SPOKEN ARABIC DIGITS RECOGNITION

USING DEEP LEARNING

ABDULAZIZ SALEH MAHFOUDH BA WAZIR

FACULTY OF ENGINEERING

UNIVERSITY OF MALAYA

KUALA LUMPUR

2018

University of Malaya

SPOKEN ARABIC DIGITS RECOGNITION

USING DEEP LEARNING

ABDULAZIZ SALEH MAHFOUDH BA WAZIR

DISSERTATION SUBMITTED IN FULFILMENT OF

THE REQUIREMENTS FOR THE

DEGREE OF MASTER OF ENGINEERING

(MECHATRONICS)

2018

University of Malaya

UNIVERSITY OF MALAYA

ORIGINAL LITERARY WORK DECLARATION

Name of Candidate: **Abdulaziz Saleh Mahfoudh Ba Wazir**
Matric No: **KQF160016**
Name of Degree: **Master Degree of Engineering (Mechatronics)**
Title of Project Paper/Research Report/Dissertation/Thesis ("this Work"): **Spoken Arabic Digits Recognition Using Deep Learning**
Field of Study:
Artificial Intelligence and Deep Learning

I do solemnly and sincerely declare that:

(1) I am the sole author/writer of this Work;

(2) This Work is original;

(3) Any use of any work in which copyright exists was done by way of fair dealing and for permitted purposes and any excerpt or extract from, or reference to or reproduction of any copyright work has been disclosed expressly and sufficiently and the title of the Work and its authorship have been acknowledged in this Work;

(4) I do not have any actual knowledge, nor do I ought reasonably to know that the making of this work constitutes an infringement of any copyright work;

(5) I hereby assign all and every right in the copyright to this Work to the University of Malaya ("UM"), who henceforth shall be owner of the copyright in this Work and that any reproduction or use in any form or by any means whatsoever is prohibited without the written consent of UM having been first had and obtained;

(6) I am fully aware that if in the course of making this Work I have infringed any copyright whether intentionally or otherwise, I may be subject to legal action or any other action as may be determined by UM.

Candidate's Signature Date:

Subscribed and solemnly declared before,

Witness's Signature Date:

Name:

Designation:

University of Malaya

ABSTRACT

The dissertation proposes an Arabic digits speech recognition model utilizing recurrent neural network. Speech Recognition model select the finest speech signal representation by feature extraction of Mel-Frequency Cepstrum Coefficients (MFCCs) after been processed for noise reduction and digits seperation. Digit speeches extracted features are fed into a network with long short-term memory (LSTM) cells. The LSTM cells have the capability to solve problems associated with temporal dependencies and require learning long-term and solve the vanishing gradient problems associated with RNN. A dataset of 1040 samples of spoken Arabic digits from different dialects is used in this study where 840 samples used to train the network and another 200 samples are used for testing purpose. The model training is carried out using GPU. The LSTM model learning parameters is tuned for optimization purpose to achieve higher accuracy of 94% during model training. The testing results of the finest tuned parameters model shows that the LSTM model is 69% accurate in recognizing spoken Arabic digits samples. Model highest accuracy obtained when recognizing the digit zero with 80%.

University of Malaya

ABSTRAK

Tesis ini mencadangkan penggunaan sebuah model yang menggunakan rangkaian neural yang berulang dalam mengenalpasti sebutan angka Arab di dalam percakapan. Model pengenal pastian ini akan mengenalpasti angka dengan menggunakan Mel-Frequency Cepstrum Coefficients (MFCCs) setelah diproses bagi mengurangkan bunyi dan mengasingkan digit yang kemudiannya akan dimasukkan ke dalam rangkaian yang dilengkapi dengan Long Short-Term Memory (LSTM). Sel-sel LSTM mempunyai kebolehan untuk menyelesaikan masalah yang bergantung kepada masa dan memerlukan masa untuk difahami dan masalah yang berkaitan dengan RNN. Set data sebanyak 1040 sampel sebutan angka Arab dalam dialek yang berbeza telah digunakan, 840 sampel digunakan untuk melatih rangkaian manakala 200 sampel untuk tujuan menguji model. Model ini telah dilatih menggunakan GPU. LSTM telah dilaraskan untuk mencapai tahap ketepatan yang optimum iaitu 94% ketika melatih model. Keputusan yang direkodkan menunjukkan model LSTM mempunyai ketepatan sebanyak 69% dalam mengenalpasti sampel sebutan angka Arab. Model telah mencapai tahap ketepatan yang paling tinggi sebanyak 80% dalam mengenal pasti angka sifar.

University of Malaya

ACKNOWLEDGEMENT

Thank you to Allah Almighty for enabling me to complete this research project report.

This project would have not been complete without the efforts of many individuals and organization who have helped me in pursuing my goals and objectives for this project.

Firstly, I would like to express a special gratitude to my supervisor for this project IR. DR. Chuah Joon Huang for his encouragement to proceed with this proposed topic and accepting me as a student, in VIP Research Laboratory, to pursue this project with his guidance and sincere efforts in helping me aquire knowledge and discussing every phase of this project on what can be achieved.

I would like to also thank my friends Mazen Baabbad, Hafedh Almashgari, Hesham Algariri, Abdullah Alaameri, Muhsin Abdul Mahmud for their support, guidance and assists for this project.

Finally, I would like to thank my parents and brothers who have supported me throughout my journey, without whom I would have not reached this position of my life and would not have been able to do my research project for my masters degree.

University of Malaya

Table of Contents

University of Malaya

University of Malaya

University of Malaya

LIST OF FIGURES

University of Malaya

University of Malaya

LIST OF TABLES

University of Malaya

LIST OF ABBREVIATIONS

Adam: Adaptive Moment estimation

ANN: Artificial Neural Networks

ASR: Automatic Speech Recognition

CNN: Convolutional Neural Network

CPU: Central Processing Units

DCT: Discrete Cosine Transform

DNN: Deep Neural Network

FFT: Fast Fourier Transform

FPGA: Field Programmable Gate Arrays

GPU: Graphical Processing Unit

GMM: Gaussian Mixture Model

HMM: Hidden Markov Model

LSTM: Long Short-Term Memory

LTM: long term memory

MFCCs: Mel-frequency Cepstral Coefficients

MLP: Multilayer Perceptron

MSE: Mean Square Error

OCR: optical character recognition

ReLU: Rectified Linear Unit

RNN: Recurrent Neural Network

University of Malaya

CHAPTER 1: INTRODUCTION

This chapter introduces the preliminary background for the research work done for this thesis, problem overview, research objectives, and the scope of study. The preliminaries and the groundwork for this research are highlighted in brief.

1.1 Background

Speech has always been the primary mode of communication between humans, this is due to large information that can be transferred by the speaker to the listener in a short amount of time. With the recent advancements, this mode of speech communication is being used to interact between humans and machines with great advancements such as the Alphabet's "Google Assistant" known as "Google Now" previously, Apple's "Siri" and Amazon's "Alexa" where These technologies have generated a huge impact on the industry in such as home automation, handheld devices, content captioning for videos and hands-free devices in automotive (Stenman, 2015).

Speech recognition has been developed through research works over 50 years of advancements. It has been improving drastically from the implementations of basic word recognition to automatic speech recognition that uses continuous speech in the era of Deep Neural Networks (Halageri, Bidappa, Arjun, Sarathy, & Sultana, 2015).

Deep Neural Network (DNN) is one of the most dominant methods of speech analyzing due to its advantage of minimizing the error rate and optimization problems. DNNs have gained tremendous success due to the ability of training large vocabulary size and their high accuracy of detecting words (Graves, Mohamed, & Hinton, 2013). Deep

University of Malaya

Learning is used in many applications today, these applications include self-driving vehicles, image recognition, and artificial intelligence. This is highly due to the reliability of this method in applications providing excellent outcomes.

1.2 Problem Statement

Arabic is the 5th most spoken language in the globe. However, in terms of research regarding Arabic speech recognition systems is rare while speech recognition is used commonly in this era, with handheld devices, home automation systems, and many more that makes speech recognition is essential in this era. A few methods have been applied include Gaussian Mixture Model (GMM) and Hidden Markov Model (HMM). However, with recent developments in research it is found that the most accurate method of speech recognition is by implementing the use of Deep Neural Networks (DNN) (Deng & Platt, 2014). Hence, Arabic language speech recognizer is needed.

1.3 Objectives of Research

The main objectives of this research project are:

1. To develop a deep learning-based speech recognition model for the Arabic language.
2. To evaluate the parameters to be used to train the developed model.
3. To evaluate the developed model performance in terms of the classification accuracy.

1.4 Scope of Study

The scope of this research project is limited to the design and training of Arabic digits voice recognition using deep learning method of Recurrent Neural Network (RNN), specifically Long Short-Term Memory (LSTM) in Python and it involves the use of Google's TensorFlow library based on the use of GPU. The analysis is based on systems

University of Malaya

ability to recognize and differentiate the different type of Arabic digits speeches. A reasonable number of recorded Arabic digits voices are used as this project dataset.

CHAPTER 2: LITERATURE REVIEW

2.1 Introduction

This chapter presents a review on automatic speech recognition, looking into the acoustic model, signal processing, previous methods of speech recognition, etc. Furthermore, this chapter discusses about DNNs and learning types, and the hardware required for deep learning.

2.2 Automatic Speech Recognition

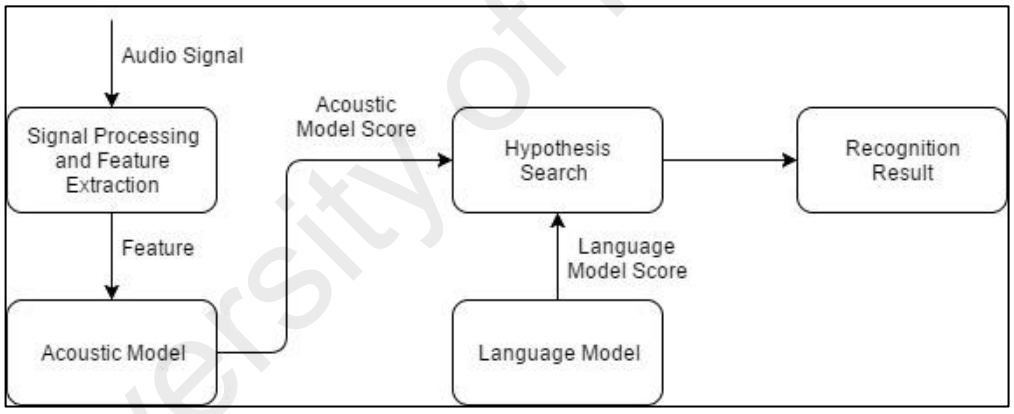

Figure 2.1: Automatic Speech Recognition System Blocks (Huang, Ariki, & Jack, 1990)

The basic architecture of an Automatic Speech Recognition System (ASR) is illustrated in Figure 2.1, involving 4 major stages that are signal processing and feature extraction stage which takes an input of audio signals, filtering out noise and distortions in the signal and converts the signal into frequency domain. In acoustic model stage, information of phonetic and acoustic are utilized to generate an acoustic model score. The language model stage consists of structure of language and words which is used to estimate the word. The hypothesis search block considers the two scores provided by the

University of Malaya

acoustic model and the language model with hypothesis search block can recognize the word and produce the final output of the recognition (Huang et al., 1990).

2.3 Signal Processing

Voice signal is an analog signal. A high accuracy of voice recognition system is necessary. Therefore, the input to the system is required to go through some processing methods, this processing includes sampling and converting and then noise removal is conducted to generate a clear signal, as well as the right representation for the signal feature extraction this can be achieved either by Mel-frequency Cepstral Coefficients (MFCCs) or spectrograms (Muda, Begam, & Elamvazuthi, 2010).

2.3.1 Noise Removal

Muda et al., (2010) stated that voice recorded samples can be represented in a waveform that are the visualization of the signal with respect to time. The signal amplitudes depend on how loud the voice is. The amplitude is high on loud voice samples and low on quiet voice samples. Figure 2.2 shows an example of these waveforms contain noise. This representation is nearer representation to how the human ears will notice the voice sample.

