AUTOMATIC SOUND CLASSIFICATION USING DEEP LEARNING NETWORKS

Dissertation submitted in fulfilment of the requirements for the Degree of

MASTER OF TECHNOLOGY

In

COMPUTER SCIENCE AND ENGINEERING

By

KARAN MEHTAB SINGH

11602389

Supervisor

ADITYA KHAMPARIA

School of Computer Science and Engineering

Lovely Professional University

Phagwara, Punjab (India)

November 2017

November 2017

TOPIC APPROVAL PERFORMA

School of Computer Science and Engineering

Program : P172::M.Tech. (Computer Science and Engineering) [Full Time]

COURSE CODE : CSE548 **REGULAR/BACKLOG :** Regular **GROUP NUMBER :** CSERGD0039

Supervisor Name : Aditya Khamparia **UID :** 17862 **Designation :** Assistant Professor

Qualification : _____ **Research Experience :** _____

SR.NO.	NAME OF STUDENT	REGISTRATION NO	BATCH	SECTION	CONTACT NUMBER
1	Karan Mehtab Singh	11602389	2016	K1637	9872263522

SPECIALIZATION AREA : Intelligent Systems **Supervisor Signature:** _____

PROPOSED TOPIC : Automatic sound classification using Convolutional deep neural networks

Qualitative Assessment of Proposed Topic by PAC		
Sr.No.	**Parameter**	**Rating (out of 10)**
1	Project Novelty: Potential of the project to create new knowledge	7.25
2	Project Feasibility: Project can be timely carried out in-house with low-cost and available resources in the University by the students.	7.25
3	Project Academic Inputs: Project topic is relevant and makes extensive use of academic inputs in UG program and serves as a culminating effort for core study area of the degree program.	7.00
4	Project Supervision: Project supervisor's is technically competent to guide students, resolve any issues, and impart necessary skills.	7.50
5	Social Applicability: Project work intends to solve a practical problem.	7.25
6	Future Scope: Project has potential to become basis of future research work, publication or patent.	6.75

PAC Committee Members		
PAC Member 1 Name: Prateek Agrawal	UID: 13714	Recommended (Y/N): NA
PAC Member 2 Name: Pushpendra Kumar Pateriya	UID: 14623	Recommended (Y/N): Yes
PAC Member 3 Name: Deepak Prashar	UID: 13897	Recommended (Y/N): Yes
PAC Member 4 Name: Kewal Krishan	UID: 11179	Recommended (Y/N): Yes
PAC Member 5 Name: Anupinder Singh	UID: 19385	Recommended (Y/N): NA
DAA Nominee Name: Kanwar Preet Singh	UID: 15367	Recommended (Y/N): Yes

Final Topic Approved by PAC: **Automatic sound classification using Convolutional deep neural networks**

Overall Remarks: Approved

PAC CHAIRPERSON Name: 11024::Amandeep Nagpal **Approval Date:** 05 Mar 2017

Abstract

The deep learning is a subclass of Machine Learning algorithms which tends to learn the multiple representations of data. These representations are learned with different level of abstraction in each representation. The working of deep learning is based on the cascade style processing means it uses multiple nonlinear layers for feature extraction from given data. Sound classification is a major problem in many applications due to the different pitch rates and different dialects for a single word. An automatic system for classifying the sounds using deep learning algorithms would have many applications. There are many deep learning algorithms are available these days. In this research work, the proposed model will classify the sounds by using Tensor Deep Stacking Network (T-DSN). Along with Tensor deep stacking network a HMM and Softmax layer will be used to gain the low error rate in classification.

DECLARATION STATEMENT

I hereby declare that the research work reported in the dissertation proposal entitled "AUTOMATIC SOUND CLASSIFICATION USING DEEP LEARNING NETWORKS" in partial fulfilment of the requirement for the award of Degree for Master of Technology in Computer Science and Engineering at Lovely Professional University, Phagwara, Punjab is an authentic work carried out under supervision of my research supervisor Mr. Aditya Khamparia. I have not submitted this work elsewhere for any degree.

I understand that the work presented herewith is in direct compliance with Lovely Professional University's Policy on plagiarism, intellectual property rights, and highest standards of moral and ethical conduct. Therefore, to the best of my knowledge, the content of this dissertation represents authentic and honest research effort conducted, in its entirety, by me. I am fully responsible for the contents of my dissertation work.

.....

Karan Mehtab Singh

R.No: 11602389

SUPERVISOR'S CERTIFICATE

This is to certify that the work reported in the M.Tech dissertation proposal entitled "**AUTOMATIC SOUND CLASSIFICATION USING DEEP LEARNING NETWORKS"**, submitted by **Karan Mehtab Singh** at **Lovely Professional University, Phagwara, India** is a bonafide record of his original work carried out under my supervision. This work has not been submitted elsewhere for any other degree.

.....

Aditya Khamparia

Date: 28 Nov 2017

Counter Signed by:

1) **Concerned HOD:**
   HoD's Signature: _____

   HoD Name: _____

   Date: _____

2) **Neutral Examiners:**

   External Examiner

   Signature: _____

   Name: _____

   Affiliation: _____

   Date: _____

   Internal Examiner

   Signature: _____

   Name: _____

   Date: _____

ACKNOWLEDGEMENT

I would like to thanks Mr. Aditya Khampariya for his timely help and guidance for this dissertation. I would also like to thanks to my classmates and friends for their continuous support for my work. Most importantly I would like to thanks my family and best friend for their support and motivation for this whole work.

LIST OF FIGURES

TABLE OF CONTENTS

TABLE OF CONTENTS

CHAPTER-1

INTRODUCTION

1.1 Learning

1.1.1 Definition of Learning:

In Artificial Intelligence an agent is someone who has brain and capable of acquiring new knowledge from its surroundings. The process of acquiring this new knowledge is known as learning [1]. Learning is either a internal process to a machine or external process. The level of intelligence of a machine depends upon learning. Because in a way similar to humans learning a set of knowledge is what makes a person intelligent. Knowledge about a concept helps the machine in inference or simply in decision making, which is directly the measure of the intelligency of a machine.

