. "Multi-view subspace clustering." In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pp. 4238-4246. 2015.

[135] Changqing Zhang, Huazhu Fu, Si Liu, Guangcan Liu, and Xiaochun Cao. "Low-rank tensor constrained multiview subspace clustering." In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pp. 1582-1590. 2015.

[136] Zhengming Ding, and Yun Fu. "Robust multi-view subspace learning through dual low-rank decompositions." In *Thirtieth AAAI conference on Artificial Intelligence*. 2016.

[137] Yanbo Fan, Jian Liang, Ran He, Bao-Gang Hu, and Siwei Lyu. "Robust localized multi-view subspace clustering." *arXiv preprint arXiv:1705.07777* (2017).

[138] Fredrik Lindsten, Henrik Ohlsson, and Lennart Ljung. "Just relax and come clustering!: A convexification of k-means clustering." *Linköping University Electronic Press*, 2011.

[139] Toby D. Hocking, Armand Joulin, Francis Bach, and Jean-Philippe Vert. "Clusterpath an algorithm for clustering using convex fusion penalties." In *28th International Conference on Machine Learning (ICML)*, p. 1. 2011.

[140] Eric C. Chi, and Kenneth Lange. "Splitting methods for convex clustering." *Journal of Computational and Graphical Statistics* 24, no. 4 (2015): 994-1013.

[141] Eric C. Chi, Genevera I. Allen, and Richard G. Baraniuk. "Convex biclustering." *Biometrics* 73, no. 1 (2017): 10-19.

[142] Feiping Nie, Heng Huang, Xiao Cai, and Chris H. Ding. "Efficient and robust feature selection via joint ℓ2, 1-norms minimization." In *Advances in Neural Information Processing Systems (NIPS)*, pp. 1813-1821. 2010.

[143] Kohei Ozaki, Masashi Shimbo, Mamoru Komachi, and Yuji Matsumoto. "Using the mutual k-nearest neighbor graphs for semi-supervised classification of natural language data." In *Proceedings of the fifteenth conference on Computational Natural Language Learning*, pp. 154-162. Association for Computational Linguistics, 2011.

[144] Peter J. Green. "Iteratively reweighted least squares for maximum likelihood estimation, and some robust and resistant alternatives." *Journal of the Royal Statistical Society: Series B (Methodological)* 46, no. 2 (1984): 149-170.

[145] Stuart Geman. "Statistical methods for tomographic image reconstruction." *Bull. Int. Stat. Inst* 4 (1987): 5-21.

[146] Hong Tao, Chenping Hou, Feiping Nie, Yuanyuan Jiao, and Dongyun Yi. "Effective discriminative feature selection with nontrivial solution." *IEEE Transactions on Neural Networks and Learning Systems* 27, no. 4 (2016): 796-808.

[147] Fei-Fei Li, Rob Fergus, and Pietro Perona. "Learning generative visual models from few training examples: An incremental bayesian approach tested on 101 object categories." *Computer Vision and Image Understanding* 106, no. 1 (2007): 59-70.

[148] Arthur Asuncion, and David Newman. "UCI machine learning repository." http://www. ics. uci. edu/mlearn/MLRepository. html (2007).

[149] Vikas Sindhwani, Partha Niyogi, and Mikhail Belkin. "Beyond the point cloud: from transductive to semi-supervised learning." In *Proceedings of the 22nd International Conference on Machine Learning (ICML)*, pp. 824-831. ACM, 2005.

[150] Grigorios F. Tzortzis, and Aristidis C. Likas. "Multiple view clustering using a weighted combination of exemplar-based mixture models." *IEEE Transactions on Neural Networks* 21, no. 12 (2010): 1925-1938.

[151] Laurens van der Maaten, and Geoffrey Hinton. "Visualizing data using t-SNE." *Journal of Machine Learning Research* 9, no. Nov (2008): 2579-2605.

[152] Xiao Cai, Feiping Nie, and Heng Huang. "Multi-view k-means clustering on big data." In *Twenty-Third International Joint Conference on Artificial Intelligence*. 2013.

[153] Abhishek Kumar, Piyush Rai, and Hal Daume. "Co-regularized multi-view spectral clustering." In *Advances in Neural Information Processing Systems (NIPS)*, pp. 1413-1421. 2011.