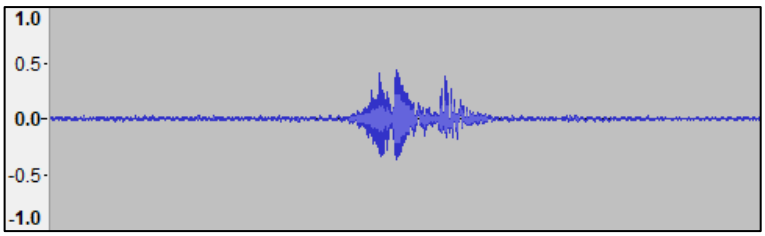

Figure 2.2: Waveform of a Sample Pronouncing Zero Consisting of Noise

The waveform can be represented on logarithmic scale, the logarithmic scale can help visualize the voice sample more accurately, as they sense quiet voice samples such

University of Malaya

as noise more accurately (Ko, Peddinti, Povey, & Khudanpur, 2015). Figure 2.3 shows a representation of logarithmic scale of a voice sample.

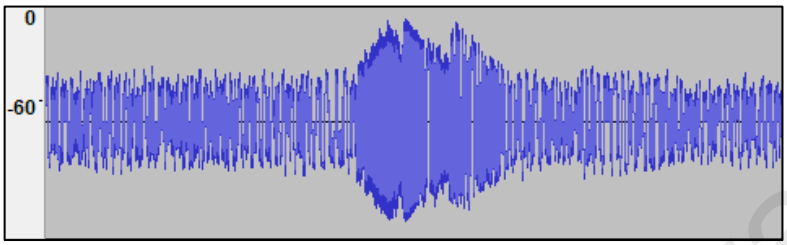

Figure 2.3: Waveform Representation in Decibels in Respect to Time

The voice representation using logarithmic scale in Figure 2.3 helps to visualize the noise even though it can not be heard in the voice sample clearly, however it exists, and may affect the representation of the signal in the system (Rebai, Benayed, Mahdi, & Lorré, 2017). Therefore, the voice samples require a removal of this noise to obtain an accurate system. Noise removal is one of the voice signals processing methods. It can be achieved using many different tools like MATLAB, Pyaudio and Audacity. Figure 2.4 represents the waveform after noise removal.

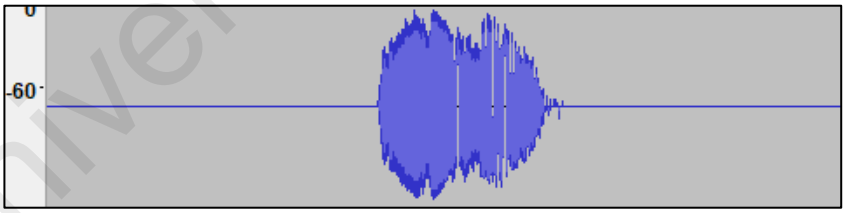

Figure 2.4: Waveform Representation in Decibels After Noise Removal

2.3.2 Data Augmentation

According to (Ko et al., 2015), data augmentation is a strategy deployed to increase the dataset quantity. It is a key strategy for the state of the systems like speech recognition and image processing. Obtaining a large dataset is often a method of increasing accuracy with voice recognition systems especially for Deep Neural Networks.

University of Malaya

However, when a large dataset could not be acquired, the obtained dataset can be modified into increasing the dataset size.

The human voice differs based on the pitch and loudness of the voice. If these two can be re-tuned on a dataset, a larger data set can be obtained. The pitch is proportional to the frequency, therefore a change in frequency will change the pitch of the dataset. The dataset can further be modified by amplifying the dataset. The voice data can be augmented using many techniques like tempo perturbation and speed perturbation and so on (Rebai et al., 2017).

2.3.3 Spectrogram

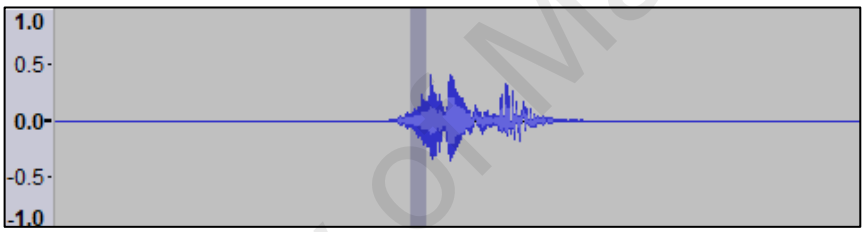

Figure 2.5: A Selection of Small Window from voice Waveform

According to Rebai et al., (2017), spectrograms are an audio signal representation of in the frequency domain. However, instead of the complete signal being represented in frequency domain, a small window generally about 20 ms of the sample is taken as shown in Figure 2.5, where this signal is then represented in frequency domain by performing a Fast Fourier Transform (FFT) onto the signal. Then, the log of the power is given by:

$$\log |FFT(X)|^2$$

This represents the amplitude of the sin wave on the frequency domain, which is later represented as a vector. Figure 2.6 shows the representation of a signal from frequency domain to a frame of the spectrogram. This procedure is repeated throughout the sample to obtain a complete spectrogram of the complete waveform as this procedure

University of Malaya

is part of the signal processing for feature extraction prior to using it in such voice recognition systems (Abraham, 2013).

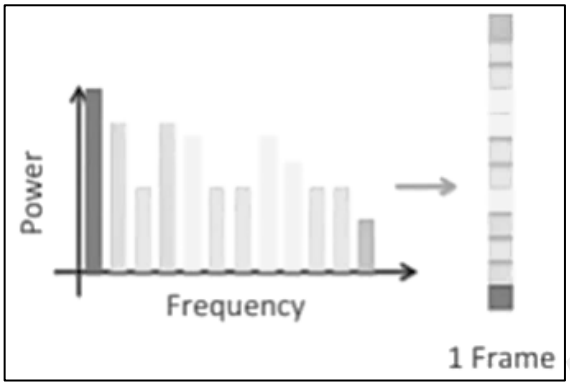

Figure 2.6: Representation of a 20ms Sample to a Slice of Spectrogram (Vyas, 2013)

2.3.4 Mel-Frequency Cepstral Coefficients

Mel-Frequency Cepstral Coefficients is a method of feature extraction that has a similar idea to the one of the spectrograms. the MFCCs conduct a FFT of a window of the signal and take the power of it, using it on to triangular overlapping windows that are shown in Figure 2.7 as the windowed power spectrum, these triangular overlapping windows are a set of 20 – 40 overlapping windows (26 is standard) (Lafta & Yousif, 2015).

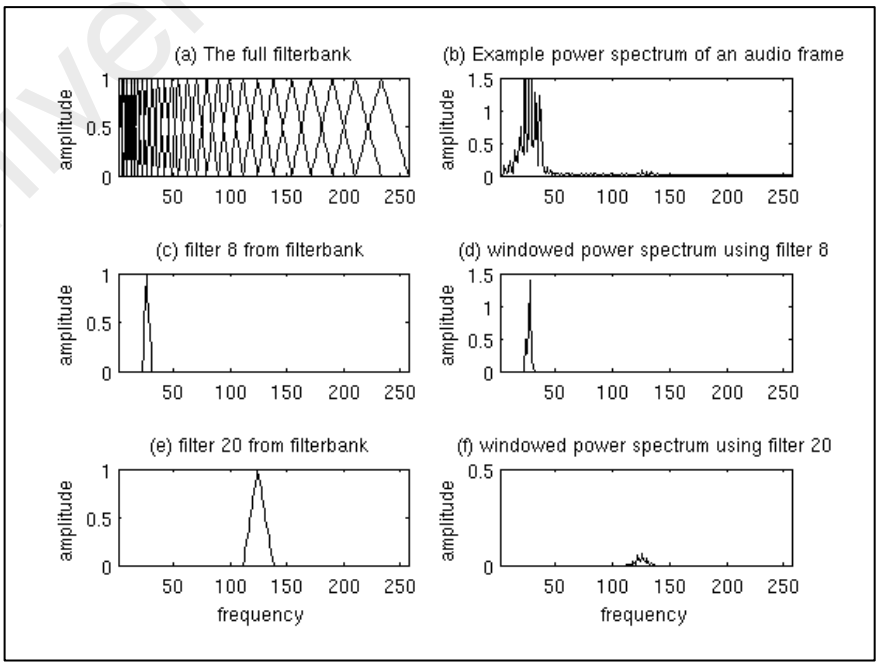

Figure 2.7: MFCC Implemented by Different Windows (Muda et al., 2010)

University of Malaya

Once the output is generated by the windows, a log is taken of the filterbanks, which yields the output of filterbank energies, once this is obtained a Discrete Cosine Transform (DCT) is preformed to the output, which generates the MFCCs (Muda et al., 2010).

2.4 Acoustic Modelling

English language consists of many phonemes. Phonemes are general tones with which the words are pronounced. Phonemes are often different from the written text, for example the word “CAT” has three phonemes “K” “A” “T”. It is possible to create any word from the dictionary if the phonemes are obtained. However, this implementation cannot be used always as an example if we take the word “night” and the word “knight” then it is required to understand the context in which the word is being used, for those cases, the language model is required (Rebai et al., 2017).

Based on Ankit et al., (2016), acoustic model has been used in ASR to formulate a relation between language phonemes an audio signal , or any other linguistic units that represent any spoken words while the language model role is model the sequence of word. The two models are used together to produce top-ranked word sequences in correspondent to any given audio segment. There are several developed acoustic models throughout more than 50 years of research like Hidden Markov Model.

2.4.1 Hidden Markov Model

Hidden Markov Model (HMM) models are statistical models in which it evaluates probability of observations. The main elements to create HMM is the number of states where each phenome is generally given 3 states that consist of a feature vector that can determine the word that is being pronounced (Chakraborty & Talukdar, 2016).

University of Malaya

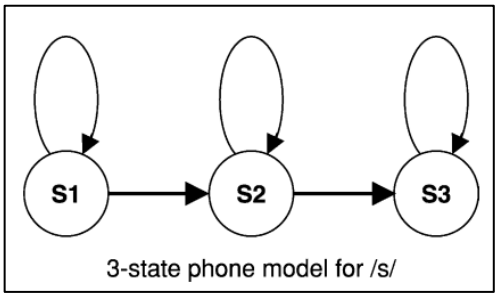

Figure 2.8: Three-state HMM for the phenome “s” (Levinson, 1986)

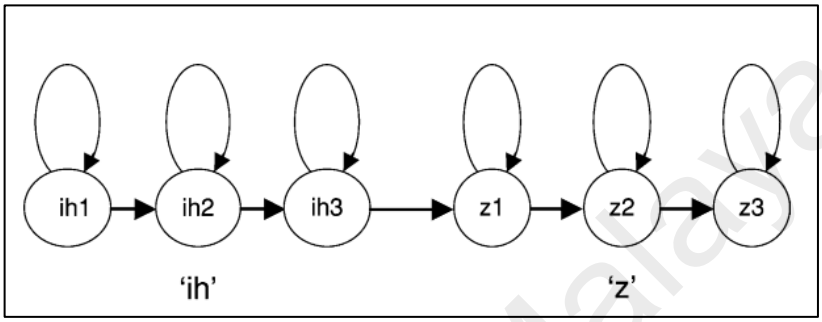

Figure 2.9: Concatenated model for the word “is” with the phenoms “ih” and “z” (Levinson, 1986)

HMM is a strong statistical method for speech recognition thanks to its ability to handle input of variable length. Additionally, it is an efficient learning algorithm that can take place directly from raw data. However, HMM successive observations are independent (Djemili, Bedda, & Bourouba, 2004).

According to Chakraborty & Talukdar, (2016), HMM is based on Markov property. It states that the probability of being in each state at time ‘t’ depends on the state at time ‘t-1’. However, dependencies extend through a few states in speech sound which considered as limitation of HMM model.

2.5 Deep Neural Networks

Initially, Neural Networks (NN) were widely known as Artificial Neural Networks (ANN). A structure like the implementation of biological neurons, with a learning structure based on probability that receives input data to decide on an output. Input data includes texts, images and audio files. Deep Learning utilizes artificial neurons

University of Malaya

to serve high dimensional data. Deep Learning can perform tasks involving communication pattern and information processing (Vallejo, Isaza, & Lopez, 2013).

