ÇUKUROVA UNIVERSITY
INSTITUTE OF NATURAL AND APPLIED SCIENCES

MSc THESIS

Serhat OK

A TURKISH BROADCAST NEWS SPEECH DATABASE FOR INVESTIGATION OF THE EFFECT OF DEEP NEURAL NETWORK AND LONG SHORT TERM MEMORY HYPERPARAMETERS ON SPEECH RECOGNITION BASED SYSTEMS

DEPARTMENT OF COMPUTER ENGINEERING

ADANA-2021

ABSTRACT

MSc THESIS

A TURKISH BROADCAST NEWS SPEECH DATABASE FOR INVESTIGATION OF THE EFFECT OF DEEP NEURAL NETWORK AND LONG SHORT TERM MEMORY HYPERPARAMETERS ON SPEECH RECOGNITION BASED SYSTEMS

Serhat OK

ÇUKUROVA UNIVERSITY
INSTITUTE OF NATURAL AND APPLIED SCIENCE
DEPARTMENT OF COMPUTER ENGINEERING

Supervisor : Assoc. Prof. Dr. Zekeriya TÜFEKÇİ
Year: 2021, Pages: 53
Jury : Assoc. Prof. Dr. Zekeriya TÜFEKÇİ
: Assoc. Prof. Dr. Umut ORHAN
: Asst. Prof. Dr. Gökay DİŞKEN

Speech recognition is the transformation of spoken words and sentences into text. It is used for various dictation operations as well as voice control applications. There have been many studies on speech recognition in many countries recently, the biggest reason being that they have large speech datasets in their own language and are accessible to them. However, studies on speech recognition applications in our country are very few, one of the reasons is the lack of voice dataset.

In this study, a Turkish speech database has been developed for Turkish speech recognition based systems. Sound recordings were obtained from news broadcasted by Turkish News Tv Channels at different times. The stages of database creation are examined step by step and the tools used are discussed. The created dataset was shared on the web in a way that everyone can access in order to set a precedent for other studies.

Additionally, we investigated and compared the effect of the number of layers and number of cells hyperparameters on Long Short Term Memory (LSTM) and Deep Neural Network (DNN) models on Turkish Broadcast News Speech Dataset that we created.

Key Words: Speech Recognition, Deep Neural Networks, Recurrent Neural Networks, Long Short Term Memory, Turkish Speech Database

ÖZ

YÜKSEK LİSANS TEZİ

A TURKISH BROADCAST NEWS SPEECH DATABASE FOR INVESTIGATION OF THE EFFECT OF DEEP NEURAL NETWORK AND LONG SHORT TERM MEMORY HYPERPARAMETERS ON SPEECH RECOGNITION BASED SYSTEMS

Serhat OK

ÇUKUROVA ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ
BİLGİSAYAR MÜHENDİSLİĞİ ANABİLİM DALI

Danışman : Assoc. Prof. Dr. Zekeriya TÜFEKÇİ
Yıl: 2021, Sayfa: 53
Jüri : Assoc. Prof. Dr. Zekeriya TÜFEKÇİ
: Assoc. Prof. Dr. Umut ORHAN
: Asst. Prof. Dr. Gökay DİŞKEN

Konuşma tanıma, söylenen kelime ve cümlelerin metne dönüştürülmesidir. Ses ile kontrol uygulamalarının yanı sıra çeşitli dikte işlemleri için de kullanılmaktadır. Son zamanlarda birçok ülkede konuşma tanıma ile ilgili bir çok çalışma yapılmıştır, bunun en büyük nedeni kendilerine ait bir dilde büyük ses veri setlerinin olması ve bunların erişime açık olmasıdır. Fakat ülkemizde konuşma tanıma uygulamaları ile ilgili yapılan çalışmalar çok azdır, bunun nedenlerinden biri ses veri seti eksikliğidir.

Bu çalışmada, Türkçe konuşma tanıma tabanlı sistemler için bir Türkçe konuşma veritabanı geliştirilmiştir. Ses kayıtları Türkçe Haber TV kanallarının farklı zamanlarda yayınladıkları haberlerden elde edilmiştir. Veri tabanı oluşturma aşamaları adım adım incelenmiş ve kullanılan araçlar ele alınmıştır. Oluşturulan veri seti diğer çalışmalara da emsal teşkil etmesi açısından herkesin erişebileceği şekilde web ortamında paylaşılmıştır.

Ek olarak, katman sayısı ve hücre sayısı hiper parametrelerinin Uzun Kısa Süreli Hafıza (LSTM) ve Derin Sinir Ağı (DNN) modelleri üzerindeki etkisini oluşturduğumuz Türkçe Yayın Haberleri Konuşma Veri seti üzerinde inceledik ve karşılaştırdık.

Anahtar Kelimeler: Konuşma Tanıma, Derin Sinir Ağları, Tekrarlayan Sinir Ağları, Uzun Kısa Süreli Bellek, Türkçe Konuşma Veritabanı

EXTENDED ABSTRACT

Speech is the fundamental communication tool among people. This communication flow has a complicated structure that is not just about the transmission of sound. There are many ways of communication exist in our lives to communicate with people, such as body language, sign language, textual language and speech.

Speech recognition is the process in which a speech signal is converted to a sequence group of words by implementing algorithms in a computer program (Santosh K. Gaikwad, 2010). Speech Recognition based systems and models are an effective area of research for seventy years and its popularity expand increasingly.

Speech recognition systems have three main steps (Yu and Deng, 2016). In first step, feature extraction process is performed on raw audio signals. In second step, acoustic model and language model are processed. Acoustic model takes extracted features as inputs and generates an acoustic model score for variable-length feature space. Language model estimates language model score for the words in training corpus. In the third step, the acoustic model score and the language model score are combined.

Nowadays, many studies have been done on speech recognition based applications and systems. When we investigate the distribution of various applications on speech recognition studies according to languages, there are extensive studies and datasets in other languages, especially in; English, Japanese, Chinese and Arabic languages. (http://kaldi asr.org/doc/examples.html). One of the most important reasons for these languages studies is that; they have large datasets, and most of them are improvable and accessible continuously.

The number of speech database created for Turkish is extremely little. Prepared speech databases are very important for researchers who are trying to develop a Turkish Speech Recognition Systems. Their number and diversity should increase. For this purpose, a Turkish speech database has been developed for

Turkish speech recognition based systems in this thesis. Sound recordings were obtained from news broadcasted by Turkish News Tv Channels at different times. This dataset contains approximately 2 hours of Turkish Broadcasts News Speech Dataset from a total of 1039 sound sentence files with their corresponding text transcripts were created. Also 8491 words (4101 different words), mono-phoneme and three-phoneme structures were created from all sentences.

The stages of database creation are examined step by step. The created dataset was shared on the web in a way that everyone can access in order to set a precedent for other studies.

Additionally, we investigated and compared the effect of the number of layers and number of cells on RNNs LSTMs and DNN hyperparameters on Turkish Broadcasting News Speech Dataset that we created. LSTM and DNN have several hyperparameters that affect the performance of deep learning systems regardless of the area of application. These hyperparameter values that change LSTMs and DNN structure, and can help achieve better performance with the same dataset. Number of layers were selected from 1 to 6 and number of units were selected as 50, 100, 150, 200, 250 and 300.

When number of layers were increased for LSTM, the performance increased until 3 and 4 layers, after 3 and 4 layers, the performance gets worse in our setup. When number of layers were increased for DNN, the performance increased until 4 and 5 layers, after 4 and 5 layers, the performance gets worse in our setup. In both models, the success rate decreases after a certain layer due to the situation of memorization.

When number of units were increased with fixed layer number for LSTM, the performance of the system gets better and later on performance rate were fixed. When number of units were increased with fixed layer number for DNN, the performance of the system gets better and later on performance rate were decreased.

LSTM produced more successful results than DNN. When previous studies on the effect of LSTM model on speech recognition systems were examined, it was expected that it gave better results and therefore LSTM would give more successful results in this study too.

Experimental results show that each parameter has its specific values for the selected number of training instances to provide lower error rates and better speech recognition performance. It is shown in this study that before selecting appropriate values for each LSTM and DNN parameters, there should be several experiments performed on the speech corpus to find the most eligible value for each parameter.

When the evaluation results of this study is investigated, all hyperparameters that we applied have effect on the performance of LSTMs and DNN for speech recognition systems.

For future work, larger dataset and better modelling strategies and hyperparameters can be used to obtain better performance. Currently, our models have less dataset for train and test than those of the studies that obtain speech recognition based models performances. Secondly the number of speech databases created for Turkish is very little. Prepared Speech databases are extremely important for researchers who are trying to develop a Turkish Speech Recognition System. Their number and diversity should increase.

As a result, this study set a precedent for creating a database for studies in Turkish speech recognition based areas.

The created dataset can be accessed from the following web extension; http://cusesveri.com/

GENİŞLETİLMİŞ ÖZET

Konuşma, insanlar arasındaki temel iletişim aracıdır. Bu iletişim akışı, sadece sesin aktarımı ile ilgili olmayan karmaşık bir yapıya sahiptir. Hayatımızda insanlarla iletişim kurmanın beden dili, işaret dili, metin dili ve konuşma gibi birçok yolu vardır.