1.1.2 Types of Learning:

Learning is a mapping of input to output, means evaluating the current state and generating a proper action for that state is a learning for a agent in artificial intelligence. Learning is based on training data and it consists of examples. The training data is either labeled or unlabeled. Based upon the relation between input and output learning can be divided into 3 main categories [2]. These are:

a) Supervised Learning
b) Unsupervised Learning
c) Reinforcement Learning

- **Supervised Learning**: The training data is available in this type and it must be labeled. Simply saying the input and output of the machine are well determined and are directly supplied to the training algorithm [3]. For every supplied input error is calculated by comparing the actual output of the machine with the desired output of the machine. After error calculation, the parameters of the system are changed to reduce the error. The learning algorithm generalizes from the training data to work with unseen data. Supervised learning in machine learning can be compared with human learning in the presence of a teacher and concept learning. Supervised learning suffers from the bias variance dilemma [4].

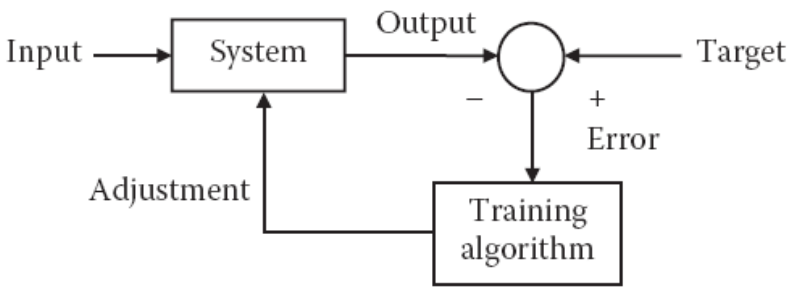

Figure 1.1 Supervised Learning [13]

- **Unsupervised Learning:** In unsupervised learning the training data is unlabeled. System draws an inference based on this unlabeled training data. According to this inference the parameters of the system are changed [5]. Since the training data given to the system is unlabeled, there is no evaluation of the output generated by the system [6].

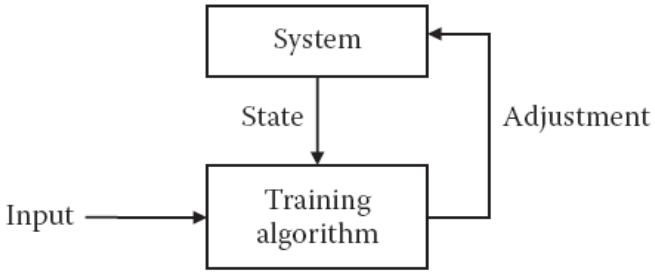

Figure 1.2 Unsupervised Learning [13]

- **Reinforcement Learning:** The working of reinforcement learning is very different from other two types. Here in this learning system works on a reward and punish mechanism. The software agent takes an action in its environment to maximize the reward and minimize the punishment [7]. The reinforcement learning uses the basic principles of dynamic programming and it does not require any knowledge about the Markov Decision Process instead it focuses on the online performance [7] [8].

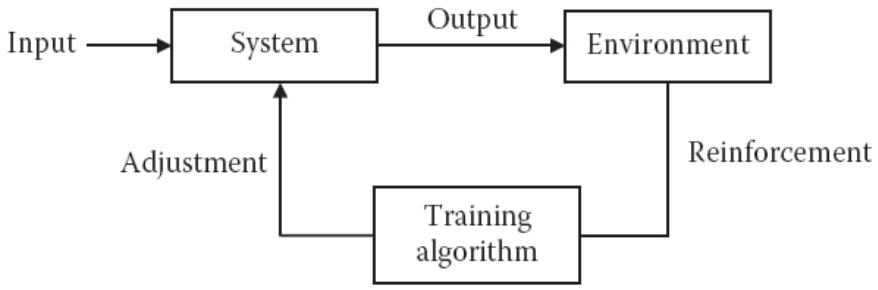

Figure 1.3 Reinforcement Learning [13]

1.2 Deep Learning

The deep learning is a subclass of Machine Learning algorithms which tends to learn the multiple representations of data. These representations are learned with different level of abstraction at each level. It can be supervised or unsupervised [9] [10]. The working of deep learning is very much similar to human brain.

Deep learning uses multiple levels of nonlinear processing units to extract the important features from the given data. The output of current layer will be used as the input to the next layer [11]. In the training phase stochastic gradient descent approach is used via back propagation algorithm to achieve the saturation.

1.3 Why Deep Learning?

Traditional learning algorithms are not useful where he have large dimensionality in data means number of inputs and outputs are very large. There is no way to direct the feature

extraction process to a particular direction in traditional algorithms. But in deep learning the multiple layers of the network finds the important features in the data and guides the extraction process to a more promising direction [12].