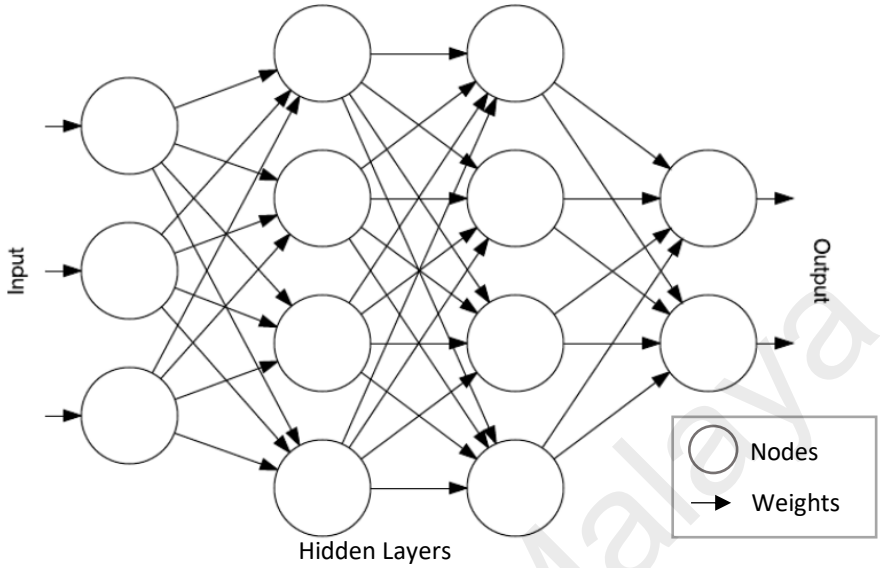

Figure 2.10: Structure of Neural Networks (Dey & Learning, 2016)

Figure 2.10 illustrates the basic structure of NN, with the first layer named as the input and the last as the output with the middle layers being the hidden layers hence the name “deep”. With further research, structure was improved with the implementation of backpropagation. The backpropagation algorithm has two parts, the first is for the input to forward the data towards the output, the second is to evaluate and estimate the error and backpropagate error values to the neural networks for error correction (Dey & Learning, 2016).

Each node in is a feature extracted from the portion before. The number of neurons on each layer vary for each system, as the number of neuron on each layer must be able to capture the essential layer. The first layer models are generally high in number of neurons (Sharma & Kumar, 2017).

Deep learning sets a whole new area of research with the architecture that has fixed many issues that were in the traditional systems of HMM where the model was used

University of Malaya

for specific usage or particular commands. DNNs were able to be a more uniform model that would be applied on many applications as well as different users. The architecture of DNNs can be implemented on several applications, such as the Google Assistant that can be used on mobile phones (Android), also can be used in search engines (Google.com) as well as being implemented on household devices (Google Home), and also used in video captioning such as the video captioning on YouTube (Chakraborty & Talukdar, 2016).

2.5.1 Hardware requirement for Deep Learning

Deep learning models require a high computational power. Thus, the implementation of deep learning requires either of two specific hardware:

a) Field Programmable Gate Arrays (FPGA)

FPGAs offer great real-time capabilities and require less power for usage. Thangavelautham, (2017) claims that FPGAs are a compelling solution and are emerging as a competitor to the currently used architecture of GPU.

b) Graphical Processing Units (GPU)

GPU is a more rapid in terms of the implementation of deep learning than Central Processing Units (CPU). In a research done by Chen, Wang, He, & Huang, (2014) concludes by estimating a 7-11 times better performance done by the GPU than the CPU and that it is a cost-effective method. GPUs are used instead of FPGAs and CPUs, as they offer an easy to tweak system.

2.6 Deep Networks Architecture

Deep network has different types that differ on its architecture. Kim, (2014) stated that network architecture associated with determining the accuracy and speed of the result based on different types of dataset. Various architectures have been developed to carry out certain applications such as Recurrent Neural Network (RNN) and Convolutional Neural Network (CNN).

University of Malaya

2.6.1 Recurrent Neural Network

RNN performs based on the principle of saving layer output of a layer to feed it back to the input or previous layer. That helps to predict the layer outcome where the first layer in the network formed as feed forward networks with the product of weight's sum and features. RNN process applied as each neuron will remember information that had it in its previous time step. This feature of RNN implies that each of the neurons perform like a memory cell while performing computations (Mikolov, Karafiát, Burget, Cernocký, & Khudanpur, 2010).

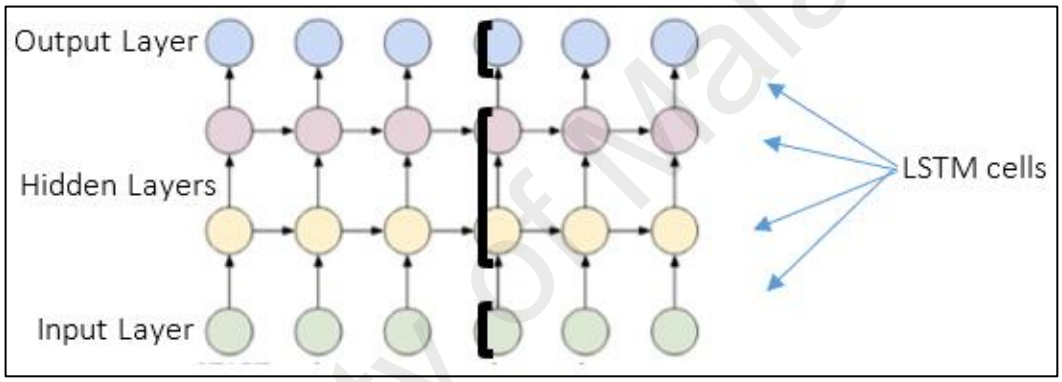

Figure 2.11: Recurrent Neural Network structure (Zeyer, Doetsch, Voigtlaender, Schlüter, & Ney, 2017)

Given an input sequence x = (x1, x2, …, x(t)), a standard RNN computes the hidden vector sequence h = (h1, h2,…, h(t)), and output vector sequence y = (y1, y2, …, y(t)) by iterating the following equations from t = 1 to T:

$$h_t = H(W_{xh}x_t + W_{hh}h_{t-1} + b_h)$$

$$y_t = W_{hy}h_t + b_y$$

where

W = weight matrices

b = bias vectors

H = hidden layer function

University of Malaya

Long short-term Memory (LSTM) is a type of RNNs which solves the issue that is generally common in RNN as it can not solve the type of problems that require the learning of long-term temporal dependencies. This is due to the exponential decay of the gradient of loss function that is called the vanishing gradient problem. LSTM is not a standard RNN where it uses special units in addition to the standard ones. LSTM units contains memory cells that is able to maintain information in memory for long period of time. LSTM has become the most common solving method for vanishing gradient problem and it has the ability to remember values for a short of long time period (Lipton, Kale, Elkan, & Wetzel, 2015).

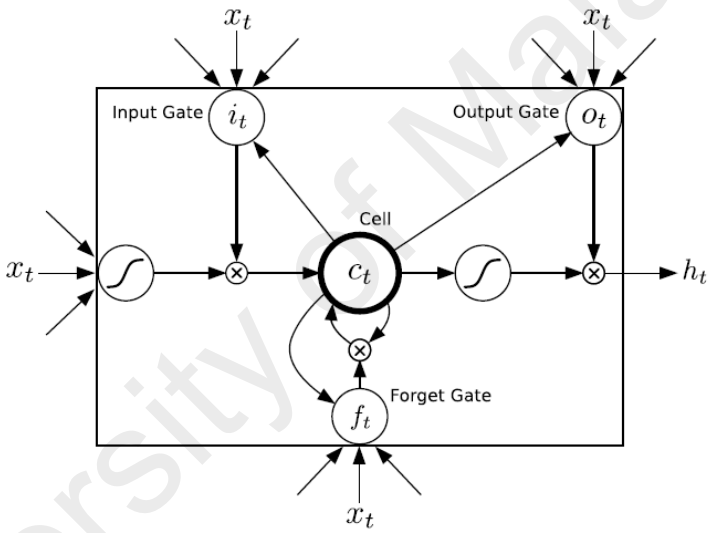

Figure 2.12: LSTM memory cell (Sak, Senior, & Beaufays, 2014)

Figure 2.12 represents the LSTM network architecture and a memory cell. The memory cell consists of several gates to control the cell input and output and manage the flow information within the LSTM network. The three gates are input, output, and the forget gates. Each gate has its own characteristics. The input gate controls the information that are given to the cell, while the forget gate consists of the time required for the information to be retained, and lastly the output gate control when the information is to be passed on as an output to other cells. This is all conducted while also consisting DNN

University of Malaya

probability weights. LSTM perform miraculously well for applications such as speech recognition (Fachrie & Harjoko, 2015).

According to Beaufays, Sak, & Senior, (2014), the output of the cell (h) is determined by

$$i_t = \sigma (W_{xi}x_t + W_{hi}h_{t-1} + W_{ci}c_{t-1} + b_i)$$
$$f_t = \sigma (W_{xf}x_t + W_{hf}h_{t-1} + W_{cf}c_{t-1} + b_f)$$
$$c_t = f_t c_{t-1} + i_t \tanh (W_{xc}x_t + W_{hc}h_{t-1} + b_c)$$
$$o_t = \sigma (W_{xo}x_t + W_{ho}h_{t-1} + W_{co}c_t + b_o)$$
$$h_t = o_t \tanh(c_t)$$

where

σ = logistic sigmoid function

i = input gate

f = forget gate

o = output gate

c = cell activation vectors

2.6.2 Convolutional Neural Network

Convolutional neural networks (CNN) are like feed-forward neural networks, where each neuron has learn-able biases weights. All the neuron in a layer will receives input, execute dot products and maybe be followed by a non-linearity. CNN is made up of one several layers of convolutional and fully connected layers. In between of the convolutional layers, usually there will be a subsampling or pooling layer. It is used mostly in the applications of signal and image processing (Haoxiang & Lin, 2015).

Convolutional networks architecture design is best to perform recognition for image or audio. This kind of problem is solved based on the following restricted rules; from input data, the neurons in first layer will extract informative and useful elements or features to help in the learning process which is called ‘feature extraction’; the group of features will be channeled to ‘feature mapping’ which carry the same synaptic weights

University of Malaya

and biases; and reducing the input data to be transmitted by performing 'subsampling', which is scaling down the dimensionality of previous feature maps (Haoxiang & Lin, 2015).

According to Kim, (2014), CNN perform optical character recognition (OCR) to digitize text where language processing became possible on hand-written documents and analog way too. It is used in sound applications when it is represented as an image like spectrogram. Recently, it has been applied to text analytics in addition to graph data as in graph convolutional networks.

CNNs works by passing many types of filters over an image where each filter will pick up a different signal of the image. At early layers, the filters are passing like horizontal, vertical, and diagonal lines to induce only a map of the image edges. CNNs take the slices of the image's feature from each filter to map them one by one that creates a map of each feature occurs that allows CNN to easily engineer the robust and scalable feature of image by learning different portions of image features. CNNs are easy to train and have fewer parameters than the fully-connected networks with the same number of hidden units (Haoxiang & Lin, 2015). Figure 2.13 shows a CNN complete training layers

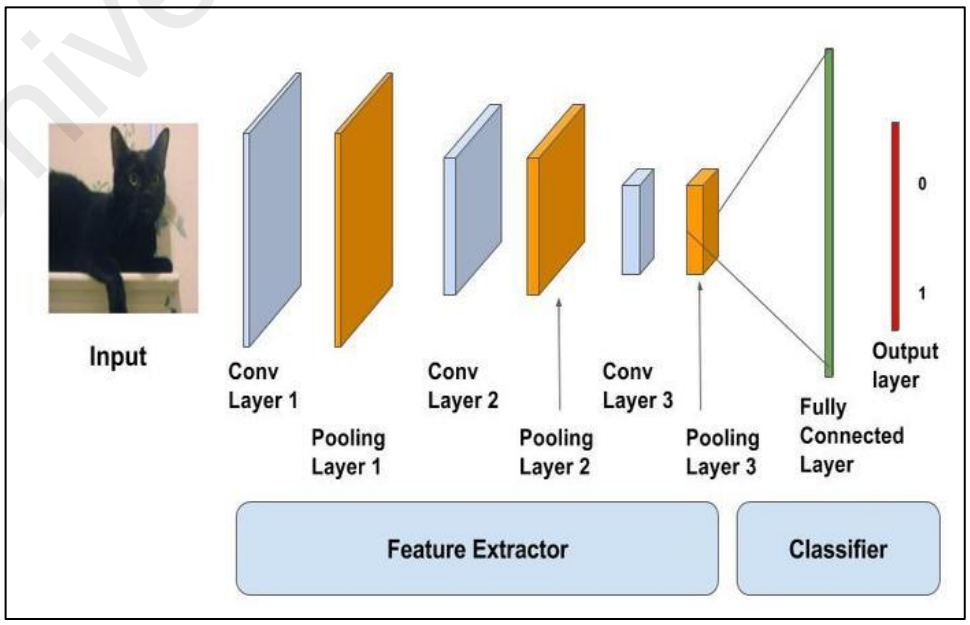

Figure 2.13: Convolutional Neural Network Architecture (Haoxiang & Lin, 2015)

University of Malaya

2.7 Learning Algorithm Types

Deep learning process is to learn a dataset of data input where it learns differently based on the signals used as input data. Learning algorithms are classified into four major types that are supervised, semi-supervised, unsupervised and reinforcement learning (Sharma & Kumar, 2017).