Konuşma tanıma, bir konuşma sinyalinin bir bilgisayar programında algoritmalar uygulanarak bir dizi kelime grubuna dönüştürüldüğü süreçtir (Santosh K. Gaikwad, 2010). Konuşma Tanıma tabanlı sistemler ve modeller yetmiş yıldır etkili bir araştırma alanıdır ve popülaritesi giderek artmaktadır.

Konuşma tanıma sistemlerinin üç ana adımı vardır (Yu ve Deng, 2016). İlk aşamada, ham ses sinyalleri üzerinde özellik çıkarma işlemi gerçekleştirilir. İkinci adımda akustik model ve dil modeli işlenir. Akustik model, çıkarılan özellikleri girdi olarak alır ve değişken uzunluklu özellik alanı için bir akustik model puanı oluşturur. Dil modeli, eğitim külliyatındaki kelimeler için dil modeli puanını tahmin eder. Üçüncü adımda, akustik model puanı ve dil modeli puanı birleştirilir.

Günümüzde konuşma tanıma tabanlı uygulamalar ve sistemler üzerine birçok çalışma yapılmıştır. Konuşma tanıma çalışmalarına ilişkin çeşitli uygulamaların dillere göre dağılımını incelediğimizde, diğer dillerde özellikle; İngilizce, Japonca, Çince ve Arapça dilleri. (http://kaldi asr.org/doc/examples.html) birçok çalışma vardır. Bu dil çalışmalarının en önemli nedenlerinden biri; büyük veri kümelerine sahiptirler ve çoğu geliştirilebilir ve sürekli erişilebilir durumdadır.

Türkçe için oluşturulan konuşma veritabanı sayısı son derece azdır. Hazırlanmış konuşma veritabanları, Türkçe Konuşma Tanıma Sistemleri geliştirmeye çalışan araştırmacılar için çok önemlidir. Sayıları ve çeşitliliği artmalıdır. Bu amaçla, bu tezde Türkçe konuşma tanıma tabanlı sistemler için bir Türkçe konuşma veri tabanı geliştirilmiştir. Türk Haber Tv Kanallarının farklı zamanlarda yayınlanan haberlerden ses kayıtları alınmıştır. Bu veri seti, toplam 1039 ses cümle dosyasından yaklaşık 2 saatlik Türkçe Yayın Haber Konuşma Veri

Seti içermektedir ve bunlara karşılık gelen metin transkriptleri oluşturulmuştur. Ayrıca tüm cümlelerden 8491 kelime (4101 farklı kelime), tekli fonem ve üçlü fonem ses yapıları oluşturulmuştur. Veritabanı oluşturma aşamaları adım adım incelendi. Oluşturulan veri seti, diğer çalışmalara emsal teşkil etmesi açısından herkesin erişebileceği şekilde web üzerinde paylaşıldı.

Ek olarak, katman sayısının ve hücre sayısının RNNs LSTM'ler ve DNN hiperparametreleri üzerine oluşturduğumuz Turkish Broadcasting News Speech Dataset üzerindeki etkisini araştırdık ve karşılaştırdık. LSTM ve DNN, uygulama alanı ne olursa olsun derin öğrenme sistemlerinin performansını etkileyen birkaç hiperparametreye sahiptir. LSTM'leri ve DNN yapısını değiştiren bu hiperparametre değerleri ve aynı veri kümesiyle daha iyi performans elde edilmesine yardımcı olabilir. Katman sayısı 1'den 6'ya kadar seçildi ve birim sayısı 50, 100, 150, 200, 250 ve 300 olarak seçildi.

LSTM için katman sayısı artırıldığında performans 3 ve 4 katmana kadar arttı, 3 ve 4 katmandan sonra kurulumumuzda performans kötüleşti. DNN için katman sayısı artırıldığında performans 4 ve 5 katmana kadar arttı, 4 ve 5 katmandan sonra kurulumumuzda performans kötüleşti. Her iki modelde de ezberleme durumundan dolayı belli bir katmandan sonra başarı oranı düşmektedir.

LSTM için sabit katman sayısı ile birim sayısı artırıldığında, sistemin performansı daha iyi hale geldi ve daha sonra performans oranı sabitlendi. DNN için sabit katman sayısı ile birim sayısı artırıldığında, sistemin performansı daha iyi hale geldi ve daha sonra performans oranı düşmektedir.

LSTM, DNN'den daha başarılı sonuçlar üretti. LSTM modelinin konuşma tanıma sistemleri üzerindeki etkisine ilişkin önceki çalışmalar incelendiğinde, daha iyi sonuçlar vermesi ve dolayısıyla LSTM'nin bu çalışmada da daha başarılı sonuçlar vermesi beklenen bir durumdu.

Deneysel sonuçlar, daha düşük hata oranları ve daha iyi konuşma tanıma performansı sağlamak için her parametrenin seçilen eğitim örneği sayısı için kendi özel değerlerine sahip olduğunu göstermektedir. Bu çalışmada, her bir LSTM ve

DNN parametresi için uygun değerleri seçmeden önce, her bir parametre için en uygun değeri bulmak için konuşma veri seti üzerinde birkaç deney yapılması gerektiği gösterilmiştir.

Bu çalışmanın değerlendirme sonuçları incelendiğinde, uyguladığımız tüm hiperparametreler LSTM'lerin ve DNN'nin konuşma tanıma sistemleri için performansına etki etmektedir.

Gelecekteki çalışmalar için, daha iyi performans elde etmek için daha büyük veri kümesi ve daha iyi modelleme stratejileri ve hiperparametreler kullanılabilir. Şu anda, modellerimiz eğitim ve test için, konuşma tanıma tabanlı model performansları elde eden çalışmalara göre daha az veri kümesine sahiptir. İkinci olarak, Türkçe için oluşturulan konuşma veri tabanlarının sayısı çok azdır. Hazırlanmış Konuşma veritabanları, Türkçe Konuşma Tanıma Sistemi geliştirmeye çalışan araştırmacılar için son derece önemlidir. Sayıları ve çeşitliliği artmalıdır.

Sonuç olarak bu çalışma, Türkçe konuşma tanıma temelli alanlarda yapılan çalışmalar için bir veri tabanı oluşturmaya bir örnek oluşturmuştur.

Oluşturulan veri setine aşağıdaki web uzantısından erişilebilir; http://cusesveri.com/

ACKNOWLEDGEMENTS

I would like to thank my advisor Assoc. Prof. Dr. Zekeriya TÜFEKÇİ for his continuous and unconditional support, invaluable guidance and encouragement.

I would like to thank each and every member of the evaluation committee for their guidance.

I also want to thank my parents who never stopped believing in me. I appreciate all of their support and encouragements.

I would like to thank ZİRAAT TEKNOLOJİ A.Ş family not forgetting all of my beloved best friends, office friends, who supported me especially Bilal Burak

KOŞAR and Muhammed Emir ÖZÇEVİK for support during my MSc education.

I would like to especially thank İbrahim KARA and Şahin BEZENG because of the precious information and labor their gave me during the project.

CONTENTS **PAGE**

LIST OF TABLES **PAGES**

LIST OF FIGURES **PAGES**

ABBREVIATIONS

HMM	: Hidden Markov Model
RNN	: Recurrent Neural Network
ANN	: Artificial Neural Network
SRN	: Simple Recurrent Neural Network
LSTM	: Long-Short Term Memory Recurrent Neural Network
FFNN	: Feed Forward Neural Network
CTC	: Connectionist Temporal Classification
DL	: Deep Learning
DNN	: Deep Neural Network
LDC	: Linguistic Data Consortium
CAP	: Credit Assignment Path
VOA	: Voice Of America
TV	: Television
DARPA	: Defense Advanced Research Projects Agency
PCM	: Pulse Code Modulation
Wave –Wav	: Waveform Audio File Format
Nist-Sphere	: National Institute for Standards and Technology
LER	: Label Error Rate

1. INTRODUCTION

Speech is the fundamental communication tool among people. This communication flow has a complicated structure that is not just about the transmission of sound. There are many ways of communication exist in our lives to communicate with people, such as body language, sign language, textual language and speech. Gestures, the language, the subject and the capability of the listener contribute to this process. (Çömez, M.A, 2003). Moreover, speech is considered to be the richest and the most powerful communication due to its rich context and character. (Abdulkarim, M. D, 2015).

People are using many types of communication in daily life. The most common types are; written communications such as electronic mail and text messaging services, telephones and face-to-face speech communications. When these communication kinds are examined, the most important method in terms of providing the fastest, easiest and mutually effective communication is voice communication. This is because it interacts with the person by using the natural spoken language that both know.

Speech recognition is the process in which a speech signal is converted to a sequence group of words by implementing algorithms in a computer program (Santosh K. Gaikwad, 2010). Speech Recognition based systems and models are an effective area of research for seventy years and its popularity expand increasingly. Speech communication between human beings has played a significant part owing to its natural edge over other modes of human communication. While time passed, the requirement for better control of complicated machines appeared and speech response systems have started to play a big role in human-machine communication. (Patlar, F., 2009).