[154] Yong Jae Lee, and Kristen Grauman. "Foreground focus: Unsupervised learning from partially matching images." *International Journal of Computer Vision* 85, no. 2 (2009): 143-166.

[155] Xiaochun Cao, Changqing Zhang, Huazhu Fu, Si Liu, and Hua Zhang. "Diversity-induced multi-view subspace clustering." In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 586-594. 2015.

[156] Biljana L. Stojkoska, Risteska, and Kire V. Trivodaliev. "A review of Internet of Things for smart home: Challenges and solutions." *Journal of Cleaner Production* 140 (2017): 1454-1464.

[157] Jiachen Yang, Shudong He, Yancong Lin, and Zhihan Lv. "Multimedia cloud transmission and storage system based on internet of things." *Multimedia Tools and Applications* 76, no. 17 (2017): 17735-17750.

[158] Peng Li, Zhikui Chen, Laurence Tianruo Yang, Qingchen Zhang, and M. Jamal Deen. "Deep convolutional computation model for feature learning on big data in Internet of Things." *IEEE Transactions on Industrial Informatics* 14, no. 2 (2018): 790-798.

[159] Mehdi Mohammadi, Ala Al-Fuqaha, Sameh Sorour, and Mohsen Guizani. "Deep learning for IoT big data and streaming analytics: A survey." *IEEE Communications Surveys & Tutorials* 20, no. 4 (2018): 2923-2960.

[160] Mohammad Saeid Mahdavinejad, Mohammadreza Rezvan, Mohammadamin Barekatain, Peyman Adibi, Payam Barnaghi, and Amit P. Sheth. "Machine learning for Internet of Things data analysis: A survey." *Digital Communications and Networks* 4, no. 3 (2018): 161-175.

[161] Sharath Adavanne, and Tuomas Virtanen. "A report on sound event detection with different binaural features." *arXiv preprint arXiv:1710.02997* (2017).

[162] Sharath Adavanne, Archontis Politis, and Tuomas Virtanen. "Multichannel sound event detection using 3D convolutional neural networks for learning inter-channel features." In *2018 International Joint Conference on Neural Networks (IJCNN)*, pp. 1-7. IEEE, 2018.

[163] Ilyas Ozer, Zeynep Ozer, and Oguz Findik. "Noise robust sound event classification with convolutional neural network." *Neurocomputing* 272 (2018): 505-512.

[164] Hao-min Zhang, Ian Vince McLoughlin, and Yan Song. "Robust sound event detection in continuous audio environments." (2016).

[165] Donn Morrison, Ruili Wang, and Liyanage C. De Silva. "Spoken affect classification using neural networks." In *2005 IEEE International Conference on Granular Computing*, vol. 2, pp. 583-586. IEEE, 2005.

[166] Donn Morrison, Ruili Wang, Liyanage C. De Silva, and W. L. Xu. "Real-time spoken affect classification and its application in call-centres." In *Third International Conference on Information Technology and Applications (ICITA'05)*, vol. 1, pp. 483-487. IEEE, 2005.

[167] Ruili Wang, Wanting Ji, Mingzhe Liu, Xun Wang, Jian Weng, Song Deng, Suying Gao, and Chang-an Yuan. "Review on mining data from multiple data sources." *Pattern Recognition Letters* 109 (2018): 120-128.

[168] Wanting Ji, Ruili Wang, and Junbo Ma. "Dictionary-based active learning for sound event classification." *Multimedia Tools and Applications* 78, no. 3 (2019): 3831-3842.

[169] Sharath Adavanne, Giambattista Parascandolo, Pasi Pertilä, Toni Heittola, and Tuomas Virtanen. "Sound event detection in multichannel audio using spatial and harmonic features." *arXiv preprint arXiv:1706.02293* (2017).

[170] Emre Çakir, and Tuomas Virtanen. "End-to-End polyphonic sound event detection using convolutional recurrent neural networks with learned time-frequency representation input." In *2018 International Joint Conference on Neural Networks (IJCNN)*, pp. 1-7. IEEE, 2018.