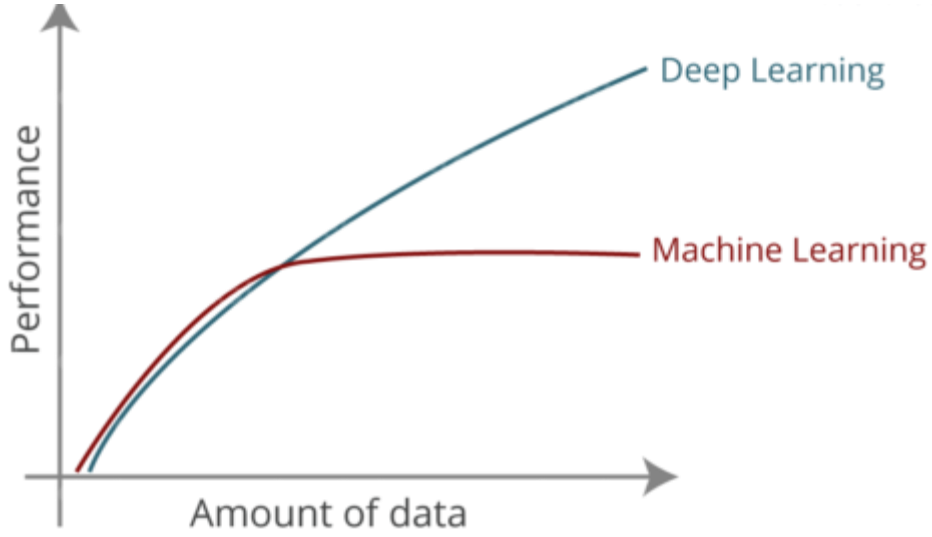

Figure 1.4 Deep Learning Performance Curve [14]

1.4 Deep Network Architectures

When humans try to solve problems those are related to AI especially related to recognition, they divide the problem into sub problems and different level of representations. These representations are used repeatedly in the sub problems to reach the final prediction to the problem [16]. For example the figure 5 shows a man sitting in field, the state of art in machine vision is an assembly of different modules are dividing the problem into sub problems. Recognition starts with the edge detection or low level feature extraction usually these low level features are invariant to geometrical transformations. Then these features are used to detect the most frequently occurring patterns in the image. These patterns are then used to detect the sub objects and these sub objects are assembled to recognize the scene [15].

The problem with simple architectures is in the abstract generating capabilities. In the Man sitting examples many different levels of abstractions will be used which are beyond the capabilities of simple neural architectures [18].

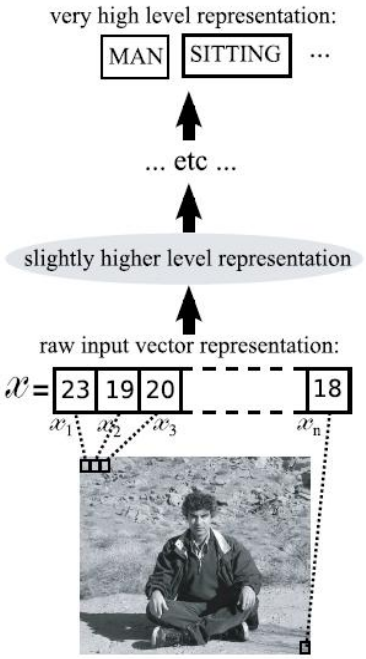

Figure 1.5 Man Detection Problem [15]

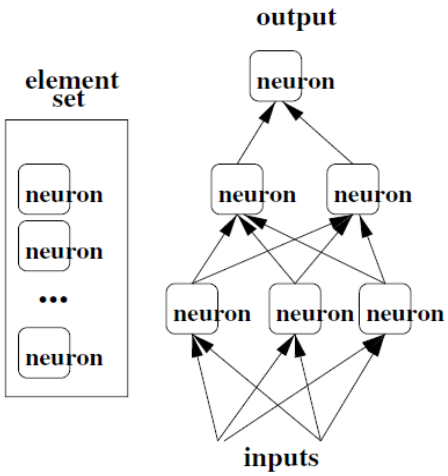

Figure 1.6 An Illustration of Deep Architecture [15]

Figure 6 shows a representation of deep architecture with a depth of 3. The image of man sitting will be given to the network at the input layer. The first hidden layer will extract the most basic feature like edges or pixel value of the image. The next layer will put these low

level features together and identify sub parts like eyes, mouth and bushes in background. The final hidden layer recognizes these subparts and generates the final prediction of the image.

1.5 Tensor Deep Stacking Network

The Tensor Deep Stacking Network (T-DSN) is an extension to the Deep stacking network (DSN). These architectures are the sub classes of Deep Generative Architectures [19]. In these graphical models the modules are stacked over one another to reach the final prediction. Sometimes the original input vector is also concatenated with the intermediate output of the hidden layer to achieve more accuracy than previous layer. Deep stacking network is different from other architectures because the here instead of using gradient descent approach, it works on the principle of mean square error between the current module's prediction and final prediction value [20].

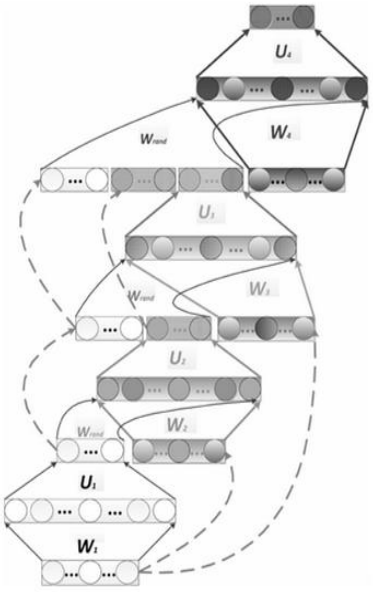

Figure 1.7 Illustration of Deep Stacking Network [20]

T-DSN is similar to DSN but instead of having sequential hidden layers in each module, it has two parallel hidden layers in each module. These two parallel hidden layer units will provide an ability to capture the higher order feature interaction through the use of cross products.

In tensor notation the operation will be:

$$y = \mu(h_{(1)}, h_{(2)}) \cong (\mu \times_1 h_1) \times_2 h_2 \quad (1)$$

Here $\times_i$ denotes the multiplication of respective hidden layer with the i^{th} dimension of the tensor μ of 3rd order [22].