Speech recognition systems have three main steps (Yu and Deng, 2016). In first step, feature extraction process is performed on raw audio signals. Noise removal, signal conversion to feature domain, and feature extraction are performed

in feature extraction. In second step, acoustic model and language model are processed. Acoustic model takes extracted features as inputs and generates an acoustic model score for variable-length feature space. Language model estimates language model score for the words in training corpus. In third step, hypothesis search combines acoustic model score and language model score to generate final score and text transcription of the audio signal. The basic speech recognition steps are presented in Figure 1.1.

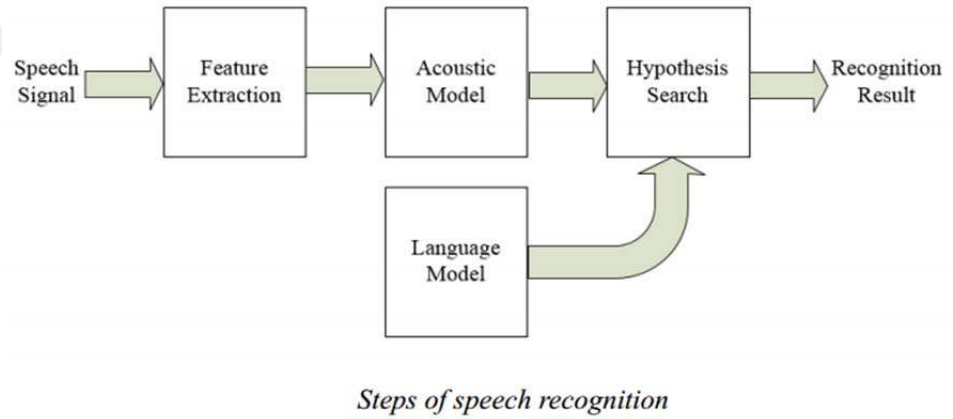


Figure 1.1. Basic Block Diagram of a Speech Recognition System.

Nowadays, many studies have been done on speech recognition based applications and systems. When we investigate the distribution of various applications on speech recognition studies according to languages, there are extensive studies and datasets in other languages, especially in; English, Japanese, Chinese and Arabic languages. (http://kaldi asr.org/doc/examples.html). One of the most important reasons for these languages studies is that; they have large datasets, and most of them are improvable and accessible continuously.

The number of speech database created for Turkish is extremely little. Prepared speech databases are very important for researchers who are trying to develop a Turkish Speech Recognition Systems. Their number and diversity should increase.

In the academic literature and other areas, many studies are presented for speech recognition based systems using deep learning architectures. The most used architectures are RNNs and its sub models Convolutional Neural Networks (CNNs), and Long-Short Term Memory (LSTM) but other architectures, such as Deep Neural Networks (DNNs), Deep Belief Networks (DBNs), and Deep Auto Encoder (DAE), are also used. (Tüfekci, Z., Dokuz, Y., 2020).

DNN is a Neural Network with multiple layers between the input and destination (output) layers. (Bengio, 2009). For a neural network system to be considered as a deep neural network, it is sufficient for the number of layers to be more than two. This value is the number of layers the model needs to complete the process. It can solve and analysis the complicated non-linear and uncomplicated linear relationships. The model moves through the between input and output layers calculating the probability of each output like a Feed Forward Neural Network (FFNN) structure.

RNN is one of the deep learning architectures which is efficient in processing sequential data inputs, like time series, or speech signals (Graves et al., 2013). RNNs process one input at a time and generate results at every time step. However, training RNNs is problematic because of the exploding or vanishing gradient problem at back-propagation process. To overcome this limitation, Long Short Term Memory (LSTMs) structure is proposed. (Hochreiter and Schmidhuber, 1997).

LSTMs is a special kind of RNN architecture which is capable of forgetting previous inputs that is not useful for current output (Graves et al., 2013b). For defining usefulness of previous inputs, LSTMs unit is composed of a cell, an input gate, an output gate and a forget gate. The cell remembers values over arbitrary time intervals and the three gates regulate the flow of information into and out of the cell. These gates carry several information between time steps and with the help of these gates exploding or vanishing gradient problem can be solved. LSTMs are also successfully applied in speech recognition based systems studies.

When the academic literature studies are analyzed, all of the deep learning systems have different architectures, frameworks, and configurations and optimization algorithms. Besides, all of these studies select hyperparameters for deep learning architecture based on their computation power. Therefore, real effect of deep learning architectures on speech recognition based systems performance is not easily observed from the literature.

Deep Learning Based Models have several hyperparameters that affect the performance of systems regardless of the area of application. These hyperparameters are values that change models' s structure, and can help achieve better performance with the same dataset. In particular, batch size, number of layers, number of units (cells), and number of epochs are evaluated on accuracy of speech recognition system. The number of layers and the number of units will apply as hyperparameters in this thesis study.

The aim of this thesis is primarily to create a speech database that can be used in Turkish speech recognition based systems and share this dataset in an electronic (web) environment. Additionally, we will investigate the effect of DNN and RNNs LSTMs hyperparameters on speech recognition performance.

We hope that this work will provide a further basis for further study in Turkish Speech Recognition and contribute to the accumulation of knowledge on this subject.

2. PREVIOUS WORK

2.1. Previous studies with Turkish Speech Database

In this section, information about the speech datasets created by using Turkish language will be given.

2.1.1. Turkish Broadcast News Speech and Transcripts

Turkish News Speech Dataset was developed by Bogazici University, Istanbul, Turkey and contains approximately 130 hours of Voice of America (VOA) Turkish radio news broadcasts by using a personal computer and Television/Radio card setup between 2006 and 2009.

This dataset is portion of a larger structure of Turkish broadcast news data gathered and voice-text transcribed with the purpose of facilitating research in Turkish speech recognition based systems and their relevant applications such as; speech signal, speech recognition and speech retrieval. The voice dataset was recorded at 32000 Hz (32 kHz) and resampled at 16000 Hz (16 kHz). (https://catalog.ldc.upenn.edu/LDC2012S06).

2.1.2. Middle East Technical University Turkish Microphone Speech Dataset

The METU dataset contains text, speech and alignment files. The Database's size about 600Mbytes. 120 speakers (60 male and 60 female) speak 40 sentences each (aproximately 300 words per speaker), that makes approximately 500 minutes of speech in total. The 40 utterances are selected randomly for each speaker from a treephone-balanced set of 2462 Turkish sentences. (https://catalog.ldc.upenn.edu/LDC2006S33)

2.1.3. Turkish Mobile Device Speech Dataset

This dataset contains approximately 6.1 hours of Turkish recorded speech, these speeches were recorded by using smartphones for Turkish speech

recognition. This speech database contains 549 speakers including males and females and totaly includes 5182 utterances.

2.2. Previous studies with Broadcast News Speech Database in Other Languages

In this section, information about speech datasets created using news broadcasts in other languages are reviewed. The information about the speech databases was taken from the LDC. (https://catalog.ldc.upenn.edu/byyear).

2.2.1. Czech Broadcast News Speech

Czech News Speech Dataset includes voice recording gathered from two Czech National TV channels and three Czech radio channels. It contains approximately 50 hours of broadcast news and 286 audio files. The voice files are mono-channel, 16 bit linear Waveform Audio File Format (Wave) and 22.05 kHz files. (https://catalog.ldc.upenn.edu/LDC2004S01).

TV Channels	Radio Stations
Ceska Televize	Cesky rozhlas 1 Radiozurnal
Prima TV	Cesky rozhlas 2 Praha
	Cesky rozhlas 3 Vltava

2.2.2. Spanish Broadcast News Speech (HUB4-NE)

Spanish News Speech Database contains approximately 30 hours of news voice recordings from the Tv sources.

The Speech recordings, gathered from a dedicated Tv satellite receiver are stored as; 16-kHz, mono-channel and 16-bit pulse code modulation (PCM) with NIST SPHERE format. The period of each speech recording is half hour (30 minutes) or 1 hour (60 minutes) . (https://catalog.ldc.upenn.edu/LDC98T29).

TV Channels

Televisa

Univision

VOA

2.2.3. Korean Broadcast News Speech

Korean News Speech Database contains 18 voice files by using VOA Radio News Channels. The Speech recordings, gathered from a dedicated radio and tv satellite receiver are stored as; 16-kHz, mono-channel and 16-bit pulse code modulation (PCM) with NIST SPHERE format. The period of each speech recording is half hour (30 minutes) or 1 hour (60 minutes). (https://catalog.ldc.upenn.edu/LDC2006S42)

2.2.4. English Broadcast News Speech (HUB4)

English HUB4 News Speech Dataset contains approximately 104 hours of news broadcast by using 3 native tv channels and two radio channels with corresponding voice-text transcripts. The speech files exist as a training database, development evaluation data. (www.catalog.ldc.upenn.edu/LDC97S44).

The following Tv and Radio channels were used in this corpus:

TV Channels

American Broadcasting Company Nightly News, World News Tonight, Nightline

Cable-Satellite Public Affairs Network (C-SPAN) - Washington Journal

CNN News Programs (Early Prime, Headline, Prime Time, The World Today)

Radio Channels

National Public Radio

Public Radio International

2.2.5. Mandarin Broadcast News Speech and Translations (HUB4-NE)

Mandarin News Speech Database contains approximately 30 hours of news voice recordings from the Tv and Radio stations sources. The voice database contains of 376000 words belong of English text and 517000 characters of belong to Mandarin text (https://catalog.ldc.upenn.edu/LDC2007T19).