[171] Adam Santoro, Ryan Faulkner, David Raposo, Jack Rae, Mike Chrzanowski, Theophane Weber, Daan Wierstra, Oriol Vinyals, Razvan Pascanu, and Timothy Lillicrap. "Relational recurrent neural networks." In *Advances in Neural Information Processing Systems (NIPS)*, pp. 7310-7321. 2018.

[172] Annamaria Mesaros, Toni Heittola, Antti Eronen, and Tuomas Virtanen. "Acoustic event detection in real life recordings." In *2010 18th European Signal Processing Conference*, pp. 1267-1271. IEEE, 2010.

[173] Toni Heittola, Annamaria Mesaros, Antti Eronen, and Tuomas Virtanen. "Context-dependent sound event detection." *EURASIP Journal on Audio, Speech, and Music Processing*2013, no. 1 (2013): 1.

[174] Annamaria Mesaros, Toni Heittola, Onur Dikmen, and Tuomas Virtanen. "Sound event detection in real life recordings using coupled matrix factorization of spectral representations and class activity annotations." In *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 151-155. IEEE, 2015.

[175] Emre Cakir, Toni Heittola, Heikki Huttunen, and Tuomas Virtanen. "Polyphonic sound event detection using multi label deep neural networks." In *2015 international joint conference on neural networks (IJCNN)*, pp. 1-7. IEEE, 2015.

[176] An Dang, Toan H. Vu, and Jia-Ching Wang. "A survey of deep learning for polyphonic sound event detection." In *2017 International Conference on Orange Technologies (ICOT)*, pp. 75-78. IEEE, 2017.

[177] Il-Young Jeong, Subin Lee, Yoonchang Han, and Kyogu Lee. "Audio event detection using multiple-input convolutional neural network." *Detection and Classification of Acoustic Scenes and Events (DCASE)* (2017).

[178] Yukun Chen, Yichi Zhang, and Zhiyao Duan. "DCASE2017 sound event detection using convolutional neural network." *Detection and Classification of Acoustic Scenes and Events* (2017).

[179] Huy Phan, Martin Krawczyk-Becker, Timo Gerkmann, and Alfred Mertins. "DNN and CNN with weighted and multi-task loss functions for audio event detection." *arXiv preprint arXiv:1708.03211* (2017).

[180] Sharath Adavanne, Giambattista Parascandolo, Pasi Pertilä, Toni Heittola, and Tuomas Virtanen. "Sound event detection in multichannel audio using spatial and harmonic features." *arXiv preprint arXiv:1706.02293* (2017).

[181] Toan H. Vu, and Jia-Ching Wang. "Acoustic scene and event recognition using recurrent neural networks." *Detection and Classification of Acoustic Scenes and Events* 2016 (2016).

[182] Giambattista Parascandolo, Heikki Huttunen, and Tuomas Virtanen. "Recurrent neural networks for polyphonic sound event detection in real life recordings." In *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 6440-6444. IEEE, 2016.

[183] Jianchao Zhou. "Sound event detection in multichannel audio LSTM network." *Detection and Classification of Acoustic Scenes and Events (DCASE)* (2017).

[184] Emre Cakır, Giambattista Parascandolo, Toni Heittola, Heikki Huttunen, and Tuomas Virtanen. "Convolutional recurrent neural networks for polyphonic sound event detection." *IEEE/ACM Transactions on Audio, Speech, and Language Processing* 25, no. 6 (2017): 1291-1303.

[185] Sharath Adavanne, Pasi Pertilä, and Tuomas Virtanen. "Sound event detection using spatial features and convolutional recurrent neural network." In *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 771-775. IEEE, 2017.

[186] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, and Illia Polosukhin. "Attention is all you need." In *Advances in Neural Information Processing Systems (NIPS)*, pp. 5998-6008. 2017.

[187] Annamaria Mesaros, Toni Heittola, and Tuomas Virtanen. "TUT database for acoustic scene classification and sound event detection." In *2016 24th European Signal Processing Conference (EUSIPCO)*, pp. 1128-1132. IEEE, 2016.

[188] Annamaria Mesaros, Toni Heittola, and Tuomas Virtanen. "Metrics for polyphonic sound event detection." *Applied Sciences* 6, no. 6 (2016): 162.