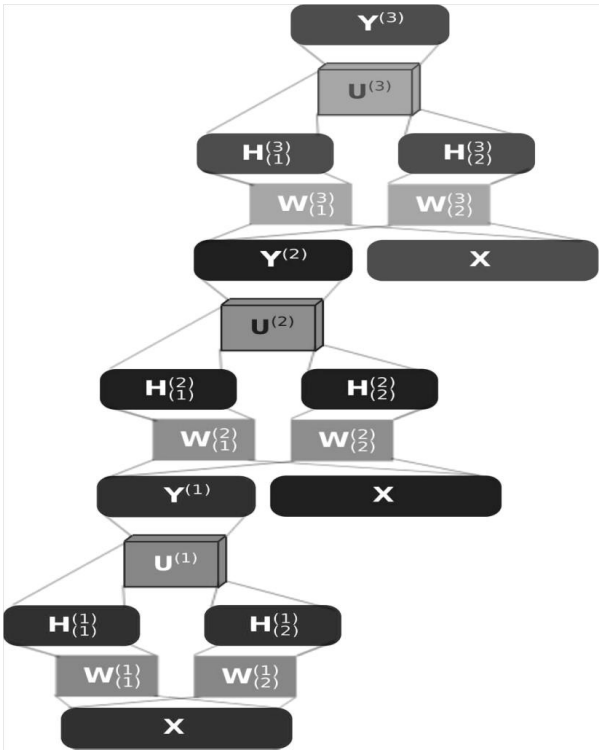

Figure 1.8 Tensor Deep Stacking Network [21]

X represents the input vector, W_i^j represents the weight from input to i^{th} hidden layer of j^{th} block of the architecture, H_i^j represents the i^{th} hidden layer of j^{th} block, U^j represents the 3rd order weight tensor to combine output of two hidden layer for final prediction.

These two parallel hidden layers ($H_{1,1}$ and $H_{1,2}$) will produce two different representations of the input data and a third order tensor (U) in each module is used to produce bilinear mapping of these representations to give a prediction for each module [21].

The concatenation of original input vector X with prediction Y(1) of current layer will guarantee a better generalization in next layer prediction.

1.6 Sound Classification

Sound waves are made up of high pressure and low pressure regions moving through a medium. These high and low pressure regions forms a specific type of pattern to every distinguish sound. These waves have few characteristics like wavelength, frequency, wave speed and time periods [23]. These characteristics are used to classify the sounds into different categories like humans do.

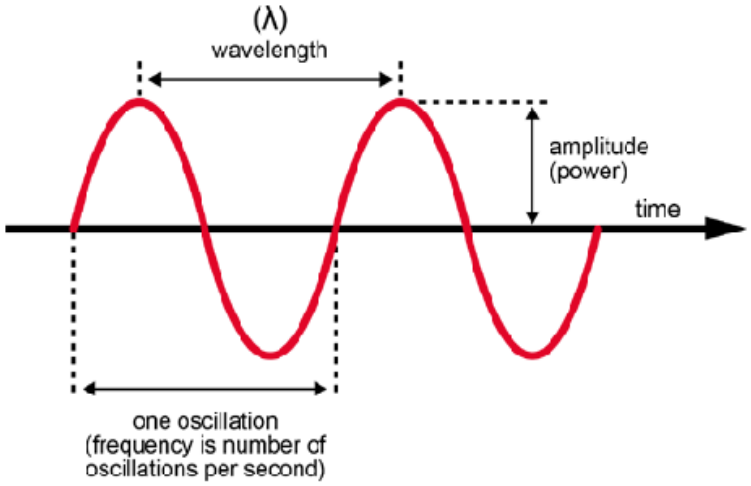

Figure 1.9 Sound Wave Graph [23]

Sound classification is very popular for few applications like automatic speech to text, environmental sound classification, hearing aids, and space surveillance applications.

Sound classification is quite difficult for a machine because a sound wave is not always a pure sound wave, it carries a level of noise or disturbance with it. For a successful classification of these sound waves, noise must be separated from the actual sound.

CHAPTER-2

LITERATURE SURVEY

The literature about the deep learning and various architectures is available in many journal articles and books. Literature is available for last few decades but the literature which is directly related deep learning is found after 2005. For this literature survey many journals, books and conference proceedings are used. Few of the important citations those are very important for understanding, are explained in this chapter.

2.1 Learning Deep Architectures for AI (Y. Bengio, 2009)

Y.Bengio describes the necessity of deep architectures to learn complex tasks like vision and language processing [25]. Shallow architectures are not capable enough to represent the complicated function in an efficient way. Deep architectures are made up of multiple layers of non linear processing units. These processing units try to process data in a sequential manner and try to use less hardware to overcome the problems of parallel computations. This paper explains the working of deep architectures with the help of human brain. Human brain does not always pre process the data but it passes the data from a number of modules organized in a hierarchical manner. This provides the various levels of abstractions or simply saying it gives different representations of this data. Author also explained the limitations of local estimation and local generalizations. Local estimation techniques divide the input space into various regions. In case of very complex function number of required regions for a good generalization, are also very large. This creates a problem in the training of the algorithm. This paper gives an overview of various deep architectures and their learning algorithms. It gives a detailed overview of the deep generative architectures and the method of their

training. It explains the relationship between Deep Belief networks and sigmoid belief networks and also explains the difference in topmost layers of these two networks. It also gives the overview of CNN and few energy based models.

2.2 On the quantitative analysis of deep belief networks (R. Salakhutdinov, et.al, 2008)

R. Salakhutdinov, et.al, describes the deep belief network architecture and the new algorithm for training it. The deep belief network is a probabilistic generative model with multiple hidden variables. In this architecture the hidden layer tries to find a high order correlation with the layer below it or previous layer. The fundamental building block of Deep Belief Network is restricted Boltzmann machine which is a bipartite undirected graphical model [27]. The training of this architecture is done by a different procedure than traditional methods. It uses greedy layer by layer unsupervised learning algorithm [28]. The use of partition function used in RBM needs a very good estimator for controlling the model complexity.