TV Channels	Radio Stations
China Central TV	KAZN-AM
Voice Of America	

2.2.6. Arabic Broadcast News Speech

Arabic News Speech Dataset contains approximately 10 hours of voice recorded by using VOA radio station news in native Arabic language. The Speech recordings, gathered from a dedicated radio satellite receiver are stored as; 16-kHz, mono-channel and 16-bit pulse code modulation (PCM) with NIST SPHERE format. The period of each speech recording is half hour (30 minutes) or 1 hour (60 minutes). (https://catalog.ldc.upenn.edu/LDC2006S46).

2.2.7. GALE Phase 4 Chinese Broadcast News Speech

GALE4 Chinese News Speech Database contains approximately 134 hours of Mandarin Chinese news speech.

This dataset contains 256 voice files presented in waveform, 16kHz and mono-channel 16-bit PCM. (https://catalog.ldc.upenn.edu/LDC2017S25).

3. MATERIAL AND METHOD

3.1. Material

This section describes the dataset which we have created for Turkish Speech Recognition Based Systems and information about the programs used to create the database.

3.1.1. Programming Language and Framework

We used Python software programming language for development. It is an object oriented based programming. Nowadays, widely used in data science and academic areas because of interpretering and compiling upper level language programming. It is compatible with more deep learning frameworks than any other programming language. Also, we choosed Tensorflow as our framework because of its performance in speech recognition.

3.1.2. Libraries

3.1.2.1. NumPy

Numerical Python (NumPy) is a library for advanced mathematical operations that helps us work with multidimensional arrays and matrices belonging to the Python programming language. Nowadays, Numpy is a library that is frequently used by Python programmers, especially those working on data science.

3.1.3. Database

3.1.3.1. Turkish Broadcast News Speech Database

Çukurova University-Turkish Broadcast News Speech Database was created in Adana, Turkey by using the Turkish TV Broadcasts news channels published at different times. This dataset is portion of a structure of Turkish broadcast news dataset gathered and voice-text transcribed with the purpose of facilitating research in Turkish speech recognition based systems and their relevant

applications such as; speech signal, speech recognition and speech retrieval. The dataset contains approximately 2 hours of Turkish Tv news broadcasts; a total of 1039 sound sentence files and corresponding transcripts the text version of each sound file; a total of 1039 text sentence files, 8491 words (4101 different words), mono phoneme and three phoneme structures from all sentences.

Due to the lack of a Turkish voice data set, this data set was primarily created to be used in Turkish Speech Recognition Systems. Created dataset was shared on the web area to publicly accessible.

3.1.4. Process of Creating Database

3.1.4.1. Download of Audio Files

In the first phase of creating the dataset, different collected news records were downloaded in mp3 format with the Flvto Youtube Downloader program (https://www.flvto.biz/tr/youtube-downloaded).

3.1.4.2. Creation of Audio and Sentences Files

In speech recognition applications, sound recordings are processed in wav format, so downloaded audio files in mp3 format have been converted to wav format with the following line of code with the ubuntu (Linux) program.

- “ffmpeg -i Show_01_01_17.mp3 Show_01_01_17.wav ”

The same conversion from mp3. to wav. can also be done with the Audacity program (https://www.audacityteam.org/download/) on via of (File → Export →Export as Wav). Because of saving the wav files with small size, conversion were done on Ubuntu program.

Later on, audio files were imported into the Audacity program. The following figure (Figure 3.1) shows the sample audio file of a News Broadcast.

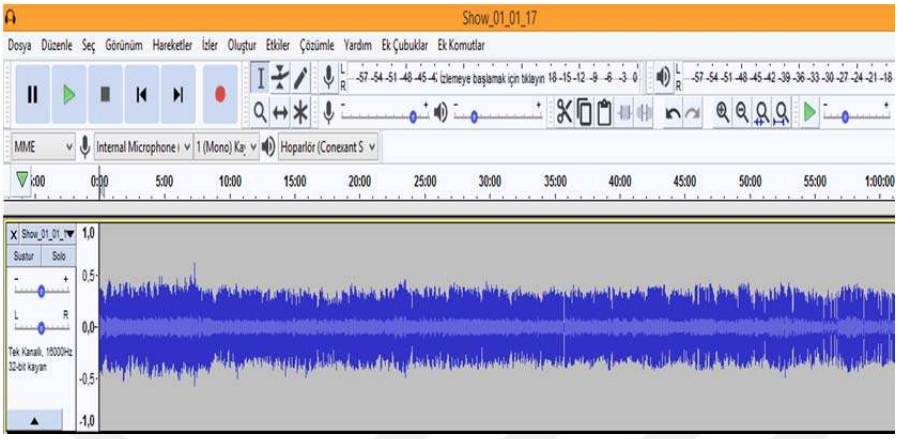


Figure 3.1. Importing Audio Files to Audacity

In order to define the expression in Speech Recognition Based Applications (SRBA), the datasets to be created should be selected in the form of a full sentence with Subject - Verb relation.

The following figure (Figure 3.2) shows the process of separating sentences from the audio file. System settings were set as below;

Default Sample Rate: 16000 Hz(16 kHz),

Default Sample Format: 16 bit,

Channels: 1 single channel (mono)

For the silence (sil) phoneme to be recognized by the system, the beginning and the end of the sentences were chosen together with a piece of silence.

We separated the sound datasets into 4 groups when extracting sentences from the audio file; noisy - clean (noiseless) - with music - mixed.

channel_date_type_number.wav format was adopted for naming the resulting dataset files as shown in the example below.

S_01_01_17_c_001 expression represent the;

S: News Channel Name

_01_01_17: Date of News

_c: Record type (clean)

_001: specifies the number of records received from the date of the relevant news

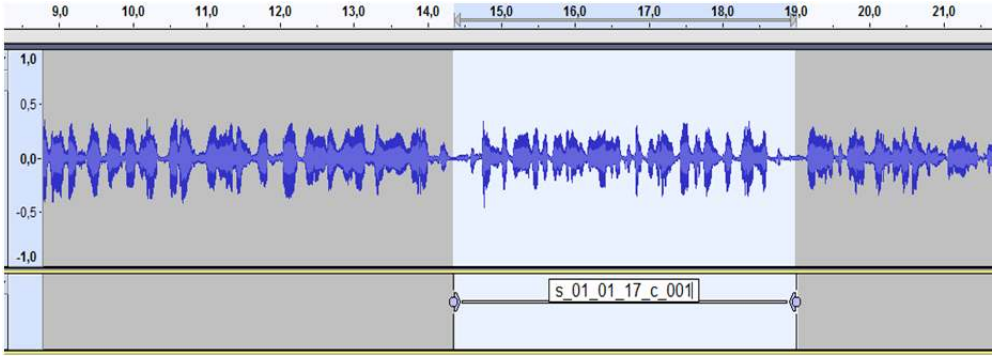


Figure 3.2. Separation of Sentences from the Audio File

Once the audio files were created, the labeling process was passed on the labeling process, the audio files are labeled as words. (Figure 3.3)

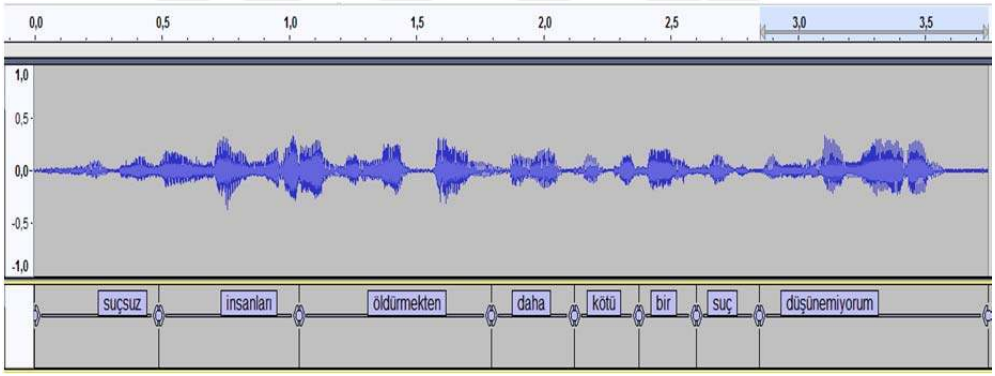
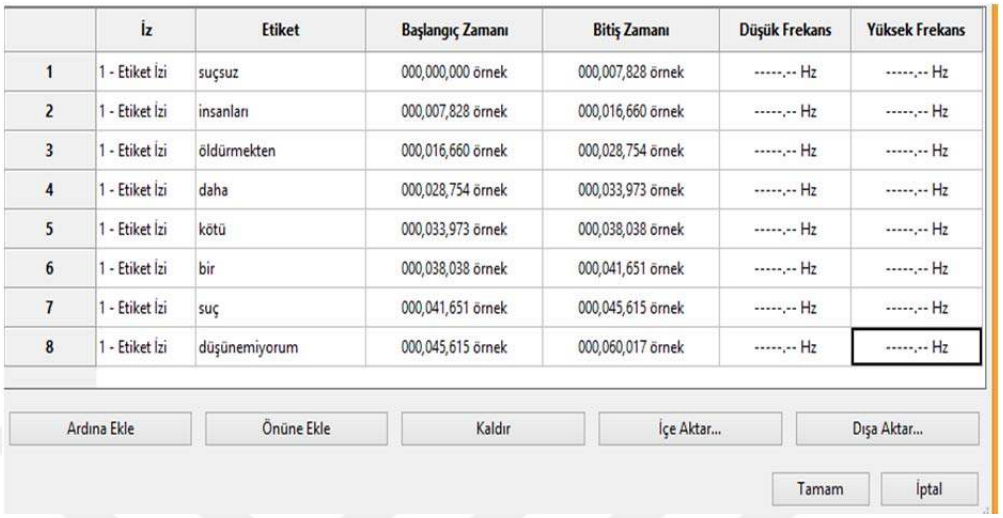


Figure 3.3. Creation of Word Labels

The transition between the labelled words should follow one another, the end time of a word must be synchronized and matched with initial time of the other word. In the following figures shows that (Figure 3.4 and Figure 3.5) a sample labeling model's output and Figure 3.6 shows the created sentences.