[189] Graham E. Poliner, and Daniel PW Ellis. "A discriminative model for polyphonic piano transcription." *EURASIP Journal on Advances in Signal Processing* 2007, no. 1 (2006): 048317.

[190] http://dcase.community/

[191] https://github.com/deepmind/sonnet

[192] Martín Abadi, Paul Barham, Jianmin Chen, Zhifeng Chen, Andy Davis, Jeffrey Dean, Matthieu Devin *et al.* "Tensorflow: A system for large-scale machine learning." In *12th Symposium on Operating Systems Design and Implementation (OSDI 16)*, pp. 265-283. 2016.

[193] Diederik P. Kingma, and Jimmy Ba. "Adam: A method for stochastic optimization." *arXiv preprint arXiv:1412.6980*(2014).

[194] Sergey Ioffe, and Christian Szegedy. "Batch normalization: Accelerating deep network training by reducing internal covariate shift." *arXiv preprint arXiv:1502.03167* (2015).

[195] Roderick JA. Little. "Modeling the drop-out mechanism in repeated-measures studies." *Journal of the American Statistical Association* 90, no. 431 (1995): 1112-1121.

[196] http://www.cs.tut.fi/sgn/arg/dcase2016/task-sound-event-detection-in-real-life-audio

[197] http://www.cs.tut.fi/sgn/arg/dcase2017/challenge/task-sound-event-detection-in-real-life-audio

[198] Ying-Hui Lai, Chun-Hao Wang, Shi-Yan Hou, Bang-Yin Chen, Yu Tsao, and Yi-Wen Liu. "DCASE report for task 3: Sound event detection in real life audio." *IEEE AASP Challenge: Detection and Classification of Acoustic Scenes and Events* (*DCASE* 2016).

[199] Christian Kroos, and Mark Plumbley. "Neuroevolution for sound event detection in real life audio: A pilot study." *Detection and Classification of Acoustic Scenes and Events (DCASE 2017) Proceedings 2017* (2017).

[200] An Dang, Toan H. Vu, and Jia-Ching Wang. "Deep learning for DCASE2017 challenge." *Detection and Classification of Acoustic Scenes and Events (DCASE 2017)*.

[201] Thomas Elsken, Jan Hendrik Metzen, and Frank Hutter. "Neural Architecture Search: A Survey." *Journal of Machine Learning Research* 20, no. 55 (2019): 1-21.

[202] Chenxi Liu, Barret Zoph, Maxim Neumann, Jonathon Shlens, Wei Hua, Li-Jia Li, Li Fei-Fei, Alan Yuille, Jonathan Huang, and Kevin Murphy. "Progressive neural architecture search." In *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 19-34. 2018.

[203] Barret Zoph, and Quoc V. Le. "Neural architecture search with reinforcement learning." *arXiv preprint arXiv:1611.01578*(2016).

[204] Hieu Pham, Melody Guan, Barret Zoph, Quoc Le, and Jeff Dean. "Efficient Neural Architecture Search via Parameter Sharing." In *International Conference on Machine Learning (ICML)*, pp. 4092-4101. 2018.

[205] Han Cai, Tianyao Chen, Weinan Zhang, Yong Yu, and Jun Wang. "Efficient architecture search by network transformation." In *Thirty-Second AAAI Conference on Artificial Intelligence*. 2018.

[206] Xiaojin Zhu. "Semi-supervised learning literature survey." *University of Wisconsin-Madison Department of Computer Sciences*, 2005.

[207] Burr Settles. "*Active learning literature survey.*" *University of Wisconsin-Madison Department of Computer Sciences*, 2009.

[208] Meng Wang, and Xian-Sheng Hua. "Active learning in multimedia annotation and retrieval: A survey." *ACM Transactions on Intelligent Systems and Technology* 2, no. 2 (2011): 10.

[209] Jürgen Bernard, Marco Hutter, Matthias Zeppelzauer, Dieter Fellner, and Michael Sedlmair. "Comparing visual-interactive labeling with active learning: An experimental study." *IEEE Transactions on Visualization and Computer Graphics* 24, no. 1 (2018): 298-308.

[210] Sicheng Xiong, Javad Azimi, and Xiaoli Z. Fern. "Active learning of constraints for semi-supervised clustering." *IEEE Transactions on Knowledge and Data Engineering* 26, no. 1 (2014): 43-54.