2.3 Multi-column deep neural network for traffic sign classification (D. Cireşan, et. al, 2012)

D. Cireşan, et.al, describes a deep neural network approach that achieved better than human accuracy in object recognition [30]. A simple neural network has one hidden layer for intermediate processing. There is no restriction in adding the hidden layers but going beyond two layers in neural network decreased the computational speed and accuracy of the system. In a deep neural network more than two hidden layers are used to intermediate computations. The use of GPU computing made this possible to add more number of layers. In this approach multiple convolutional and max pooling layers in a hierarchy. The current layer will receive the output of previous as its input. The training in this approach starts with the preprocessing of given data. The preprocessed data is fed into the first convolutional layer. This convolutional layer performs a 2D convolution over the data. After this the data is passed through an activation function, which will be non linear in nature. The max pooling layer will

work as a down sampling layer which with try to reduce the dimensionality of the output of the previous layer. This forms a single block of the DNN. Multiple blocks are arranged in a hierarchy to obtain the final prediction. This approach achieved a whooping accuracy of 99.46% on traffic signal recognition in German Traffic Department.

2.4 Scalable stacking and learning for building deep architectures (L. Deng, et.al, 2012)

L. Deng, et.al, introduced a deep architecture which overcomes the parallel training problem of the deep neural network architecture. DNN architecture is very successful in few applications like object or image recognition but the training of deep neural network with 3 or more layers is very difficult. The problem arises with the use of stochastic gradient descent approach because it is very difficult to parallelize in machine learning. This paper describes a scalable deep architecture for faster computation and with low complexity. The architecture is known as Deep Stacking Network (DSN). The architecture is in the form of modules stacked over one another. The formation of different module is different for every module. The lower module has three layers. The first layer is simple input vector layer, next hidden layer is made up of the sigmoidal activation units. The final layer is prediction layer. The output of current layer is concatenated with the original input which serves as the input to the next module [31]. The fine tuning of the architecture is done in a module by module manner. It reduces the problem of over fitting because there is no global fitting function is used for the whole architecture.

2.5 Understanding the difficulty of training deep feed forward neural networks (X. Glorot, et.al, 2010)

Major problems or difficulties in the training of the Deep Neural Networks are described by the authors in this paper. In last few years various papers proved that the training of DNN is not purely successful. Models those have shown accuracy in training are quite different from the traditional deep neural architectures. Author used MNIST, CIFAR-10 and ImageSet image datasets for the experiment. The MNIST data set used in the experiment contains around 70,000 images of size 28×28 grey scale pixel images. 50,000 were used for training, 10,000 for validation and 10,000 for testing. The focus of this paper is on the selection of activations

and gradients in the classical neural network architectures. This paper describes that the classical architectures of deep neural network provides poor solution and took a great time for converging. The use of sigmoid activation function always results in a poor result and causes saturation in top layer [32].

2.6 Extracting and composing robust features with denoising auto encoders (P. Vincent, et.al, 2008)

P. Vincent, et.al, explains he problems with the intermediate representations of architectures. The intermediate representation of the data plays a huge role in the final prediction label generated by the system. The problem lies in the no restriction or absence of any standard for the middle level representation. Basically a middle level representation should retain certain level of information from the input vector. This paper offers a method for robustness in the case of partially degraded or destroyed images. The architecture used is the Denoising auto encoder architecture. A basic auto encoder takes the input vector X and maps it to the hidden representation Y by using a deterministic mapping technique say Y= f(X). The main difference lies in this result Y. By using this Y one can reconstruct the input vector X. If W is the weight matrix used in input to hidden representation mapping then W* is the weight matrix used in reverse mapping. The relation between W and W* is defined by: $W^* = W^T$

In Denoising Auto Encoders, a noised or corrupted image is used to construct an original image [33]. In this paper, original input vector X was partially destroyed to form X* by using a technique from stochastic mapping. Then this X* will be mapped to Y. The modified auto encoder will try to reconstruct the X i.e., the uncorrupted version from Y by maintaining the minimum average reconstruction error. This approach is quite similar to the training with noise approach used by many other authors.

This approach specifies a technique which can be used to learn representations from the corrupted or partially corrupted data vectors.

2.7 Exploring Strategies for Training Deep Neural Networks (H. Larochelle, et.al, 2009)

Deep architectures are capable to represent the complex highly varying functions in a very compact form and with the use of minimum hardware. Generally in conventional strategies for training weights are randomly initiated and most of the time gradient descent approach using back propagation algorithm is used to update these weights [34]. This approach yields poor results in networks with more than three layers. Major problem of gradient descent approach is that it easily get stuck in poor local minima, which results in poor solutions. The solution to this problem is greedy layer wise training which skips any strategies for whole architecture [28]. The second major problem in gradient descent approach is the highly varying generalization errors means different solutions with great variation from each other. The solution for this problem is the use of regularization. Various unsupervised learning based regularization procedures are available to overcome this problem. Apart from gradient descent approach, adding two many layers in the deep architecture also results in the bad generalization problem due to the problem of over fitting. This paper compares the generative models with encoding models. The generative models try to generate new data based on the representation of data supplied to machine in training phase. But on the other side an encoding model tries to learn a representation of data, by using which they can generate the original input with minimum information loss [34]. This paper suggests that the technique which uses low number of hyper parameters is better over other techniques.

2.8 A novel scheme for speaker recognition using a phonetically-aware deep neural network (Y. Lei, et.al, 2014)

Y. Lei, et.al introduced a new approach for speaker recognition using Deep Neural Networks instead of using conventional GMM method. The ‘i-vector’ extraction paradigm used in this approach is much efficient in speaker recognition as compared to conventional methods. The whole paradigm can be divided into three levels: 1) Collection of speech statistics 2) process of extraction of i-vector from speech and 3) A sophisticated approach known as probabilistic linear discriminant analysis (PLDA) backend [36]. The generated library of speech statistics contains the sequence of feature vectors which forms a universal background model (UBM).