	İz	Etiket	Başlangıç Zamanı	Bitiş Zamanı	Düşük Frekans	Yüksek Frekans
1	1 - Etiket İzi	suçsuz	000,000,000 örnek	000,007,828 örnek	-----,-- Hz	-----,-- Hz
2	1 - Etiket İzi	insanları	000,007,828 örnek	000,016,660 örnek	-----,-- Hz	-----,-- Hz
3	1 - Etiket İzi	öldürmekten	000,016,660 örnek	000,028,754 örnek	-----,-- Hz	-----,-- Hz
4	1 - Etiket İzi	daha	000,028,754 örnek	000,033,973 örnek	-----,-- Hz	-----,-- Hz
5	1 - Etiket İzi	kötü	000,033,973 örnek	000,038,038 örnek	-----,-- Hz	-----,-- Hz
6	1 - Etiket İzi	bir	000,038,038 örnek	000,041,651 örnek	-----,-- Hz	-----,-- Hz
7	1 - Etiket İzi	suç	000,041,651 örnek	000,045,615 örnek	-----,-- Hz	-----,-- Hz
8	1 - Etiket İzi	düşünemiyorum	000,045,615 örnek	000,060,017 örnek	-----,-- Hz	-----,-- Hz

Ardına Ekle | Önüne Ekle | Kaldır | İçe Aktar... | Dışa Aktar...

Tamam | İptal

Figure 3.4. Export of Word Labels with Sample Unit

s_01_01_17_c_001.txt

```
1   0.000000    0.484563    suçsuz
2   0.484563    1.044375    insanları
3   1.044375    1.837875    öldürmekten
4   1.837875    2.073125    daha
5   2.073125    2.383625    kötü
6   2.383625    2.568687    bir
7   2.568687    2.857188    suç
8   2.857188    3.751063    düşünemiyorum
9
10  suçsuz insanları öldürmekten daha kötü bir suç düşünemiyorum
```

Figure 3.5. Export of Word Labels with Second Unit (s)

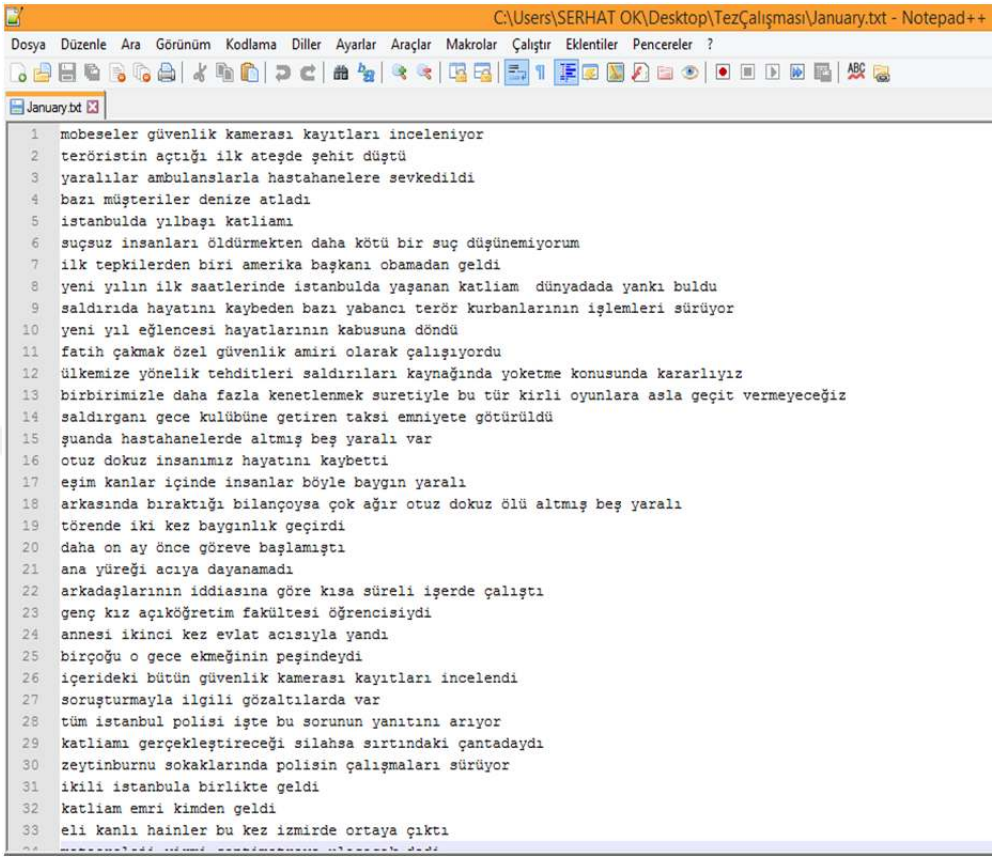


Figure 3.6. Created Sentence Examples

3.1.4.3. Creation of Phoneme Structure

In Speech Recognition Application, each phoneme is affected by the phoneme before it and affects the next phoneme.

In order to carry out statistical studies on phoneme basis for speech database; All special characters, except "dots(.) and commas(,)" must be cleared in the text.

Numbers must also be converted to letters. Because, in the framework of the speech signals, their pronunciations are meaningful rather than the numerical expressions. Numerical expressions can be read differently; some history, some

money, some number e.t.c. The some conversion examples are the dataset is as follows:

- polisler **63T5** sokaktaki sevim apartmanın **2.** katına geldiler
  polisler **altmış üç taksim beş** sokaktaki sevim apartmanın **ikinci** katına geldiler
- yaklaşık **5 6 cm** kar kalınlığı var
  yaklaşık **beş altı santim** kar kalınlığı var

Nowadays most programs used for speech recognition such as HTK do not recognize some characters in Turkish. In the American Standard Code for Information Interchange (ASCII) (Figure 3.7) 6 of these letters which belong to Turkish alphabet ç, ı, ı, ş, ö, ü not found in ASCII. Therefore, since most programs used for speech recognition cannot recognize these characters, these non-ASCII characters (ç, ğ, ı, ş, ö, ü) were defined by us in the phoneme list as follows.
Defining new values: **the closest letter + 1 (one = digit)** definition

- ç → c1
- ğ → g1
- ı → i1
- ş → s1
- ö → o1
- ü → u1

ASCII Code Chart

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	NUL	SOH	STX	ETX	EOT	ENQ	ACK	BEL	BS	HT	LF	VT	FF	CR	SO	SI
1	DLE	DC1	DC2	DC3	DC4	NAK	SYN	ETB	CAN	EM	SUB	ESC	FS	GS	RS	US
2		!	"	#	$	%	&	'	(	)	*	+	,	-	.	/
3	0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?
4	@	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
5	P	Q	R	S	T	U	V	W	X	Y	Z	[	\	]	^	_
6	`	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
7	p	q	r	s	t	u	v	w	x	y	z	{	|	}	~	DEL

Figure 3.7. ASCII Code Chart

There are 29 alphabets in Turkish (Figure 3.8), 6 of these letters ç, ı, ı, ş, ö, ü are not found in ASCII. For these non-ASCII letters (ç, ğ, ı, ş, ö, ü) new values have been defined in the phoneme list (c1, g1, i2, s1, o1, u1). Apart from these phonemes, the "sil" phoneme was defined to represent the silence parts, and "sp" phoneme was defined to represent the silence parts between words and for punctuation marks. As a result 31-character phoneme list was created. 29 letters + SIL + SP = 31 phonemes (Figure 3.8.).

TURKISH ALPHABET	ASCII CODE TURKISH PHONEME ALPHABET
a	a
b	b
c	c
ç	c1
d	d
e	e
f	f
g	g
ğ	g1
h	h
ı	i1
i	i
j	j
k	k
l	l
m	m
n	n
o	o
ö	o1
p	p
r	r
s	s
ş	s1
t	t
u	u
ü	u1
v	v
y	y
z	z
silence	sil
space	sp

Figure 3.8. Turkish Alphabet and Phonemes List

Monophoneme and threephoneme structures are used to increase the recognition of an expression in speech recognition. The order of the characters in an expression is called **monophoneme,** the combining of each character in an expression with the previous and next character called **threephoneme**. The monophonem and threephonem expansion of the **'üç zanlı gözaltına alındı.'** is as follows.