2.7.1 Supervised Learning

Supervised learning type is mainly used in two main areas which are classification and regression. Classification problems involve predicting discrete output. For examples given an image of tumor and to determine whether the tumor is benign or malignant. Regression problems involve predicting the future of continuous input or inferring a continuous function such as predicting the age of a person using an image of the person (Erik G., 2014).

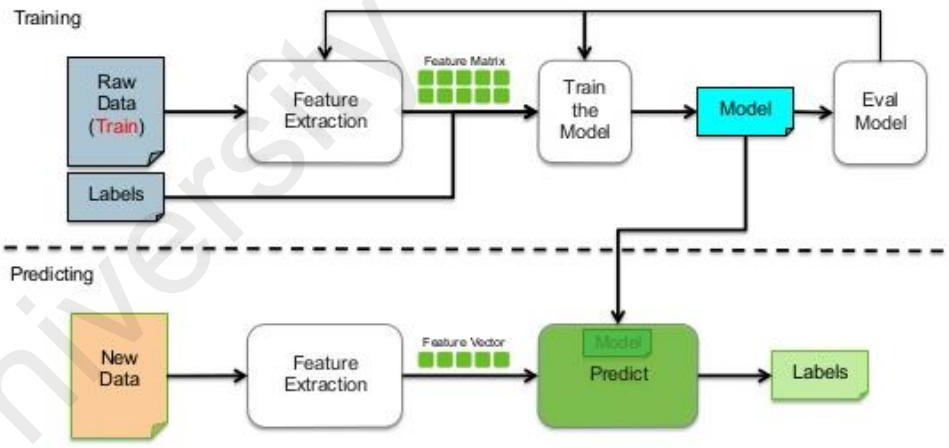

Figure 2.14: Supervised learning workflow (Sharma & Kumar, 2017)

Training dataset consists of n pairs of (x1,y1),…,(xn,yn) where x(i) is the input data and y(i) is the corresponding output or the label or class of the input data. X(i) contains features that leads it to a class in the vector y(i) for example if the algorithm learned that if the object is round and green then it belongs to the class “lime”. Test dataset on the other hand is the unlabeled input used to validate the inferences made from training

University of Malaya

dataset and obtain the learning accuracy (Dey & Learning, 2016; Sharma & Kumar, 2017).

Supervised learning type is mainly used in two main which are classification and regression. Classification problems involve predicting discreet output. For examples given an image of tumor and to determine whether the tumor is benign or malignant. Regression problems involve predicting the future of continuous input or inferring a continuous function such as predicting the age of a person using an image of the person (Erik G., 2014).

2.7.2 Unsupervised Learning

Unsupervised learning does not need labeled input data like supervised type. Unsupervised learning processes a cluster of input data and learns by itself different elements and features. The input data, x(i) is used to obtain the model of probability distribution of data p(x). Unsupervised learning performs task by identifying the features of raw input data and group them according to the similar traits that each input has. For example, a set of mixed types of fruits is used, then they are grouped to separate the different types of fruits (Karhunen, Raiko, & Cho, 2015).

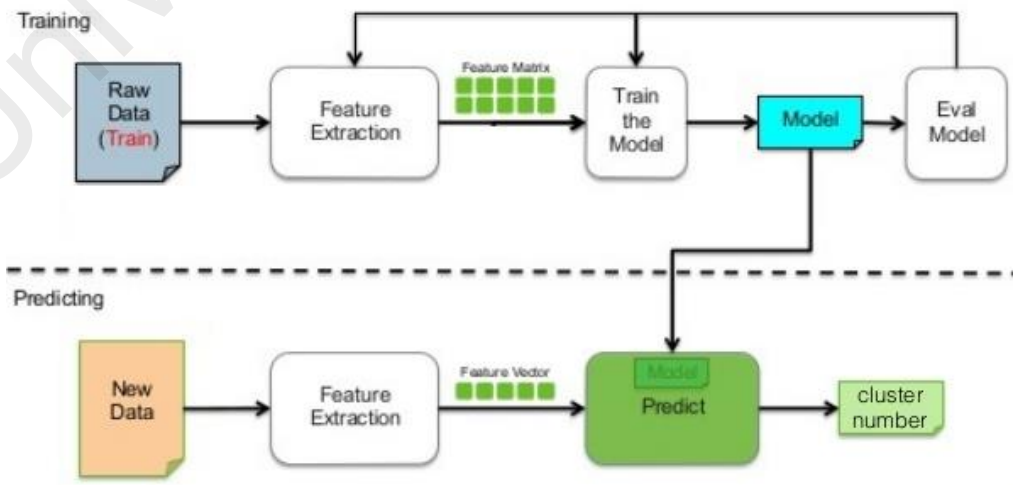

Figure 2.15: Supervised learning workflow (Ghahramani, 2003)

University of Malaya

Unsupervised learning has been used to perform the tasks of clustering as well as other tasks like compression tasks, generative modelling, and dimensionality reduction. This learning paradigm is popularly used to create self-organizing maps, k-means clustering, and nearest neighbor mapping. Machine learning experts anticipate for unsupervised learning to become more significantly important (Ghahramani, 2003; Weber, Welling, & Perona, 2000).

2.7.3 Semi-supervised Learning

Semi-supervised learning is a combination of supervised and unsupervised learning types to extract features and form function to predict an output where both unlabeled and labeled data types are used to train the system. It is useful learning algorithm especially when large amount of labeled data is needed as labeling large amount of data is a time consuming. This learning method uses a small amount of labeled data and learn from it and then uses the unlabeled input and extract similar features based on the labeled ones. Semi-supervised is widely used in face recognition, speech recognition and web-content classification (Hajighorbani, Mohammad, Hashemi, Broumandnia, & Faridpour, 2016).

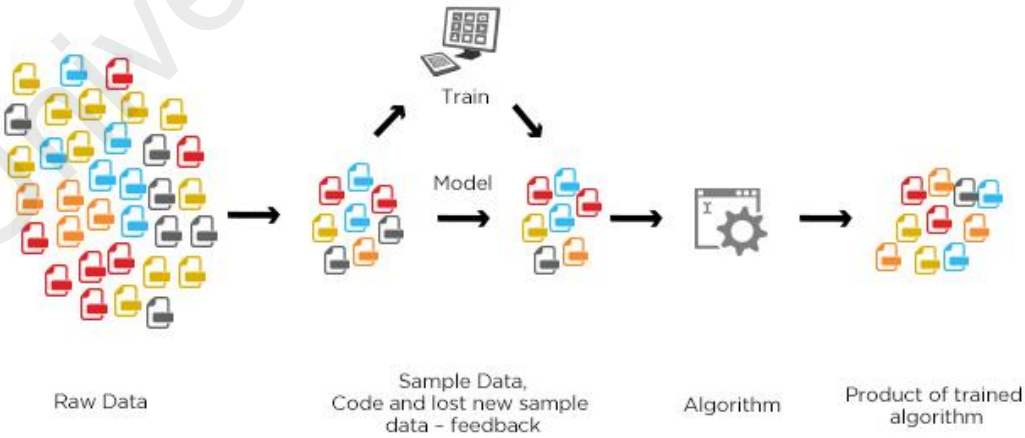

Figure 2.16: Semi-supervised learning process (Hajighorbani et al., 2016)

University of Malaya

2.7.4 Reinforcement Learning

According to Zhan, Ammar, & Taylor, (2016), reinforcement learning deals with a series of decision making which utilizes a weaker training set compared to supervised learning. The training set is its dynamic environment that interacts with the algorithm. This type of learning depends on the level of rewards to produce outcomes through trial and error. Three main components involve in this method of learning are the agent, which is the system or learner, the environment which is what the agent interacts with, and action which is the agent reaction as a response after interacting. The agent or the system learns through feedback given in terms of rewards and punishment. Reinforcement learning will take actions which gives the best rewards as it learns along a given time.

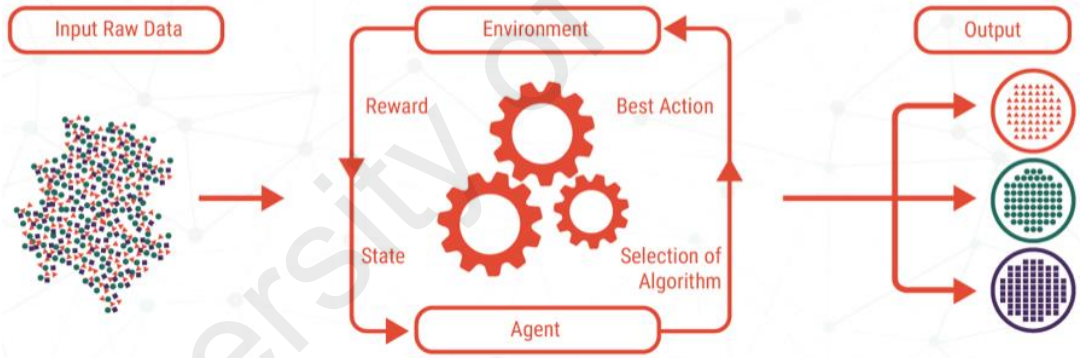

Figure 2. 17: Reinforcement learning process (Sharma & Kumar, 2017)

For example, images of different fruits are used as raw data, the agent at first will not know what the answer when given an input like orange. As the agent provides a certain answer, a feedback stating whether it is right or wrong will be given and the agent will learn from it. For the next time an orange is the input, the agent will already know that it is an orange from previous reward or punishment it receives. Reinforcement learning can be applied to control robotic arms optimally by figuring out the most efficient navigation as it learns from feedback when a collision occurs and how to avoid. It is also used in some games applications (Zhan et al., 2016).

University of Malaya

2.8 Training Parameters

The parameters that are used to tune the training in deep neural network training are known as training parameters that help in improving accuracy, as well as improving training time. The following are training parameters:

a) Batch Size

The batch size is the number of datasets given to a network per iteration, the number of batch sizes affect the model training time, if a complete dataset is passed through the network on each iteration, the training time maybe less. However the accuracy will be affected as the network will become more generalized to the dataset that we already have (Dey & Learning, 2016).

b) Epoch

Epoch is the number of times the dataset is being given completely to the network, for instance if the dataset size is 1100 and the batch size is 100 it would require 11 iterations to complete one epoch, epochs can help reuse the same datasets for training again.

c) Loss

Loss is the value that is calculated after each iteration to define the error. For example, if we have a dataset of number 0-9 and the given dataset for this training phase was the number 9, the output of the training will yield probabilities for each class of numbers. These probabilities are used in the loss function which differ from different models. Several loss functions are usually used in deep learning like Mean Square Error (MSE) for linear layers or cross-entropy for softmax layers such that the back-propagated error becomes the difference of the prediction and the target (Ankit et al., 2016).

University of Malaya

d) Optimizer

The optimizer helps reduce the output error of the loss function by changing the weights and bias values in the model. This differs based on one optimizer to another. Several optimizers used in deep learning like Gradient Decent, Stochastic, and Adam (Adaptive Moment estimation).

Gradient Decent is the most important technique and the foundation of how we train and optimize intelligent systems. Gradient Descent calculates the gradient of the whole dataset but will only perform one update at a time. Hence it is very slow that make it hard to control the large datasets and does not fit in the memory. Stochastic Gradient Descent (SGD), on the other hand, performs a parameter update for each training example that make it usually much faster technique. It performs one update at a time (Deng & Platt, 2014).

Adam stands for Adaptive Moment Estimation. Adam is another technique that computes adaptive learning rates for each parameter and stores the average exponential decay of the past squared gradients as it keeps an exponentially decaying average of past gradients. Adam is suitable for the need of fast convergence and highly complex neural network because it outperforms every other optimization algorithm (Zazo, Lozano-Diez, Gonzalez-Dominguez, Toledano, & Gonzalez-Rodriguez, 2016).

e) Activation Function

Activation function that are used in deep learning are like sigmoid, Softsign, Rectified Linear Unit (ReLU), Gaussian,and Softmax. Its role is to chooses whether the neuron should fire the data or not, this is done by obtaining the value received from the neuron and reevaluating it (Deng & Platt, 2014). For instance, in the ReLU function as shown in the Figure 2.18. The negative values are given as zero. Meaning that the output of the neuron will not be fired if the calculated data from the neuron is negative.