[211] Min Wang, Fan Min, Zhi-Heng Zhang, and Yan-Xue Wu. "Active learning through density clustering." *Expert Systems with Applications* 85 (2017): 305-317.

[212] Sheng Li, Zhiqiang Tao, Kang Li, and Yun Fu. "Visual to Text: Survey of Image and Video Captioning." *IEEE Transactions on Emerging Topics in Computational Intelligence* (2019).

[213] Ranjay Krishna, Kenji Hata, Frederic Ren, Li Fei-Fei, and Juan Carlos Niebles. "Dense-captioning events in videos." In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pp. 706-715. 2017.

[214] Lianli Gao, Zhao Guo, Hanwang Zhang, Xing Xu, and Heng Tao Shen. "Video captioning with attention-based LSTM and semantic consistency." *IEEE Transactions on Multimedia* 19, no. 9 (2017): 2045-2055.

[215] Luowei Zhou, Yingbo Zhou, Jason J. Corso, Richard Socher, and Caiming Xiong. "End-to-end dense video captioning with masked transformer." In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR),* pp. 8739-8748. 2018.

[216] Yang Yang, Jie Zhou, Jiangbo Ai, Yi Bin, Alan Hanjalic, Heng Tao Shen, and Yanli Ji. "Video captioning by adversarial lstm." *IEEE Transactions on Image Processing* 27, no. 11 (2018): 5600-5611.

[217] Yehao Li, Ting Yao, Yingwei Pan, Hongyang Chao, and Tao Mei. "Jointly localizing and describing events for dense video captioning." In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR),* pp. 7492-7500. 2018.

[218] Junbo Wang, Wei Wang, Yan Huang, Liang Wang, and Tieniu Tan. "M3: multimodal memory modelling for video captioning." In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR),* pp. 7512-7520. 2018.

[219] Ning Xu, An-An Liu, Yongkang Wong, Yongdong Zhang, Weizhi Nie, Yuting Su, and Mohan Kankanhalli. "Dual-stream recurrent neural network for video captioning." *IEEE Transactions on Circuits and Systems for Video Technology* (2018).

[220] Jeong-Sik Park, and Seok-Hoon Kim. "Sound learning–based event detection for acoustic surveillance sensors." *Multimedia Tools and Applications* (2019): 1-13.

[221] Noor Almaadeed, Muhammad Asim, Somaya Al-Maadeed, Ahmed Bouridane, and Azeddine Beghdadi. "Automatic detection and classification of audio events for road surveillance applications." *Sensors* 18, no. 6 (2018): 1858.

[222] Minh Pham, Yehenew Mengistu, Ha Do, and Weihua Sheng. "Delivering home healthcare through a cloud-based smart home environment (CoSHE)." *Future Generation Computer Systems* 81 (2018): 129-140.

[223] Syed M. Adnan, Aun Irtaza, Sumair Aziz, M. Obaid Ullah, Ali Javed, and Muhammad Tariq Mahmood. "Fall detection through acoustic Local Ternary Patterns." *Applied Acoustics*140 (2018): 296-300.

[224] Ryosuke Kojima, Osamu Sugiyama, Kotaro Hoshiba, Reiji Suzuki, and Kazuhiro Nakadai. "HARK-Bird-Box: A Portable Real-time Bird Song Scene Analysis System." In *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 2497-2502. IEEE, 2018.

[225] Dan Stowell, Michael D. Wood, Hanna Pamuła, Yannis Stylianou, and Hervé Glotin. "Automatic acoustic detection of birds through deep learning: the first Bird Audio Detection challenge." *Methods in Ecology and Evolution* 10, no. 3 (2019): 368-380.