Sentence : üç zanlı gözaltına alındı.

MonoPhoneme : **sil** ü ç **sp** z a n l ı **sp** g ö z a l t ı n a **sp** a l ı n d ı **sil**

ThreePhoneme : **sil** u1+c1 u1-c1 **sp** z+a z-a+n a-n+l n-l+i1 l-i1 **sp** g+o1 g-o1+z o1-z+a z-a+l a-l+t l-t+i1 t-i1+n i1-n+a n-a **sp** a+l a-l+i1 l-i1+n i1-n+d n-d+i1 d-i1 **sp sil**

Mono-Phoneme	**Three-Phoneme**
sil	**sil**
u1	u1+ c1
c1	u1 - c1
sp	**sp**
z	z + a
a	z – a + n
n	a – n + l
l	n – l + i1
i1	l - i1
sp	**sp**
g	g + o1
o1	g - o1 + z
z	o1 – z + a
a	z – a + l
l	a – l + t

t	l – t + i1
i1	t - i1 + n
n	i1 – n + a
a	n - a
sp	**sp**
a	a + l
l	a – l + i1
i1	l - i1 + n
n	i1 – n + d
d	n – d + i1
i1	d - i1
sp	**sp**
sil	**sil** .

3.1.4.4. Creation of Dictionary and Word Files

A file containing the word list was created (Figure 3.9.), all words in the sentences are in this file, there are 4101 different words in total. This file has a list of all the words in the sentences.txt file, eliminated repetitions words, and alphabetical order of the words (alphabetical order must be done according to the ASCII set).

Obtained in the htk program as in the script below to create the wordlist. (http://www.voxforge.org/home/dev/acousticmodels/linux/create/htkjulius/tutorial/data-prep/step-2)

- julia ../bin/prompts2wlist.jl sentence.txt wordlist

sentence.txt: sentence file in the database

word.list: word file to be generated

prompts2wlist.jl: This script can create the word list file by using sentence.txt that we created.

word.list

```
4078  zarar
4079  zaten
4080  zavallisin
4081  zehir
4082  zehirlenecekti
4083  zekeriya
4084  zengin
4085  zeytinburnu
4086  zeytinburnunda
4087  zeytinburnundaki
4088  zilleri
4089  zincirleme
4090  zirhli
4091  zirvesinin
4092  ziyai
4093  ziyaret
4094  zor
4095  zorbasi
4096  zorbasinin
4097  zorla
4098  zorlanamayacak
4099  zorlastiriyor
4100  zorunda
4101  zorundaydi
4102
```

Figure 3.9. Word List

A dictionary file connects all value defined in the grammar to its suitable language model. A dictionary structure can be built using dataset for words pronunciations, text lists, and vocabulary. Such files are required to recognize larger data.

Pronunciation Dictionary file was created in (Figure 3.10.). This file contains all the words and their pronunciations in the sentences. The column on the left contains the words. The column on the right is the phonetic expansions of these words and the symbols they produce are in the middle column. (Gürler, 2014).

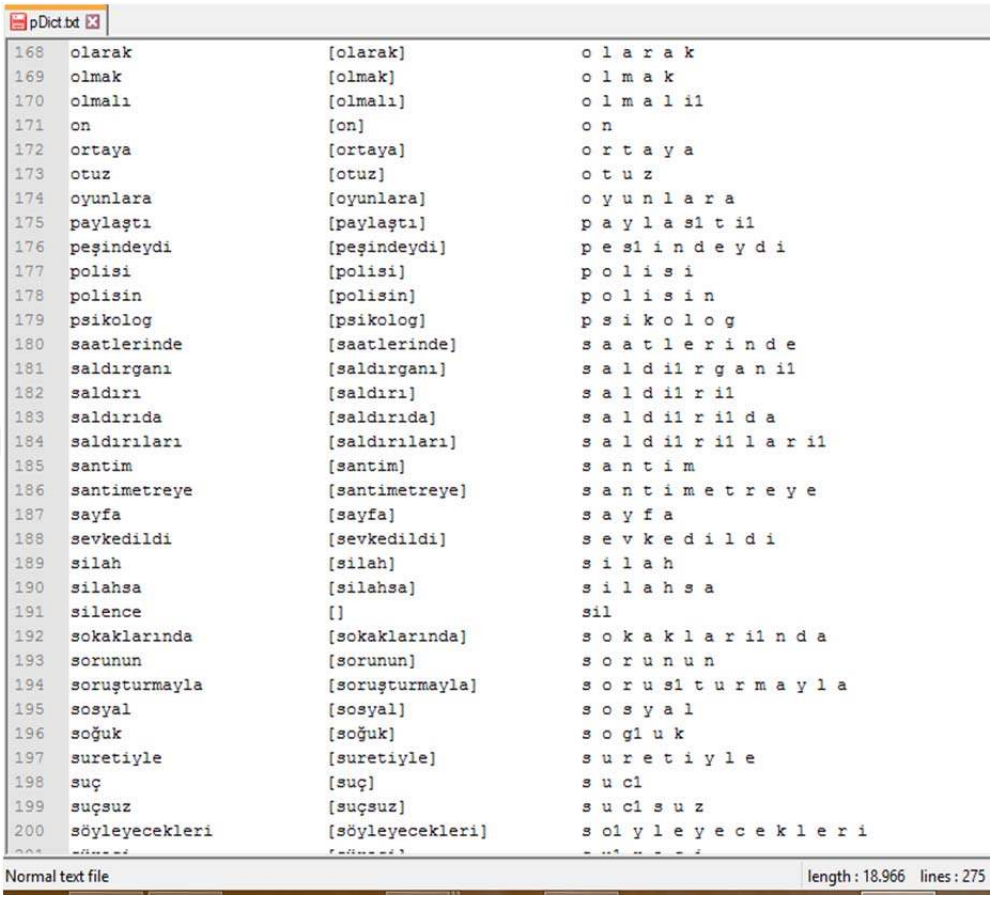
Obtained in the ubuntu programme with the following code in HTK to create a Pronunciation Dictionary list (pDict.list).
(http://www.voxforge.org/home/dev/acousticmodels/linux/create/htkjulius/tutorial/data-prep/step-2).

- HDMan -A -D -T 1 -m -w word.list –n phonemes -i –l dlog pDict../dictionary/CukurovaDict.txt

pDict: Grammar pronunciation dictionary to create and improve a phonetically balanced Acoustic Language Model.

phonemes: List of alphabets (mono-phoneme) for that language (Figure 3.9.).

Word	**Word symbols (phoneme)**	**Phonetic expansions**
paylaştı	[paylaştı]	p a y l a s1 t i1
açıköğretim	[açıköğretim]	a c1 i1 k o1 g1 r e t i m
baygınlık	[baygınlık]	b a y g i1 n l i1 k
genç	[genç]	g e n c1
hüzün	[hüzün]	h u1 z u1 n



pDict.txt

168	olarak	[olarak]	o l a r a k
169	olmak	[olmak]	o l m a k
170	olmalı	[olmalı]	o l m a l ı1
171	on	[on]	o n
172	ortaya	[ortaya]	o r t a y a
173	otuz	[otuz]	o t u z
174	oyunlara	[oyunlara]	o y u n l a r a
175	paylaştı	[paylaştı]	p a y l a s1 t ı1
176	peşindeydi	[peşindeydi]	p e s1 i n d e y d i
177	polisi	[polisi]	p o l i s i
178	polisin	[polisin]	p o l i s i n
179	psikolog	[psikolog]	p s i k o l o g
180	saatlerinde	[saatlerinde]	s a a t l e r i n d e
181	saldırganı	[saldırganı]	s a l d ı1 r g a n ı1
182	saldırı	[saldırı]	s a l d ı1 r ı1
183	saldırıda	[saldırıda]	s a l d ı1 r ı1 d a
184	saldırıları	[saldırıları]	s a l d ı1 r ı1 l a r ı1
185	santim	[santim]	s a n t i m
186	santimetreye	[santimetreye]	s a n t i m e t r e y e
187	sayfa	[sayfa]	s a y f a
188	sevkedildi	[sevkedildi]	s e v k e d i l d i
189	silah	[silah]	s i l a h
190	silahsa	[silahsa]	s i l a h s a
191	silence	[]	sil
192	sokaklarında	[sokaklarında]	s o k a k l a r ı1 n d a
193	sorunun	[sorunun]	s o r u n u n
194	soruşturmayla	[soruşturmayla]	s o r u s1 t u r m a y l a
195	sosyal	[sosyal]	s o s y a l
196	soğuk	[soğuk]	s o g1 u k
197	suretiyle	[suretiyle]	s u r e t i y l e
198	suç	[suç]	s u c1
199	suçsuz	[suçsuz]	s u c1 s u z
200	söyleyecekleri	[söyleyecekleri]	s o1 y l e y e c e k l e r i

Normal text file length : 18.966 lines : 275

Figure 3.10. Pronunciation Dictionary

3.2. Method

In this section, information will be given about the speech recognition-based models and sub-models that we compared on the dataset that we created.