University of Malaya

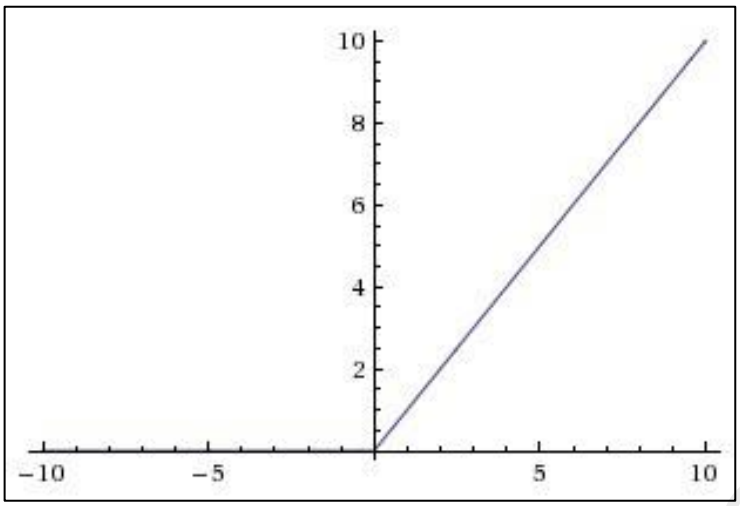

Figure 2.18: ReLu Activation Function

f) Learning Rate

Learning rate is a factor that is used along with the optimizer in changing the weights of the function, with a lower learning rate the model will take a longer time but end up more accurate. Higher learning rates often help in initially training the model. Once the model reaches to an acceptable accuracy, a lower learning rate will help us improve the accuracy better.

2.9 Spoken Arabic Digits Recognition

Arabic is a Semitic language and considered as one of the oldest languages in the world. It is the fifth widely used language of more than 200 million native speakers. Arabic is the first language in the 26 Arab countries. Standard Arabic has 34 phonemes, of which 28 are consonants and six are vowels. A phoneme is the smallest unit of sound that indicates a difference in the meaning of words (Alotaibi, 2005).

The Arabic digits zero to nine (*sĕfr, wa-hĕd, ʻaath-nāyn, tha-lăthah, ʻaar-ba-ʻaah, kham-sah, sĕt-tah, sub-ʻaah, tha-mă-nyĕh, and tĕsâh*) . All the Arabic digits polysyllabic words except zero that is a monosyllable. The only syllables in Arabic language are CVCC, CVC, CV, where C indicates a consonant, while V indicates a either a long or short vowel. Arabic utterances can only start with a consonant. Table 2.1 shows

University of Malaya

the ten Arabic digits, its pronunciation, number of syllables, and types of syllables in every spoken digit (Alotaibi, 2004).

Spoken digits recognition is one of the challenging tasks in the field of automatic speech recognition. Recognition of spoken digit is needed in many applications like the one that needs the spoken numbers as input, such as telephone dialing using speech, airline reservations, and automatic directories to send or retrieve information.

Table2.1: Arabic digits pronunciation and syllables (Touazi & Debyeche, 2017)

Digit	Arabic Writing	Pronunciation	Syllables	Syllables
0	صفر	Sĕfr	CVCC	1
1	واحد	wa-hĕd	CV-CVC	2
2	اثنين	aath-nāyn	CVC-CVCC	2
3	ثلاثه	tha-lăthah	CV-CV-CVC	3
4	اربعه	aar-ba-‘aah	CVC-CV-CVC	3
5	خمسه	kham-sah	CVC-CVC	2
6	سته	sĕt-tah	CVC-CVC	2
7	سبعه	sub-‘aah	CVC-CVC	2
8	ثمانيه	tha-mă-nyĕh	CV-CV-CVC	3
9	تسعه	Tĕsâh	CVC-CVC	2

A few different techniques have been used in the researches of voice recognition on Arabic language digits. Arabic digits were investigated by (Alotaibi, 2009) and developed a system that is an isolated word speech recognizer that was implemented as in multi-speaker mode. The used data samples were processed for noise removal from digitized speech using band-pass filters. The signals were pre-emphasized and windowed by Hamming window.

A time alignment algorithm specifically Dynamic Time Wrapping with Multilayer Perceptron (MLP) was used to compensate the differences in utterance lengths and misalignments between phonemes. MFCC has been used for features extraction for several of 17 individual male Arabic speakers that were asked to utter all digits 10 times

University of Malaya

where the same speakers are used in both training and testing phase, this system recognized a 99.48% of the spoken digits (Alotaibi, 2009).

(Djemili et al., 2004) have adapted the use of the MLP back-propagation training algorithm to train the model. Hidden Markov Model was implemented to extract temporal features for the speech signal of five elements to represent the states, in addition to twelve Mel-frequency Cepstral Coefficients as input to the network. Hence, a training set consisting of twenty occurrences of each digit by 20 speakers was used. Half the talkers were male, half female. The same 20 talkers as were used in the testing phase and, however 77.3% accuracy were achieved.

(Abraham, 2013) proposed a speech recognition model using neural network structure with long term memory (LTM) that is inspired by long term memory of human cortex. The feature extraction technique used is MFCCs to produce fine representation of speech signal. The extracted features are then fed into the neural network with LTM cells that can learn the sequences.

The dataset used by (Abraham, 2013) consists of 8800 samples of utterances collected from 88 speakers where each of the speaker has repeated each digit from 0 – 9 for 10 times. Half of the data was used as training dataset and the other half used in the testing phase. Each of the digit's features is extracted by 13 MFCCs. The results show that the developed LTM model with the finest tuned parameters is 99% accurate in spoken Arabic digits datasets recognition.

(Saeed & Nammous, 2005) have used a totally different technique where the samples of speech signals have been processed as an image using Power Spectrum Estimation. This technique helps to extract the speech features from spectral analysis. Experiments concluded the technique of power spectrum estimation using Burg's model is a great approach to smooth an irregular spectral shape resulting from applying the FFT.

University of Malaya

This technique is based on the Linear Predictive Coding approach. Radial Basis Functions neural networks have been used for this study with high accuracy rate of 98%.

Table2.2: Comparison of different spoken Arabic digits recognition studies

Reference	Feature extraction technique	Network methods	Recognition rate
(Saeed & Nammous, 2005)	Power spectrum estimation of Burg's model	RBF	98%
(Abraham, 2013)	MFCCs	LTM	99%
(Djemili et al., 2004)	MFCCs	MLP+HMM	77.3%
(Alotaibi, 2009)	MFCCs	DTW+MLP	99.48%

University of Malaya

CHAPTER 3: METHODOLOGY

3.1 Introduction

This chapter presents the research procedure, solution methods, development, and algorithm being proposed to solve the complex problem of speech recognition on Arabic language digits. The following flow chart represent the total research methodology steps to achieve this project and train the spoken Arabic digits recognition system.

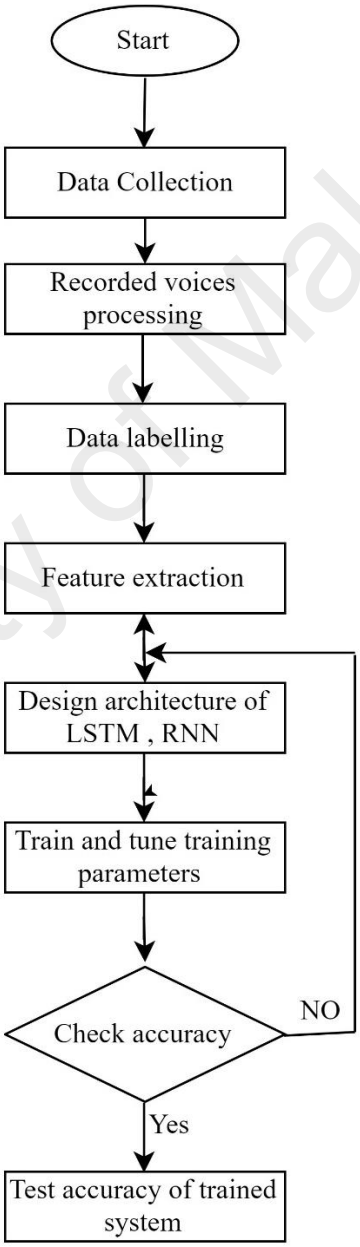

Figure 3.1: Research methodology flow chart

University of Malaya

3.2 Data Collection

The data is required to be collected from several Arabic speakers as the first task for this project. Speakers are required to record the Arabic digits 0 through 9 and pronounce each Arabic number separately, in a clear voice, with low noise in the background. Therefore, a data collection was required to create the speech recognition system, this data was collected throughout the world by different nationalities, having different dialects to pronouncing words because geographical locations affect a lot in how speech is pronounced but aren't necessarily the reason for the difference, as dialects, can be affected by family history.

The dataset was collected from 104 native Arabic speakers making total of 1040 data points as each speaker utters the 10 digits as batch in one record note with silence space in between. The data is divided into training dataset and testing dataset. The training data set is 840 data points (10 digits X 84 speakers) and the testing dataset is 200 data points (10 digits X 20 speaker).

The speakers are from different countries consisting male and female speakers with different dialects such as Yemen, Saudi Arabia, Iraq, Egypt, and Sudan. The dataset was collected through different social apps like WhatsApp, as the speakers had recorded voice samples, in opus format, Opus lossy audio coding format, which are lossy formats, are considered a better compression tool than the lossless. However, the quality of audio is affected when the audio is decompressed.

3.3 Signal Processing

As the data was gathered through different locations, each location consisted of a different noise. Some voice samples had minimal noise (in room recordings) where some were highly affected by noise (on road recordings). Therefore, noise removal was

University of Malaya

required to create an accurate system, as the noise varied for each voice sample, it was important to conduct signal processing on each signal separately.

The samples were given as a batch of one voice note by each speaker, Therefore, the samples were required to be out of noise as possible the divided into separate files where each file consist of one digit only prior to labelling. These processing methods can be done through MATLAB or Pyaudio. However, it requires a lot of effort. Therefore, a free open source software named Audacity was used. Audacity provide the function to modify the pitch effect. However to convert multiple data into a similar way, changing speed effect have been used.

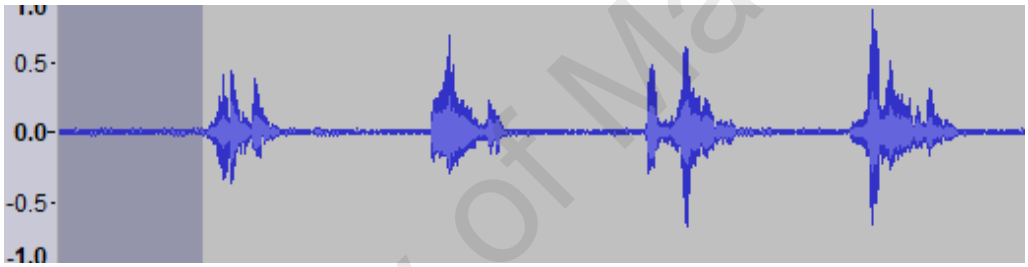

Figure 3.2: Selecting a Noise Profile in Audacity

Figure 3.2 shows a window of voice sample containing digits 0 to 4 with silence in between to select a noise profile in Audacity for noise reduction purpose. Audacity provides the user the ability to select a window of the voice sample to process each digit separately in one file. The noise reduction effect requires a noise profile which is generally a part of the voice sample that does not contain any useful information, or where the voice sample was supposed to be silent, once the data sample is selected as a noise profile, the entire track is selected, and the noise can be reduced as per preference.

The voice sample shown in Figure 3.2 are of a sample that had minimal noise in the background, although a noise reduction required to have a quality data to achieve this project objectives. Figure 3.3 and Figure 3.4 show the process of noise reduction and result of noise reduced signal.