The i-vector contains all the information about the speaker and about all the possibly viabilities in the provided speech segment. PLDA model is used to generate a matching score by comparing the i-vector with stored feature vectors for speaker verification. The i-vector which is used to represent a speech signal is the maximum a posterior (MAP) point estimate of a segment specific standard normal-distributed vector. The UBM used in this approach is represented by GMM, is very successful from few years and it is used in various speaker recognition systems. To train the UBM three datasets were used and these are: NIST SRE, Fisher, and Switchboard. This approach uses DNN to compute the posterior of each frame for every class in the model. After computing the posteriors of the frames, $Zero^{th}$ order and First order statics are generated and these are then used in PLDA to reach a final prediction.

2.9 Face recognition: a convolutional neural-network approach (S.Lawrence, et.al, 1997)

A new type of approach was introduced by author which can be used to recognize face of person in real time and much faster than the conventional techniques. Convolutional neural network was used in this approach and more stress was given to rapid classification. The preprocessing phase was excluded from the approach because this particular phase consumes lots of time in the processing. Moreover rapid classification does not require preprocessing in a very similar manner in which neocortax in mammals does not perform any preprocessing of sensory signals. CNN has three important features: 1) Local Receptive Fields 2) Shared Weights and 3) Spatial sub-sampling The CNN layer is very useful in this approach because it uses multiple planes in each layer to detect the maximum number of features. Sub sampling layer performs an operation of local averaging and sub-sampling.

The system works in a phase wise manner. First of all 5×5 2D Cartesian grid window is placed over the image to create sub samples. Then these 25 input vectors are passed to SOM which creates 125 topologically order values to reduce the dimensionality. Step one is repeated here on the output of SOM to create a large training data set. This training data set is used by the CNN layer for training. The problem of this approach was with the computational time taken by the SOM and CNN for training. This paper attracted many scholars to work on CNN based system to decrease the time by normalizing the data in a very fast way [35].

2.10 Unsupervised learning of hierarchical representations with convolutional deep belief networks (H. Lee, et.al, 2011)

The use of high resolution images for training is very difficult for conventional deep learning techniques. This paper offers and approach which is an extension of DBN and is known as Convolutional Deep Belief Network. This approach allows bottom up and top down probabilistic inference for deep learning. This approach uses probabilistic max pooling technique to scale high resolution images. This approach is based on RBM and DBN, so its construction is quite similar to these architectures. The DBN approach ignores the structure of the image but CDBN weights are shared for less computational load over the system. The architecture of CDBN uses max-pooling CRBMs and these are stacked over each other to form the final system. The training is performed is a similar manner to previous techniques i.e., greedy based layer wise training [28]. Once the training of the current layer is complete, its weights are classified as frozen and next layer uses its output for training.

The difference between this approach and previous approaches lies in its hierarchical probabilistic inference. Suppose a person's face has a excess shadow on one side, this approach will still be able to make an educated guess and mostly the prediction rate is much better to conventional approaches [38].

2.11 Efficient Learning of Deep Boltzmann Machines (R. Salakhutdinov, et. al, 2010)

R. Salakhutdinov, et.al, describes a new type of learning technique for deep Boltzmann machine which is much efficient than conventional techniques. The deep Boltzmann machine is very different from DBN because in DBM the connections between two layers are undirected but this is not the case with deep belief networks. The deep Boltzmann machine contains a set of visible binary units and set of hidden units in a sequence of multiple hidden layers. The conventional training method used in the training of DBM is known as Approximate Maximum Likelihood learning. This method uses Markov chain to calculate

both expectations i.e., data-dependent as well as model's expectation. This method is very much slow and it makes the DBMs not suitable for practical use.

The method of training described by this paper starts with the pre training of the DBM with initial weights. The next step is the variational inference in which recognition model is used to estimate the parameter vector for bottom up pass. Then this parameter vector is used to run the mean field update. The final step is stochastic approximation with is accomplished with the help the help of Gibbs sampler [39]. The use of recognition model to run the mean field fixed point equation, it speeds up the DBM.

2.12 Deep stacking networks for information retrieval (L. Deng, et.al, 2013)

L. Deng, et.al, introduced a different kind of deep architecture which has an advantage of parallel and scalable learning. The architecture is based on the stacked generalization [40]. The Deep Stacking Network (DSN) does not use the stochastic gradient descent approach for learning. As discussed earlier, gradient descent approach has few drawbacks and it makes it very difficult for parallel learning. DSN learning is based on the mean square error between the target prediction value and the current module's prediction value [28].

The architecture of DSN is module based and these modules are stacked over one another to make the final set. In each module after the first or base module the prediction of the previous layer is concatenated with the original input vector. This concatenation always helps in better generalization than the previous layer. In DSN there is a closed form constraint between the input weights and hidden weights.

Tensor-Deep stacking network is an extension of this architecture which works with the help of tensors.

2.13 Tensor Deep Stacking Networks (B. Hutchinson, et.al, 2013)

The foundation of tensor Deep Stacking Network is based on the Deep stacking network discussed earlier. It is not exactly same, the few differences between these two architectures are:

a) Instead of having a single hidden layer in each module, there are two parallel hidden layers in T-DSN.
b) For these two layers there are two weight matrices.
c) Instead of having a weight matrix between hidden to prediction layer, in T-DSN a third order weight tensor is present.

These two parallel hidden layers will provide an ability to capture the higher order feature interaction through the use of cross products.

In tensor notation the operation will be:

$$y = \mu\left(h_{(1)}, h_{(2)}\right) \cong (\mu \times_1 h_1) \times_2 h_2$$

Here $\times_i$ denotes the multiplication of respective hidden layer with the i^{th} dimension of the tensor μ of 3rd order [22].