3.2.1. Deep Neural Network

DNN model is an Neural Network with multiple layers between the input and destination (output) layers. (Bengio, 2009). It can solve and analysis the complicated non-linear and uncomplicated linear relationships. The model moves

through the between input and output layers calculating the probability of each output like a Feed Forward Neural Network (FFNN) structure.

DNN architectures generate compositional models where the object is expressed as a layered composition of primitives. (Szegedy, Toshev and Erhan, 2013). The extra layers enable composition of features from lower layers, potentially modeling complex data with fewer units than a similarly performing shallow network. (Bengio, 2009).

DNN model has nodes are little parts of the system, and they are like neurons of the human brain. When a stimulant hitting them, a process takes place between these nodes. Some of these nodes are connected and marked, and some are not, but mostly, nodes are grouped into layers. Firstly, the DNN creates a map of virtual neurons and assigns random numerical values, or "weights", to connections between them. The input nodes and weights are multiplied and return an output between 0 and 1 values is return. If the system could not fully recognize a particular pattern, an algorithm structure would set the weights (Hof, 2018). That way the model structure can make specific parameters more effective, until it determines the accurate mathematical manipulation to exactly process the model data.

A DNN model has to process layers of data throughput between the input layer and output layer to resolve a task. For a neural network system to be considered as a deep neural network, it is sufficient for the number of layers to be more than two. This value is the number of layers the model needs to complete the process.

In the following figure shows that (Figure 3.11) a basically DNN model has an input layer, two hidden layers and a destination (output) layer.

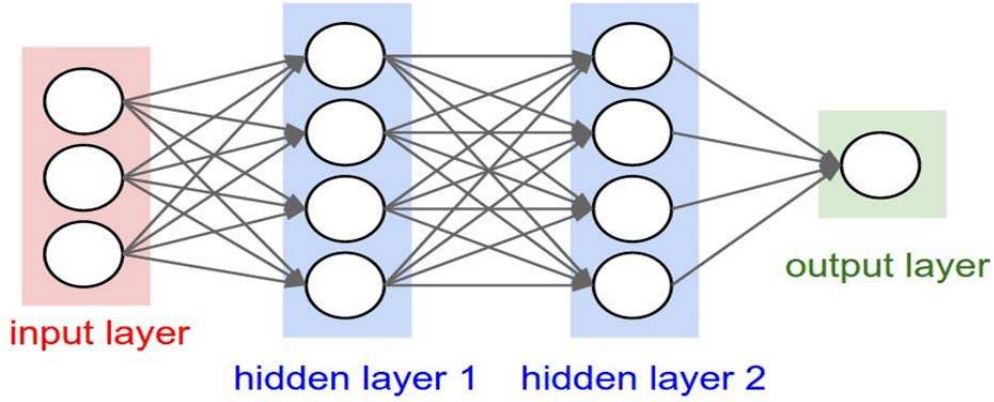


Figure 3.11. A general model of a Deep Neural Network

3.2.2. Recurrent Neural Network

Recurrent Neural Network (RNN) is the class of Artificial Neural Networks in which connections between nodes form a directed loop. This allows it to exhibit dynamic temporal behavior. Unlike DNN, RNN can use their input memory to process arbitrary rows of inputs. This attribute makes RNNs more usable method for speech recognition.

The main purpose of RNN is to use consecutive information. In a traditional neural network, we assume that all inputs and outputs are independent from each other. RNNs are called repetitive because they perform the same task for each element of an array, output depends on previous calculations. Another way to think of RNNs is that they carry "memory," which gathers information about what has been calculated so far.

In the following equations, $x^{(t)}$, $h^{(t)}$,and $y^{(t)}$ represent input, hidden and output values in the t-th time state (Zuo, Shuai, Wang, Liu, Wang, & Wang, 2016).

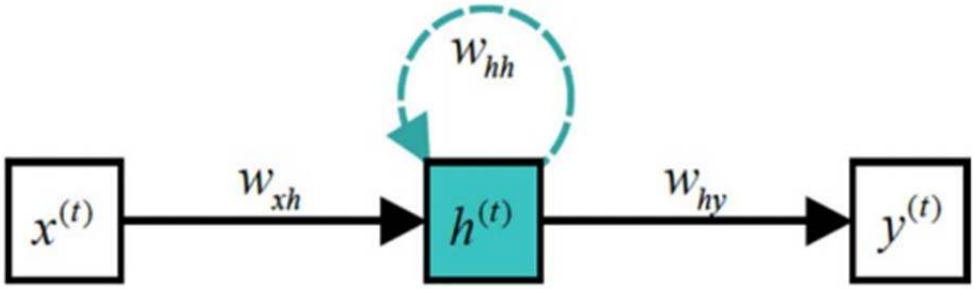


Figure 3.12. Structure of RNN (Zuo, Shuai, Wang, Liu, Wang, & Wang, 2016).

As illustrated in the Figure 3.12, calculating hidden state in a time state *t* requires hidden state of previous time state and input state at the current time state. Thus, $h^{(t)}$ and $y^{(t)}$ in a certain time step can be calculated as:

$$h^{(t)} = \psi_h\left(W_{hh}h^{(t-1)} + W_{xh}x^{(t)} + b_h\right) \quad (3.1)$$

$$y^{(t)} = \psi_y\left(W_{hy}h^{(t)} + b_y\right) \quad (3.2)$$

In the equations, W_{hh}, W_{xh} and W_{hy} are parameters to be learned. The ψh and ψy are non-linear activation functions. RRNs are not successful at learning dependencies in long sequences. Thus, LSTMs were developed (Zuo, Shuai, Wang, Liu, Wang, & Wang, 2016).

3.2.2.1. Long Short Term Memory

It is a special type of RNN that can learn about long-term dependence. LSTMs are clearly designed to avoid the problem of long-term dependence. They are widely used today because they work very well in a wide variety of problems.

All RNNs are in the form of a RNN module chain. In standard RNNs, this repeating module will have a very simple structure like a single tanh layer.

An LSTM structure includes LSTM units (cells). These cells can remember extended or short time epochs. The main idea of to this capability is that it does not use any activation functionality in its repeated fragments. Therefore, the stored value can not changed recursively and the slope is not lost while training by back propagation over period.

LSTM cells are usually implemented in "blocks" including several cells. This form is typical of DNNs and simplify applications by parallel hardware.

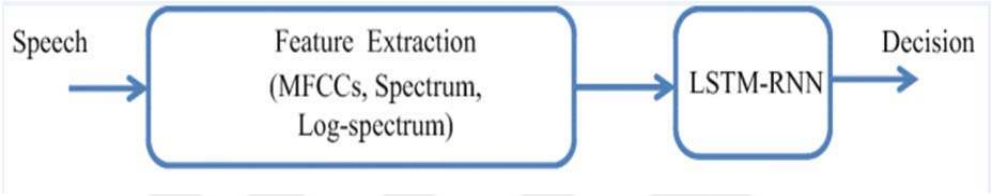


Figure 3.13. Structure of LSTM in Speech Recognition.

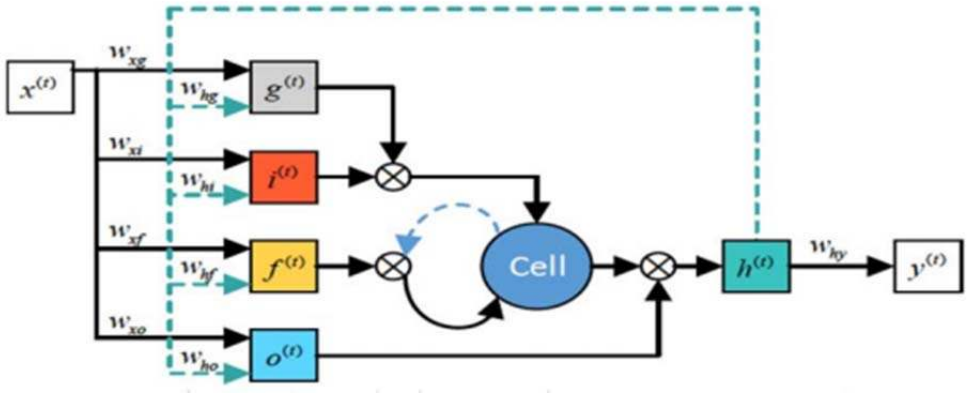


Figure 3.14. Structure of LSTM (Zuo, Shuai, Wang, Liu, Wang, & Wang, 2016).

LSTM introduced a cell state *c(t)* and four gates whose names are input gate *i(t)*, output gate *o(t)*, forget gate *f(t)* and input modulation gate *g(t)*. To calculate a value between 0 and 1 these LSTM gates perform logistic function. Replication is applied with this value to partially allow or deny information to enter

and exit from memory. As a sample, an "input" port controls how much a new value flows on memory struct. A "forget" gate checks the extent to which value stay in memory. An "output" gate checks how much that value in memory is used to calculate the output activation of the block.