University of Malaya

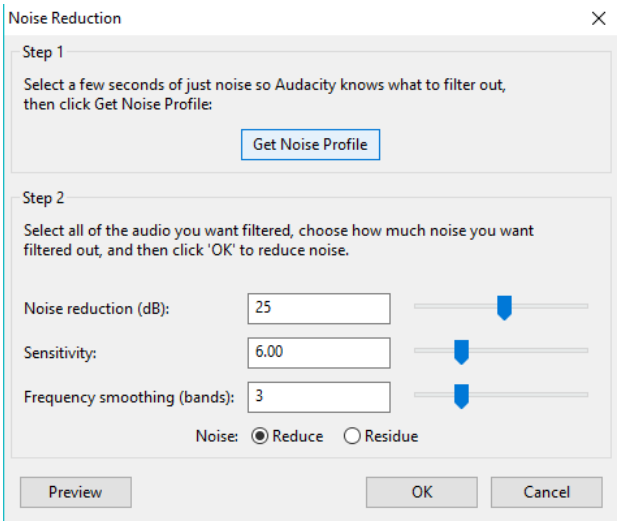

Figure 3.3: Adjusting Value of Noise Reduction

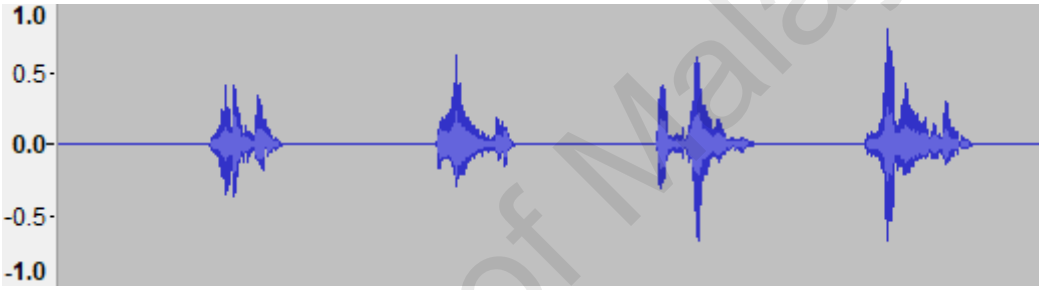

Figure 3.4: Waveform After Noise Reduction

Some other samples like in the case shown in Figure 3.5, the sample was very noisy and required a different method of processing, by first conducting the noise removal for the whole signal, then followed by listening to the exact points on where the speech is being told to silence all points where the noise was at. Figure 3.6 shows the same signal after noise reduction while Figure 3.7 shows the final result of signal after the process including silencing all the noise in between digits.

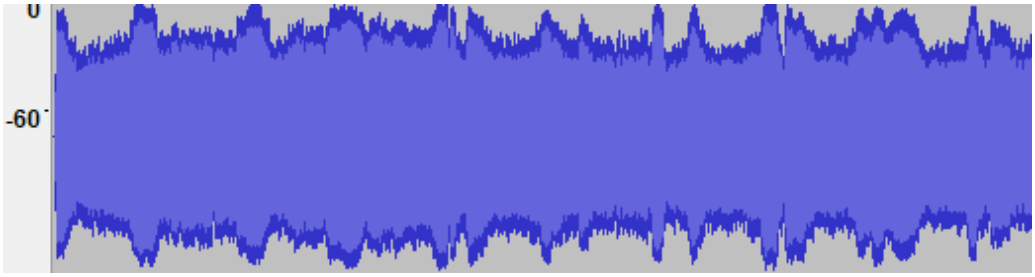

Figure 3.5: Logarithmic Waveform of Noisy Sample

University of Malaya

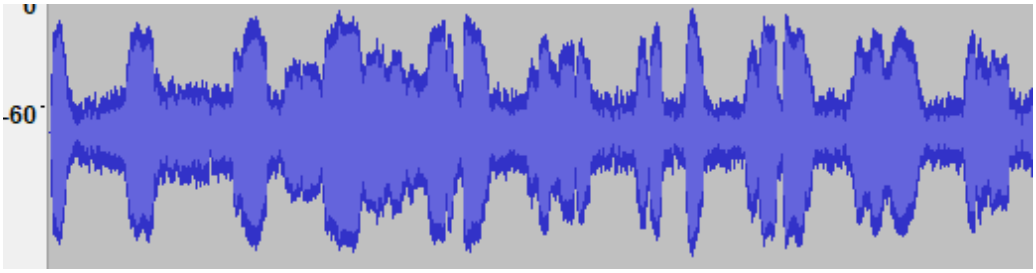

Figure 3.6: Noisy Sample After Noise reduction

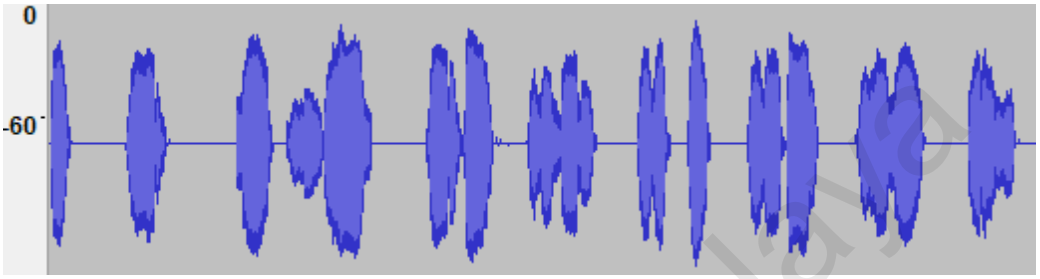

Figure 3.7: Noise Sample After Complete Processing

3.4 Data Labelling

It is essential that the data collected to be labelled, in a proper order. Once each sample of the 104 samples was processed, it was required to be divided and labeled, the Audacity software has a great tool for labelling and exporting multiple voice sample from one sample. Once the spoken number is labelled the speakers name is then added. Each voice sample is then divided, so that each digit is in one separate file only.

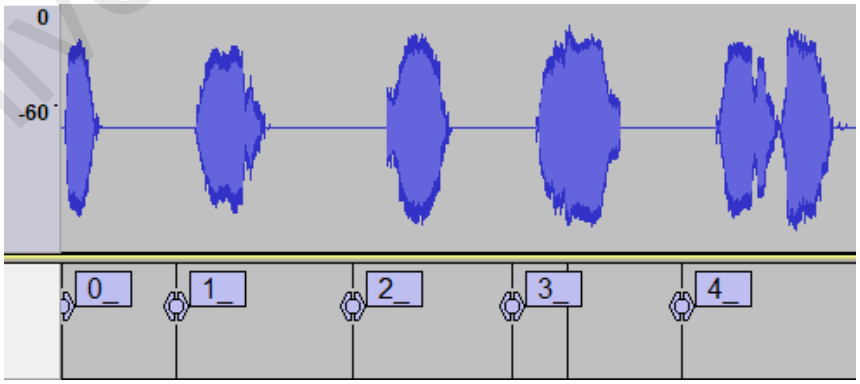

Figure 3.8: Labeling a Voice Sample

University of Malaya

3.5 Feature Extraction

Prior to model development and training, feature extraction of the audio files is required. Firstly, the files format of the all separated digits of 1040 audio file is converted into wav format to execute the feature extraction algorithm. This project used MFCCs for feature extraction purpose. Each of the spoken digit's features are extracted with 20 MFCCs. Each of the 20 MFCCs are computed based on the following conditions as shown in Table 3.1.

Table 3.1: MFCCs computation parameters

Sampling Rate	11025 Hz, 16 bits
Window applied	Hamming
alpha	0.79
Filter pre-emphasized H(z)	$1\text{-}0.95\ z^{-1}$

3.6 Training RNN Speech Recognizer

Recurrent Neural Network (RNN), Long Short-Term Memory (LSTM) speech recognizer for Arabic digits was developed using Python along with the use of Google's Tensorflow library for functions. These functions are primarily essential for the implementation. The algorithm is written in python script, as the Tensorflow library is written for Python and periodically updated.

The developed algorithm is then used to be trained for more than a trial to optimize the final model training parameters. The accuracy of the model and loss is checked through a function of TensorFlow named Tensorboard when tuning the parameters for optimization purpose. Different learning rate has been used in the model (0.0005, 0.0001, 0.002, 0.001, 0.01, and 0.1) to check which one is more efficient for the developed model. Different batch sizes have also been used (1040, 100, and 64).

University of Malaya

The developed model uses loss function to calculate the error and optimizer to optimize the result by changing the weights in the network. The optimized parameters are then used to train the final model of the recognizer. The training sessions was done under similar using laptop with GPU as it can affect computational time. The training is done of the data of 84 speakers of 10 digits each. Thus, training dataset used is 840 mixed male and female and from different dialect. The final model training progress is followed through Tensorboard.

3.7 Testing of RNN Speech Recognizer

Once the network is trained, it is required to analyze the output, this can be done by general verification of the output. The trained network is then applied to test digit speeches to estimate the accuracy of the system. Test data is 20 speakers with 10 digits for each (200 data points). These voices are unlabeled to calculate the accuracy of the developed Arabic spoken digits recognition model and produce the confusion matrix, then calculate the accuracy for each digit and the overall accuracy of the system, as well as understanding the structure of the neural network. This can be done using one of the functions for Tensorflow named as the Tensorboard. Having a good structure primarily affects the output. If the system is not organized properly the nodes may not be connected and therefore develop an inaccurate neural network.

3.8 Summary

Designing speech recognition system for Arabic digits involves phases from data collection for training and test tasks by shared records from different countries, process the voice signals for noise reduction also labeling the training voices and dividing them into separate files using Audacity software. Then Python codes with TensorFlow are used to carry out all the process of training and testing and producing results to be saved.

University of Malaya

CHAPTER 4: RESULTS AND DISCUSSION

4.1 Introduction

This chapter presents the results of this experimental study to develop a speech recognition for Arabic digits using LSTM Recurrent Neural Network. The results of parameters tuning are presented along with training results from Tensorboard, in addition to the result of model testing and its accuracy.

4.2 Environment

This research project was carried out using an Alienware laptop with AMD Radeon RX 570 GPU and CPU 2.9GHz Intel Core i7-, and memory of 16GB DDR4 SDRAM 2,400MHz and 8GB Nvidia GeForce GTX 1080. The operating system is Windows 10 and Linux. The software that has been used is Python 3.6. The total data used is 1040 data points.

4.3 Learning Rate Tuning of RNN Speech Model

The learning rate of the developed LSTM RNN has been tested for several values for final model parameters optimization purpose. The accuracy of each learning rate is presented in Figure 4.1. The graph shown is obtained from Tensorboard.

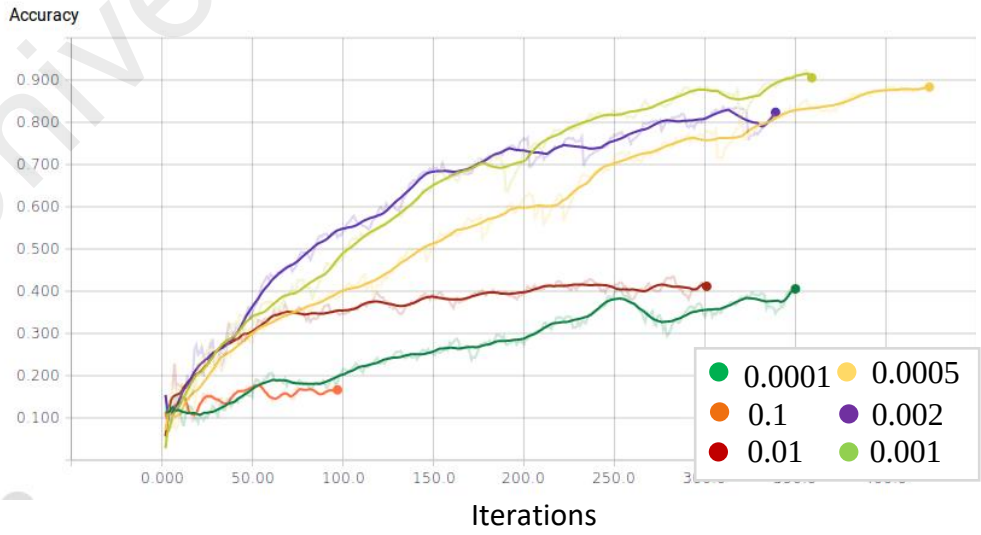

Figure 4.1: model accuracy for different learning rates

University of Malaya

The learning rates that is associated with time and accuracy convergence were tested for several rates. For 0.1 of learning rate, the accuracy of the network training algorithm per iterations is not varying and almost swinging around very low value of accuracy that is less than 20% because 0.1 learning rate is considered fast and does not allow training accuracy to increase to better value.

For 0.01 learning rate, the accuracy is changing a lot more. However, it reaches a constant value of accuracy around 40% after several iterations as 0.01 is relatively fast learning rate for speech recognition using LSTM. For 0.001 learning rate, the accuracy had risen much higher than all the previous rates as it reaches the 90% accuracy within 350 iterations which make this rate to be acceptable for training the model. A lower learning rate of 0.0001 the accuracy dropped around 50% difference than 0.001.

Therefore, 0.001 is a good learning rate for the developed network. Other rates within that range were tested to ensure that 0.001 was the correct value to be used. Two values were tested 0.002 and 0.0005. For 0.002, accuracy graph shows a similar change to the 0.001 learning rate model. However, it does not reach the same accuracy of 90% at the same iteration with 0.001 as it is dropped around 10%. For the 0.0005 learning rate, the training accuracy was good however it reaches 90% accuracy at 425 iterations. Thus, 0.001 learning rate is ensured to be the most suitable for model training.