X represents the input vector, W_i^j represents the weight from input to i^{th} hidden layer of j^{th} block of the architecture, H_i^j represents the i^{th} hidden layer of j^{th} block, U^j represents the 3rd order weight tensor to combine output of two hidden layer for final prediction.

These two parallel hidden layers ($H_{1,1}$ and $H_{1,2}$) will produce two different representations of the input data and a third order tensor (U) in each module is used to produce bilinear mapping of these representations to give a prediction for each module [21].

The concatenation of original input vector X with prediction Y(1) of current layer will guarantee a better generalization in next layer prediction.

CHAPTER-3

PROPOSED WORK

3.1 Problem Formulation

The conventional neural architectures provide a poor solution in the presence of huge data. Going beyond three layers in neural network cause the architecture to stuck with bad generalization and local minima problem. Due to these problems of local minima and bad generalization conventional architecture fails to achieve a required level of accuracy. Deep architectures are performing better in large datasets because of the multiple levels of abstraction. Tensor Deep Stacking Network architecture can be used for classification of sound signals in various applications. Combining HMM and T-DSN with a non linear activation will help an application to give classification in a quicker way. The use of Tensors in deep stacking network will provide consistency to the hidden representations and also it captures the higher order co relation between the different representations of data. The error rate will be reduced through this new approach.

3.2 Scope of Study

Deep learning is a new approach to handle the information in a way similar to the mammal brain. The human brain can represent the information in various levels of abstraction. The same level of abstraction can be constructed in machine using deep learning. In current technology most of the computational time is consumed in pre-processing of data but with the use of deep learning this time can be eliminated. Decision making capabilities are greatly impacted by the way in which our brain links the earlier stored information. Similarly the different levels of abstractions can be combined to create a level of analogy in computers.

The use of tensors in deep learning has greater possibility for achieving the high level of abstraction and fast computation. The power or property of tensors to arrange them in order, that for every observer the representation is same, can be used to handle the huge amount of data in a consistent way.

The use of T-DSN enables the machine to capture the higher order feature interaction through the use of cross products. This interaction is very useful for creating state of art architecture for decision making.

3.3 Objectives

1. To generate a performance curve of neural network with increasing amount of training and testing data.

2. To generate a performance curve of neural network with increasing number of hidden layers. This objective will show the effect of adding new hidden layers in neural network

3. To compare the performance of conventional neural network with Deep learning architecture. This objective will justify the use of deep architectures for huge volume of data.

4. Sound Classification using Tensor-Deep Stacking Network with Hidden Markov Model and non linear activation. This objective will be achieved by combining the traditionally available model for sound classification with a new approach of tensor deep stacking network.

3.4 Research Methodology

The research is based on the sound classification with the help of deep learning architecture. The dataset selected for the research is The Urban Sound dataset. It contains 8732 labeled sound from 10 classes. This data set will be divided into 3 parts: 60% data will be used for training the network, 20% for validation and 20% for testing. First of all the conventional neural network will be used to achieve the objective 1 to 3. The amount of data will changed accordingly to generate a performance curve. Then the same dataset will be used on deep architecture and its performance curve will be compared with conventional architecture. This part of research will be qualitative in nature.

The quantitative research method will be used to check the performance of T-DSN Architecture with HMM and non linear activation. In this part Data will be again divided into parts. T-DSN will be used then the result of this will be used by HMM and a non linear activation to generate a final prediction.

3.5 Expected Outcomes

Following are the various expected outcomes for this research:

1) A new architectural approach for classifying the sound signals into various categories. This approach will be based on Tensor Deep Stacking Network, Hidden Markov Model and a Non Linear Activation function for final prediction.

2) A well explained and experimental comparison of Conventional Neural network with deep architecture.

3) Experimental result of the effect of increasing amount of data and hidden layers on the performance of conventional neural network.

Conclusion

This proposed work describes a new approach for sound classification using T-DSN with HMM and activation like softmax. The training of the T-DSN will be based on MSE between the target value and the current module prediction value. The error rate and accuracy of the proposed approach in training and testing dataset will used to achieve the objectives. Chapter 1, discussing few definitions and working for complete overview about the deep architectures and their comparison with conventional architectures. Chapter 2 provides a literature survey of important research papers, those are very important for this research and to understand the working of deep architectures. Chapter 3 provides a insight into the this research. It covers the problem definition, methodology, expected outcomes of the research.

REFERENCES

[1] R. Kohavi and F. Provost, "Glossary of Terms", *Machine Learning*, vol. 30, no. 2-3, pp. 271- 274, 1998.

[2] J. Mueller and L. Massaron, *Machine Learning For Dummies*, 1st ed. 2016, pp. 40-43.

[3] M. Mohri, A. Rostamizadeh and A. Talwalkar, *Foundations of Machine Learning*, 2nd ed. The MIT Press, 2012, pp. 101-105.

[4] S. Geman, E. Bienenstock and R. Doursat, "Neural Networks and the Bias/Variance Dilemma", *Neural Computation*, vol. 4, no. 1, pp. 1-58, 1992.

[5] T. Hastie, R. Tibshirani and J. Friedman, *The elements of statistical learning*. . [6] R. Acharyya, *A new approach for blind source separation of convolutive sources*. Saarbrücken: VDM, Verlag Dr. MüĬler, 2008.

[7] L. Kaelbling, M. Littman and A. Moore, "Reinforcement learning: a survey", *J. Artif. Int. Res*, vol. 4, no. 1, pp. 237-285, 1996.

[8] [4]M. Wiering and M. Otterlo, *Reinforcement learning*. Heidelberg: Springer, 2012.

[9] Y. LeCun, Y. Bengio and G. Hinton, "Deep learning", *Nature*, vol. 521, no. 7553, pp. 436-444, 2015.