Appendix A List of Publications

- **Junbo Ma**, Ruili Wang, Wanting Ji, Jiawei Zhao, Ming Zong, and Andrew Gilman. "Robust multi-view continuous subspace clustering." *Pattern Recognition Letters* (2018). https://doi.org/10.1016/j.patrec.2018.12.004
- **Junbo Ma**, Ruili Wang, Wanting Ji, Hao Zheng, En Zhu, and Jianping Yin. "Relational recurrent neural networks for polyphonic sound event detection." *Multimedia Tools and Applications* (2019): 1-19. https://doi.org/10.1007/s11042-018-7142-7
- Ji, Wanting, Ruili Wang, and **Junbo Ma**. "Dictionary-based active learning for sound event classification." *Multimedia Tools and Applications* 78, no. 3 (2019): 3831-3842. https://doi.org/10.1007/s11042-018-6380-z
- **Junbo Ma**, Ruili Wang, Andrew Gilman, Seyed Reza Shahamiri, and Jianping Yin. "Large Margin Recurrent Neural Networks" IEEE Transactions on Cybernetics. (submitted, under review)

MASSEY UNIVERSITY
GRADUATE RESEARCH SCHOOL

STATEMENT OF CONTRIBUTION
DOCTORATE WITH PUBLICATIONS/MANUSCRIPTS

We, the candidate and the candidate’s Primary Supervisor, certify that all co-authors have consented to their work being included in the thesis and they have accepted the candidate’s contribution as indicated below in the *Statement of Originality*.

Name of candidate:	Junbo Ma
Name/title of Primary Supervisor:	Professor Ruili Wang
Name of Research Output and full reference:	
Ma, Junbo, Ruili Wang, Wanting Ji, Jiawei Zhao, Ming Zong, and Andrew Gilman. "Robust multi-view continuous subspace clustering." Pattern Recognition Letters (2018). https://doi.org/10.1016/j.patrec.2018.12.004	
In which Chapter is the Manuscript /Published work:	Chapter 3
Please indicate:	
• The percentage of the manuscript/Published Work that was contributed by the candidate:	70%
and	
• Describe the contribution that the candidate has made to the Manuscript/Published Work:	
As the first author, Junbo has made the major contribution of this published work. Junbo developed the theory, performed the computations, analyzed the results and wrote the manuscript under the supervision of Prof. Ruili Wang.	
For manuscripts intended for publication please indicate target journal:	
Candidate’s Signature:	Junbo Ma Digitally signed by Junbo Ma Date: 2019.04.27 12:13:23 +12'00'
Date:	27/04/2019
Primary Supervisor’s Signature:	Prof Ruili Wang Digitally signed by Prof Ruili Wang Date: 2019.04.28 17:19:21 +12'00'
Date:	

(This form should appear at the end of each thesis chapter/section/appendix submitted as a manuscript/ publication or collected as an appendix at the end of the thesis)

MASSEY UNIVERSITY
GRADUATE RESEARCH SCHOOL

STATEMENT OF CONTRIBUTION
DOCTORATE WITH PUBLICATIONS/MANUSCRIPTS

We, the candidate and the candidate’s Primary Supervisor, certify that all co-authors have consented to their work being included in the thesis and they have accepted the candidate’s contribution as indicated below in the *Statement of Originality*.

Name of candidate:	Junbo Ma
Name/title of Primary Supervisor:	Professor Ruili Wang
Name of Research Output and full reference:	
Ma, Junbo, Ruili Wang, Wanting Ji, Hao Zheng, En Zhu, and Jianping Yin. "Relational recurrent neural networks for polyphonic sound event detection." Multimedia Tools and Applications (2019): 1-19. https://doi.org/10.1007/s11042-018-7142-7	
In which Chapter is the Manuscript /Published work:	Chapter 4
Please indicate:	
• The percentage of the manuscript/Published Work that was contributed by the candidate:	70%
and	
• Describe the contribution that the candidate has made to the Manuscript/Published Work:	
As the first author, Junbo has made the major contribution of this published work. Junbo developed the theory, performed the computations, analyzed the results and wrote the manuscript under the supervision of Prof. Ruili Wang.	
For manuscripts intended for publication please indicate target journal:	
Candidate’s Signature:	Junbo Ma Digitally signed by Junbo Ma Date: 2019.04.27 12:16:13 +12'00'
Date:	27/04/2019
Primary Supervisor’s Signature:	Prof Ruili Wang Digitally signed by Prof Ruili Wang Date: 2019.04.28 17:20:38 +12'00'
Date:	

(This form should appear at the end of each thesis chapter/section/appendix submitted as a manuscript/ publication or collected as an appendix at the end of the thesis)