Table 4.1: Summary of learning rate tuning and corresponding accuracy

Learning rate	Accuracy (%)
0.1	18
0.01	40
0.001	90
0.0001	40
0.002	80
0.0005	90

University of Malaya

4.4 Batch Size of RNN Speech Model

The batch size of the developed LSTM RNN has been tested for several values for final model parameters optimization purpose. The accuracy and loss of each batch size is presented in Figure 4.2 and Figure 4.3 Batch size influences the model accuracy. The graph shown is obtained from Tensorboard.

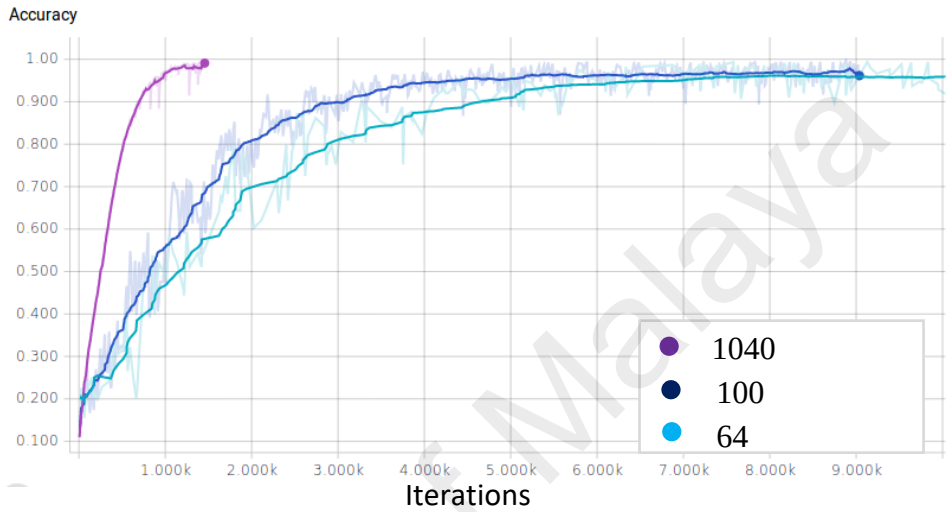

Figure 4.2: Accuracy of Different Batch sizes

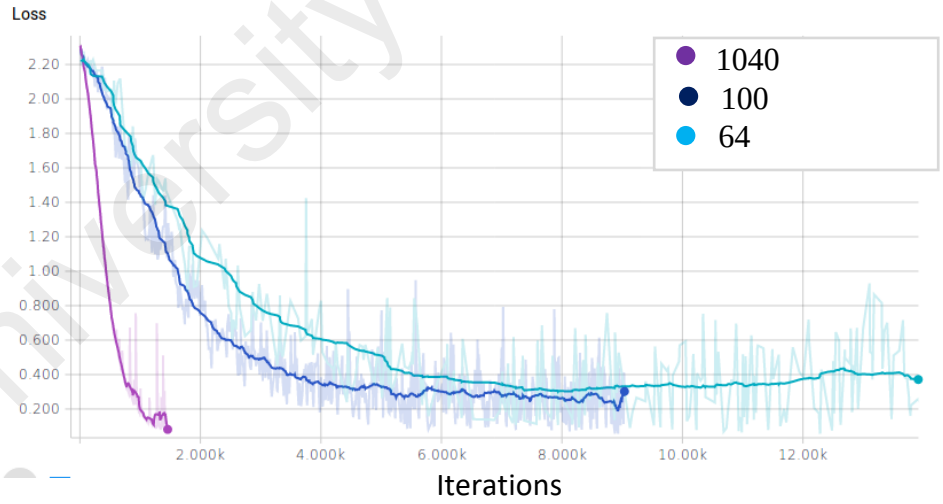

Figure 4. 3: Loss with Multiple Batch sizes

Batch sizes that were chosen are 1040 (Complete Dataset), 100 and 64 to define which batch size is better to be used for the developed network. The accuracy reaches higher and faster with 1040 batch size and the loss is minimal with around 1%. However, total dataset is not suitable to be used as training batch size due to the low number of

University of Malaya

validation data, the model will be less generalized and limited to that dataset, which means that the model will work accurately on the given data and it will not have that variation for other datasets.

For 100 and 64 batch sizes, the accuracy achieved is too high around 95% for both batch sizes and the only difference is the number of iterations taken to reach highest accuracy where 64 batch size is slower than 100 batch size. Loss is minimal for both sizes. Although the training time was affected, the 64-batch size was chosen as ideal for this model's training as it is smaller batch size that provides the network with enough time to be trained on and provides the network with more different sets of batch sizes that improves the training accuracy of the model.

4.5 Final Model Parameters and Training

Optimized learning parameters are obtained by the previous result trials. The most parameters found to be suitable is used to train the final model. The parameters used is follows

Table 4.2: Optimal training parameters

Learning rate	0.001
Training steps	3000
Activation function	Softmax
Optimizer	Adam
Loss function	Cross-entropy
Epoch	6200

The parameters in Table 4.2 used to train the network. The input to the network is converted from MFCCs. The Adam optimizer allow the network to go for higher accuracy by adaptive moment gradient change by changing the weights in the network to achieve better accuracy, while cross-entropy loss function calculate the error in

University of Malaya

exponential decay way. Learning rate used is 0.001. It is modified after 700 iterations to 0.0001 when accuracy starts to converge. The model achieved high accuracy of almost 94% as shown in Figure 4.4 and minimal losses in Figure 4.5.

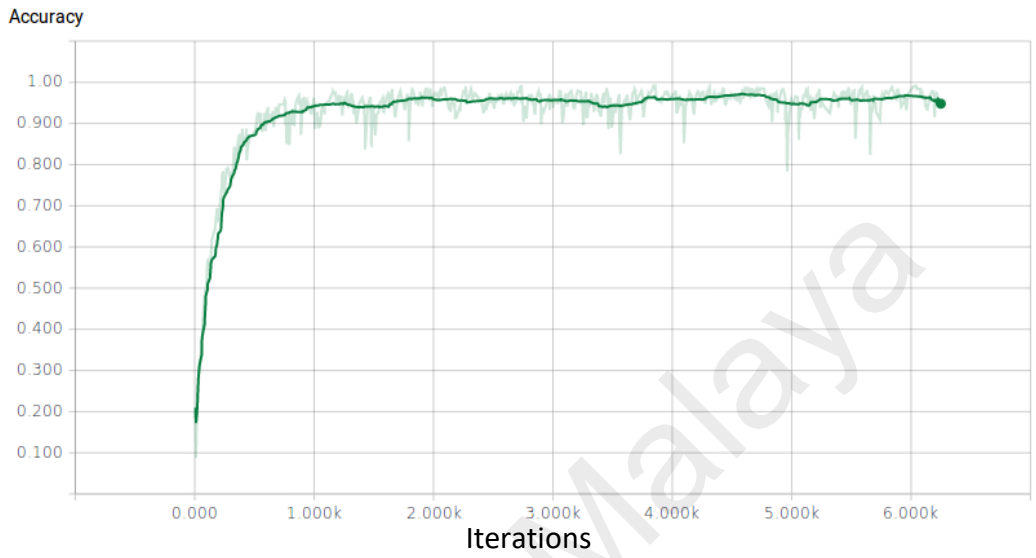

Figure 4.4: Accuracy of final model training

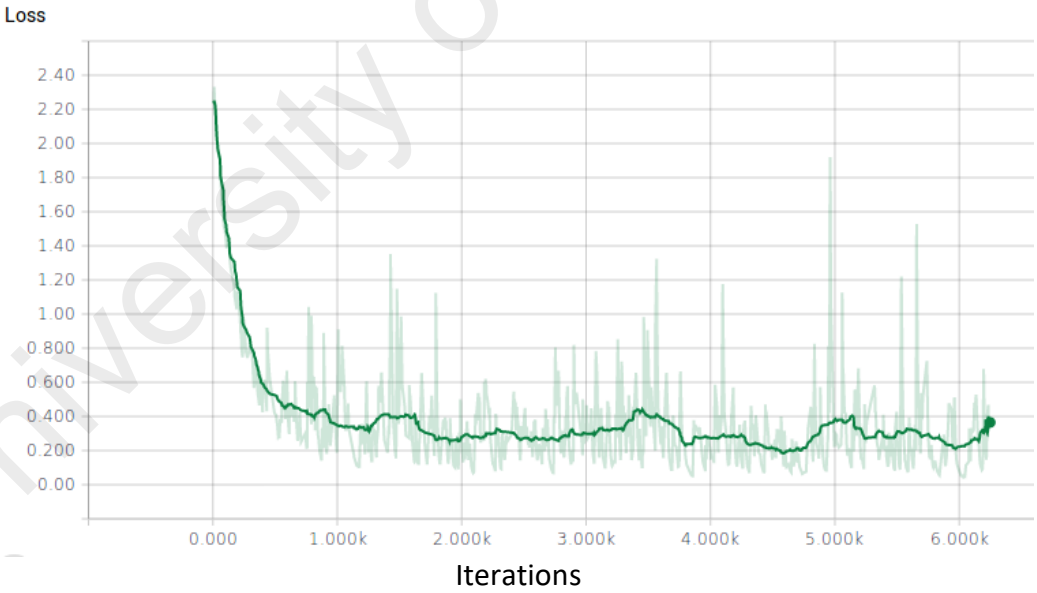

Figure 4.5: Loss Graph of final model training

4.6 Testing Result Analysis

The result for every voice sample tested varies depending on the quality of the digit voice files. Essentially, the recognizer sometimes could not recognize some of the sound features extracted if the signal has noise. The testing phase is done with 20

University of Malaya

speakers. Each speaker of 10 digits that are separated. Thus, a total of 200 data points is used for testing purpose where each digit is tested with 20 samples. Table 4.3 shows the confusion matrix of which digits recognized when specific digit is used. It also shows the recognizer accuracy for each digit and overall accuracy.

Table 4.3: Arabic digits confusion matrix and classification accuracy

	One	Two	Three	Four	Five	Six	Seven	Eight	Nine	Zero	Acc.(%)
One	15	0	1	1	0	0	1	0	1	1	75
Two	1	13	0	0	0	0	0	4	2	0	65
Three	4	0	13	3	0	0	0	0	0	0	65
Four	1	1	0	15	1	0	1	0	1	0	75
Five	0	0	1	2	13	0	0	0	3	1	65
Six	1	0	1	0	2	12	2	1	1	0	60
Seven	0	0	1	1	0	2	14	0	2	0	70
Eight	0	1	2	0	1	2	0	13	0	1	65
Nine	0	0	2	0	0	2	2	0	14	0	70
Zero	0	0	0	2	0	1	0	0	1	16	80
Total Acc.(%)											69

The best accuracy is obtained for number 0 where recognizer achieved an accuracy of 80% with 16 correct recognition and 4 mistaken digits and the least accuracy is number 6 with accuracy of 60%. Most digits are recognized with accuracy of 65%. The system achieved an average accuracy of 69% for 200 data points. This mediocre accuracy is achieved due to small of number of training dataset of 840 data points while deep learning usually requires a high number of data to train the network. Additionally, some similarity of the features for each number extracted by MFCCs that cause confusion when testing the network.

University of Malaya

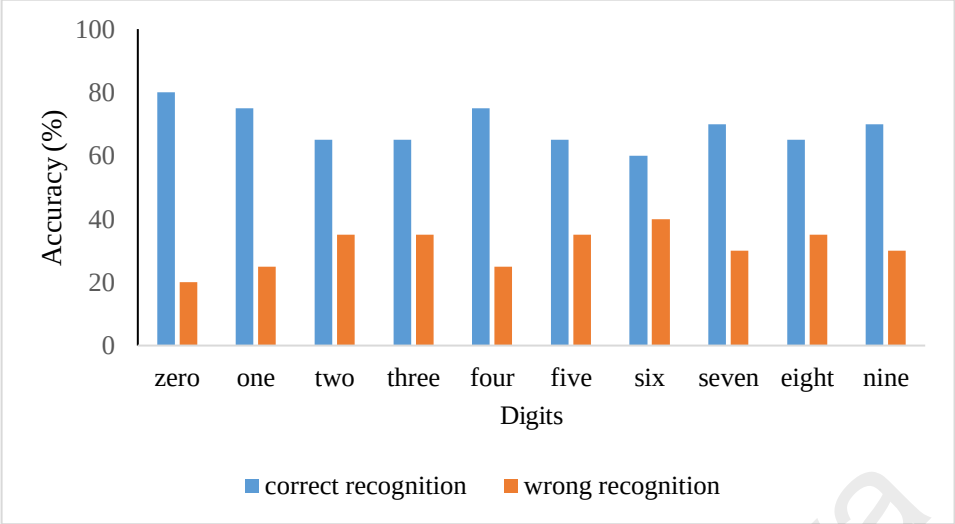

Figure 4.6: Column chart for digits recognition accuracy

Model testing accuracy can be calculated as the percentage ration between the total correctly recognized digits over the total testing dataset as follows

$$Accuracy(\%) = \frac{\text{total correct recognized digits}}{\text{total of testing dataset}} \times (100\%)$$

$$= \frac{15 + 13 + 13 + 15 + 13 + 12 + 14 + 13 + 14 + 16}{200} \times (100\%)$$

$$Accuracy(\%) = 69\%$$

Several digits have been misclassified as another digit. Digit '6' has the most misclassifications of testing result with total of 8 errors. It was mostly misclassified with digit '5' and digit '7' as shown in Table 4.3 Digit '0' has the least misclassifications of testing result with total of only 4 errors. Table 4.4 gives the list of misclassified digits.