[10] J. Dean, G. Corrado, R. Monga and A. Ng, "Large Scale Distributed Deep Networks", in *Advances in Neural Information Processing Systems 25*, 2012, pp. 1232--1240.

[11] L. Deng, "Deep Learning: Methods and Applications", *Foundations and Trends® in Signal Processing*, vol. 7, no. 3-4, pp. 197-387, 2014.

[12] "Why Deep Learning Is Suddenly Changing Your Life", *Fortune*, 2017. [Online]. Available: http://fortune.com/ai-artificial-intelligence-deep-machine-learning/. [Accessed: 20- Nov- 2017].

[13] S. Wang, W. Chaovalitwongse and R. Babuska, "Machine Learning Algorithms in Bipedal Robot Control", *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, vol. 42, no. 5, pp. 728-743, 2012.

[14] J. Brownlee, "What is Deep Learning? - Machine Learning Mastery", *Machine Learning Mastery*, 2017. [Online]. Available: https://machinelearningmastery.com/what-is-deep-learning/. [Accessed: 20- Nov- 2017].

[15] Y. Bengio, *Learning deep architectures for AI*. Hanover, Mass.: Now Publishers, 2009.

[16] J. Niebles and L. Fei-Fei, "A hierarchical model of shape and appearancefor human action classification," in *Proceedings of the Computer Vision andPattern Recognition Conference (CVPR'07)*, 2007.

[17] I. Murray and R. Salakhutdinov, "Evaluating probabilities under high dimensional latent variable models," in *Advances in Neural Information Processing Systems 21 (NIPS'08),* vol. 21, (D. Koller, D. Schuurmans, Y. Bengio, and L. Bottou, eds.), pp. 1137–1144, 2009.

[18] N. Pinto, J. DiCarlo, and D. Cox, "Establishing good benchmarks and baselines for face recognition," in *ECCV 2008 Faces in 'Real-Life' Images Workshop*, 2008.

[19] *Learning in Graphical Models*. Dordrecht: Springer Netherlands, 1998.

[20] L. Deng, X. He and J. Gao, "Deep stacking networks for information retrieval", *2013 IEEE International Conference on Acoustics, Speech and Signal Processing*, 2013.

[21] B. Hutchinson, L. Deng and D. Yu, "Tensor Deep Stacking Networks", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 35, no. 8, pp. 1944-1957, 2013.

[22] T. Kolda and B. Bader, "Tensor Decompositions and Applications", *SIAM Review*, vol. 51, no. 3, pp. 455-500, 2009.

[23] "Sound classification - Paroc.com", *Paroc.com*, 2017. [Online]. Available: http://www.paroc.com/knowhow/sound/sound-classification. [Accessed: 23- Nov- 2017].

[24] T. Lee, D. Mumford, R. Romero and V. Lamme, "The role of the primary visual cortex in higher level vision", *Vision Research*, vol. 38, no. 15-16, pp. 2429-2454, 1998.

[25] Y. Bengio, "Learning Deep Architectures for AI", *Foundations and Trends® in Machine Learning*, vol. 2, no. 1, pp. 1-127, 2009.

[26] J. Schmidhuber, "Deep learning in neural networks: An overview", *Neural Networks*, vol. 61, pp. 85-117, 2015.

[27] R. Salakhutdinov and I. Murray, "On the quantitative analysis of deep belief networks", *Proceedings of the 25th international conference on Machine learning - ICML '08*, 2008.

[28] G. Hinton, S. Osindero and Y. Teh, "A Fast Learning Algorithm for Deep Belief Nets", *Neural Computation*, vol. 18, no. 7, pp. 1527-1554, 2006.

[29] N. Le Roux and Y. Bengio, "Representational Power of Restricted Boltzmann Machines and Deep Belief Networks", *Neural Computation*, vol. 20, no. 6, pp. 1631-1649, 2008.

[30] D. Cireşan, U. Meier, J. Masci and J. Schmidhuber, "Multi-column deep neural network for traffic sign classification", *Neural Networks*, vol. 32, pp. 333-338, 2012.

[31] L. Deng, D. Yu and J. Platt, "Scalable stacking and learning for building deep architectures", *2012 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2012.

[32] X. Glorot and Y. Bengio, "Understanding the difficulty of training deep feedforward neural networks", in *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics (AISTATS-10)*, 2010, pp. 249-256.

[33] P. Vincent, H. Larochelle, Y. Bengio and P. Manzagol, "Extracting and composing robust features with denoising autoencoders", *Proceedings of the 25th international conference on Machine learning - ICML '08*, 2008.

[34] H. Larochelle, Y. Bengio, J. Louradour and P. Lamblin, "Exploring Strategies for Training Deep Neural Networks", in *Journal of Machine Learning Research*, 2009, pp. 1-40.

[35] S. Lawrence, C. Giles, Ah Chung Tsoi and A. Back, "Face recognition: a convolutional neural-network approach", *IEEE Transactions on Neural Networks*, vol. 8, no. 1, pp. 98-113, 1997.

[36] Y. Lei, N. Scheffer, L. Ferrer and M. McLaren, "A novel scheme for speaker recognition using a phonetically-aware deep neural network", *2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2014.

[37]A. Courville, J. Bergstra and Y. Bengio, "A Spike and Slab Restricted Boltzmann Machine", in *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics (AISTATS-11)*, 2011, pp. 233-241.

[38] H. Lee, R. Grosse, R. Ranganath and A. Ng, "Unsupervised learning of hierarchical representations with convolutional deep belief networks", *Communications of the ACM*, vol. 54, no. 10, p. 95, 2011.

[39] R. Salakhutdinov and H. Larochelle, "Efficient Learning of Deep Boltzmann Machines", in *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics (AISTATS-10)*, 2010, pp. 693-700.

[40] D. Wolpert, "Stacked generalization", *Neural Networks*, vol. 5, no. 2, pp. 241-259, 1992.