Table 4.4: list of misclassified digits

No.	Confused with numbers	Number of errors
0	4,6,9	4
1	3,4,7,9,0	5
2	1,8,9	7
3	1,4	7
4	1,2,5,7,9	5
5	3,4,9,0	7
6	1,3,5,7,8,9	8
7	3,4,6,9	6
8	2,3,5,6,0	7
9	3,6,7	6

University of Malaya

The misclassifications occur to several digits due to their similarities to another digit's feature that can be analyzed by comparing the spectrogram of the digit or by comparing their MFCC features. Additionally, the network used only small number of dataset for training as deep learning require a large amount of data to train the network and achieve higher accuracy when tested.

However, this developed system is simple enough compared to other systems as described. The parameters that is used in this research is optimum to reach the high accuracy of the system. Moreover, the use of larger portion of training data or more number of MFCC coefficients may give a better result, but the efficiency of the system should be considered besides the effectiveness.

University of Malaya

CHAPTER 5: CONCLUSION AND RECOMMENDATIONS

5.1 Conclusion

This research has successfully developed a speech recognition solution for Arabic digits using Recurrent Neural Network. Long Sort-Term Memory (LSTM) has been chosen to carry out this experiment and utilize MFFCs to extract features of voice files. This model can recognize speeches of different noises from their various background. Although the time taken to train this detector is optimal, it takes quite a long period of time to train the network.

The model training is carried with the use of TensorFlow library utilized in Python script with the use of GPU. Model training was based on 840 data points out of total 1040 data points has been collected from different countries and dialects. The learning parameters has been optimized for training accuracy optimization purpose. The developed model has achieved an accuracy of 94% during training phase and minimal loss.

Final purpose of this research is to test accuracy of the model using a new test data set. 200 data points for all the digits are used for the testing phase where each digit is 20 sample. Despite a quite long processing time, the result seems promising with a 69% recognition accuracy where most of the digits can be recognized with 65% accuracy. The most successfully recognized digit is ‘0’ with 80% and least is ‘6’ with 60%. These numbers show there could be enhancement on the design of the RNN layers to further improve the accuracy.

5.2 Future Work

The accuracy of RNN could be greatly improved by using a large amount of training data as deep learning require large amount of data to achieve better accuracy.

University of Malaya

More variety of digit voice samples such as with noise could give us a more robust classifier. It is noteworthy that, more training data means the training time will also increase.

The accuracy of the model affected by its input. Since the input to the network is MFCCs feature extractor. The accuracy of RNN could be greatly improved by more number of MFCC coefficients may give better result, but the efficiency of the system should be considered besides the effectiveness. Additionally, the structure of the LSTM network affects the accuracy, therefore the increment of hidden layers may give a better accuracy result.

University of Malaya

REFERENCES

Abraham, A. (2013). Continous Speech Recognition Using Long Term Memory Cells, (December).

Alotaibi, Y. A. (2004). Spoken Arabic Digit Recognizer Using Recurrent Neural Network, 1–5.

Alotaibi, Y. A. (2005). Investigating spoken Arabic digits in speech recognition setting. *Information Sciences*, *173*(1–3), 115–139. https://doi.org/10.1016/j.ins.2004.07.008

Alotaibi, Y. A. (2009). A Simple Time Alignment Algorithm for Spoken Arabic Digit Recognition. *Jkau*, *20*(1), 29–43. https://doi.org/10.4197/Eng.20-1.2

Ankit, A., Mishra, S. K., Shaikh, R., Gupta, C. K., Mathur, P., & Pawar, S. (2016). A Survey Paper on Acoustic Speech Recognition Techniques, (4), 2347–2812.

Beaufays, F., Sak, H., & Senior, A. (2014). Long Short-Term Memory Recurrent Neural Network Architectures for Large Scale Acoustic Modeling Has. *Interspeech*, (September), 338–342. https://doi.org/arXiv:1402.1128

Chakraborty, C., & Talukdar, P. H. (2016). Issues and Limitations of HMM in Speech Processing: A Survey. *International Journal of Computer Applications*, *141*(7), 975–8887.

Chen, Z., Wang, J., He, H., & Huang, X. (2014). A Fast Deep Learning System Using GPU, (1), 1552–1555.

Deng, L., & Platt, J. C. J. (2014). Ensemble Deep Learning for Speech Recognition. *Research.Microsoft.Com*, (September), 1915–1919. Retrieved from http://research.microsoft.com/pubs/219987/EnsembleDL_submitted.pdf

Dey, A., & Learning, A. S. (2016). Machine Learning Algorithms : A Review, *7*(3), 1174–1179.

Djemili, R., Bedda, M., & Bourouba, H. (2004). Recognition of Spoken Arabic Digits Using Neural Predictive Hidden Markov Models, *1*(2), 226–233.

Erik G. (2014). Introduction to Supervised Learning, 1–5. Retrieved from https://people.cs.umass.edu/~elm/Teaching/Docs/supervised2014a.pdf%0Ahttp://people.cs.umass.edu/~elm/Teaching/Docs/supervised2014a.pdf

Fachrie, M., & Harjoko, A. (2015). using Elman Recurrent Neural Network Robust Indonesian Digit Speech Recognition using Elman Recurrent Neural Network. *Prosiding Konferensi Nasional Informatika (KNIF)*, (March), 49–54.

Ghahramani, Z. (2003). Unsupervised Learning. *Advanced Lectures on Machine Learning*, *3176*, 72–112. https://doi.org/10.1007/978-3-540-28650-9_5

University of Malaya

Graves, A., Mohamed, A., & Hinton, G. (2013). Speech Recognition With Deep Recurrent Neural Networks. *Icassp*, (3), 6645–6649. https://doi.org/10.1109/ICASSP.2013.6638947

Hajighorbani, M., Mohammad, S., Hashemi, R., Broumandnia, A., & Faridpour, M. (2016). A Review of Some Semi-Supervised Learning Methods. *Journal of Knowledge-Based Engineering and Innovation*, *2*(4), 250–259. Retrieved from http://aeuso.org/jkbei/wp-content/uploads/2016/06/27-A-Review-of-Some-Semi-Supervised-Learning-Methods.pdf

Halageri, A., Bidappa, A., Arjun, C., Sarathy, M. M., & Sultana, S. (2015). Speech Recognition using Deep Learning, *6*(3), 3206–3209.

Haoxiang, L., & Lin. (2015). A Convolutional Neural Network Approach for Face Identification. *IEEE Conference on Computer Vision and Pattern Recognition*, 5325–5334. https://doi.org/10.1109/CVPR.2015.7299170

Huang, X. D., Ariki, Y., & Jack, M. (1990). Hidden Markov Models for Speech Recognition.

Karhunen, J., Raiko, T., & Cho, K. (2015). Unsupervised Deep Learning: A Short Review. *Advances in Independent Component Analysis and Learning Machines*, 125–142. https://doi.org/10.1016/B978-0-12-802806-3.00007-5

Kim, Y. (2014). Convolutional Neural Networks for Sentence Classification. https://doi.org/10.3115/v1/D14-1181

Ko, T., Peddinti, V., Povey, D., & Khudanpur, S. (2015). Audio augmentation for speech recognition. *Proceedings of the Annual Conference of the International Speech Communication Association, INTERSPEECH*, *2015–Janua*, 3586–3589.

Lafta, H., & Yousif, A. Y. (2015). Mel frequency Cepstrum Coefficients and Enhanced LBG algorithm for Speaker Recognition, (November).

Levinson, S. E. (1986). Continuously variable duration hidden Markov models for automatic speech recognition. *Computer Speech & Language*, *1*(1), 29–45. https://doi.org/10.1016/S0885-2308(86)80009-2

Lipton, Z. C., Kale, D. C., Elkan, C., & Wetzel, R. (2015). Learning to Diagnose with LSTM Recurrent Neural Networks, 1–18. https://doi.org/10.14722/ndss.2015.23268

Mikolov, T., Karafiát, M., Burget, L., Cernocký, J., & Khudanpur, S. (2010). Recurrent neural network based language model. *Interspeech*, (September), 1045–1048.

Muda, L., Begam, M., & Elamvazuthi, I. (2010). Voice Recognition Algorithms using Mel Frequency Cepstral Coefficient (MFCC) and Dynamic Time Warping (DTW) Techniques, *2*(3), 138–143. https://doi.org/10.5815/ijigsp.2016.09.03

Rebai, I., Benayed, Y., Mahdi, W., & Lorré, J. P. (2017). Improving speech recognition using data augmentation and acoustic model fusion. *Procedia Computer Science*, *112*, 316–322. https://doi.org/10.1016/j.procs.2017.08.003

University of Malaya

Saeed, K., & Nammous, M. K. (2005). A new step in arabic speech identification: Spoken digit recognition. *Information Processing and Security Systems*, 55–66. https://doi.org/10.1007/0-387-26325-X_6

Sak, H., Senior, A., & Beaufays, F. (2014). Long Short-Term Memory Based Recurrent Neural Network Architectures for Large Vocabulary Speech Recognition, (Cd). https://doi.org/arXiv:1402.1128

Sharma, D., & Kumar, N. (2017). A Review on Machine Learning Algorithms , Tasks and Applications, *6*(10), 1548–1552.

Stenman, M. (2015). Automatic speech recognition An evaluation of Google Speech. Retrieved from http://www.diva-portal.org/smash/get/diva2:852746/FULLTEXT01.pdf

Thangavelautham, J. (2017). FPGA Architecture for Deep Learning and its application to Planetary Robotics Pranay Reddy Gankidi Space and Terrestrial Robotic Exploration (SpaceTREx) Lab, (March).

Touazi, A., & Debyeche, M. (2017). An experimental framework for Arabic digits speech recognition in noisy environments. *International Journal of Speech Technology*, *20*(2), 205–224. https://doi.org/10.1007/s10772-017-9400-x

Vallejo, M., Isaza, C. V, & Lopez, J. D. (2013). Artificial Neural Networks as an alternative to traditional fall detection methods. *Conference Proceedings : ... Annual International Conference of the IEEE Engineering in Medicine and Biology Society. IEEE Engineering in Medicine and Biology Society. Conference*, *2013*(1), 1648–1651. https://doi.org/10.1109/EMBC.2013.6609833

Vyas, M. (2013). A Gaussian Mixture Model Based Speech Recognition System Using MATLAB. *Signal & Image Processing : An International Journal (SIPIJ)*, *4*(4), 109–118. https://doi.org/10.5121/sipij.2013.4409

Weber, M., Welling, M., & Perona, P. (2000). Unsupervised learning of models for recognition. *Eccv*, 18–32. https://doi.org/10.1007/3-540-45054-8_2

Zazo, R., Lozano-Diez, A., Gonzalez-Dominguez, J., Toledano, D. T., & Gonzalez-Rodriguez, J. (2016). Language identification in short utterances using long short-term memory (LSTM) recurrent neural networks. *PLoS ONE*, *11*(1). https://doi.org/10.1371/journal.pone.0146917

Zeyer, A., Doetsch, P., Voigtlaender, P., Schlüter, R., & Ney, H. (2017). A Comprehensive Study of Deep Bidirectional LSTM RNNs for Acoustic Modeling in Speech Recognition. *Proc. ICASSP*, 2462–2466. https://doi.org/10.1109/ICASSP.2017.7952599

Zhan, Y., Ammar, H. B., & Taylor, M. E. (2016). Theoretically-grounded policy advice from multiple teachers in reinforcement learning settings with applications to negative transfer. *IJCAI International Joint Conference on Artificial Intelligence*, *2016–Janua*, 2315–2321. https://doi.org/10.1038/nature14236

University of Malaya