At the t-th state, LSTM computes a series of equations as shown below.

$$i^{(t)} = \sigma\left(W_{hi}h^{(t-1)} + W_{xi}x^{(t)} + b_i\right) \quad (3.3)$$

$$f^{(t)} = \sigma\left(W_{hf}h^{(t-1)} + W_{xf}x^{(t)} + b_f\right) \quad (3.4)$$

$$o^{(t)} = \sigma\left(W_{ho}h^{(t-1)} + W_{xo}x^{(t)} + b_o\right) \quad (3.5)$$

$$g^{(t)} = \phi\left(W_{hg}h^{(t-1)} + W_{xg}x^{(t)} + b_g\right) \quad (3.6)$$

$$c^{(t)} = f^{(t)} \odot c^{(t-1)} + i^{(t)} \odot g^{(t)} \quad (3.7)$$

$$h^{(t)} = o^{(t)} \odot \phi\left(c^{(t)}\right) \quad (3.8)$$

$$y^{(t)} = \psi\left(W_{hy}h^{(t)} + b_y\right) \quad (3.9)$$

3.2.2.2. Connectionist Temporal Classification

Connectionist temporal classification (CTC) algorithm is used to make the alignment between input and output in a sequence, and basically has a structure similar to the Hidden Markov Model (HMM). It is used to train RNNs such as LSTM networks.

CTC can be used in different roles such as recognizing phonemes in speech recognition area or recognition the on-line handwriting with in digital image processing (Liwicki, Graves, and Bunke and Schmidhuber, 2007). It refers to the outputs and scoring, and is independent of the basically a neural network model structure.(Graves, Fernandes and Gomez, 2006).

CTC technique is an algorithm used to match the dynamic length **'x'** array to the dynamic length **'y'** array. (Graves, Fernandes, and Gomez and Schmidhuber, 2006). The length of the **'y'** array in the output is |y|, we can use this algorithm to align in cases that are shorter than input. In order to make the alignment indicated by **'a'**, "-" symbol is used, which indicates the space.
With a mathematical formula:

$$P(a|x) = RNN(x) \tag{3.10}$$

In this formula, **'x'** sequence and **'a'** sequence have the same length, and the RNN function can be performed by different methods.(Hochreiter, Schmidhuber, 1997).

If all the outputs are summed and normalized:

$$P(y|x) = \sum_{a \in \beta^{-1}(y)} p(a|x) \tag{3.11}$$

The β function discards the gaps in the output and repeats it with a new array. For example, **β(a-ab-) = β(-aa - abb)** and each **'a'** denotes a different alignment.

The probability P (y|x) can be optimized by considering all alignments using dynamic programming (3.11). (Graves, Fernandes, and Gomez and Schmidhuber, 2006).

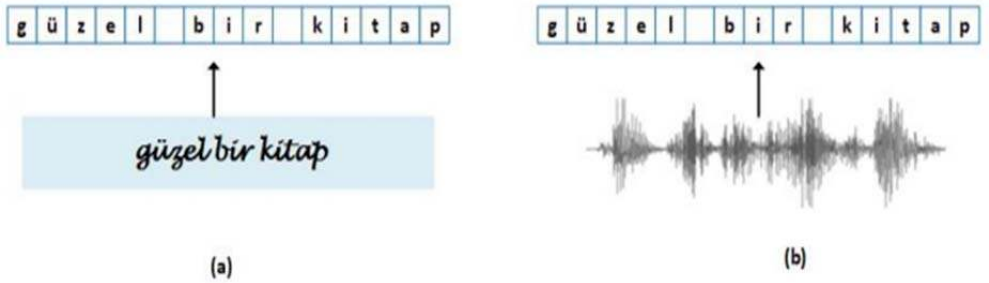


Figure 3.15. In order to train a model that recognizes handwriting, the model must learn the match between writing and characters (a). The same problem is valid for speech recognition and the alignment between sound and text must be found (b). (Asefisaray, B., 2018).

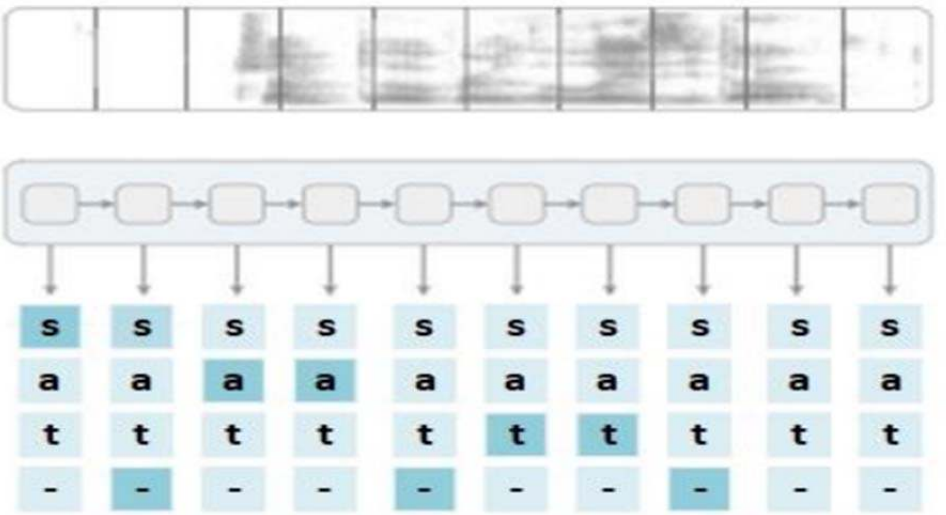


Figure 3.16. Possible alignments for the "saat" word are taken into account by the CTC, and the score for each alignment is calculated. (Asefisaray, B., 2018).

The CTC function and its different versions have been applied in many speech recognition systems, with successful results. However, all these studies using a powerful language model to improve recognition results and the acoustic model alone cannot learn the language model. This is because the CTC function assumes independence between inputs such as HMM and cannot model long dependencies. (Chan, Jaitly, and Le and Vinyals, 2015).

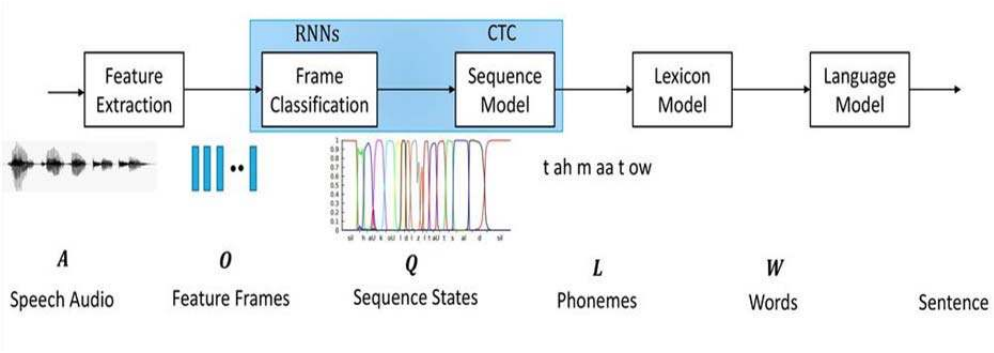


Figure 3.17. Structure of CTC in Speech Recognition.

4. RESEARCH AND DISCUSSION

4.1. Experimental Setups

In this section, experimental setups and experimental results are presented. C# and HTK programs were used for whole feature extraction phases; word - sentences extraction, and phonemic (monophoneme and threephoneme) structures. The proposed method consists of two models which they are DNN and RNNs LSTMs were trained and tested under Ubuntu 16.04 operating system with Python Programming on Tensorflow library.

Recording of voice sentences were created by using Turkish News TV Channels on Audacity program. The voice signals were recorded with sampling rate of 16,000 Hz, sample format of 16 bit, channels of one single channel (mono) formats were used and also saved as real number to avoid data loss. Our dataset' s size has 1039 sentences, also 8491 words were created from all sentences. 839 utterances and 200 utterances were used for training and testing, respectively.

This implementation can be divided into three main sections, the first one is the process of creating the speech dataset to obtain the Turkish News Channels, thereafter the second process is training the speech dataset on models using the DNN and LSTM speech model techniques. The final section is validating to recognition the speech sentence model according to the different hyperparameters values.

4.1.1. Hyperparameters

LSTM and DNN have several hyperparameters that affect the performance of deep learning systems regardless of the area of application. These hyperparameters are values that change LSTMs and DNN structure, and can help achieve better performance with the same data set. The number of layers and the number of units were used as hyperparameters in this thesis. The basic and most important hyperparameters of these models are listed and explained below:

• **Number of Layers:** This hyperparameter controls the number of layers of which deep learning systems will be built. When the number of layers are increased, the deep learning could better handle variations in the feature space but also complexity of the structure increases. (Tüfekci, Z., Dokuz, Y., 2020). Increasing the number of layers generally increases success while at the same time increasing the computation time proportionally. Therefore, the number of layers are special hyperparameters that need to be considered when building the model. (Hızlısoy, S., 2020). In this study, the number of layers were chosen from 1 to 6 on LSTM and DNN models shows that in the following figures (Figure 4.1 and Figure 4.2).

• **Number of Units:** This hyperparameter controls the number of cells that will be constructed on models. When the number of cells are increased, the more backward dependencies could be handled by the deep learning system, but also it increases complexity. (Tüfekci, Z., Dokuz, Y., 2020). Increasing the number of cells generally increases success while at the same time increasing the computation time proportionally. Therefore, the number of units are special hyperparameters that need to be considered when building the model. (Hızlısoy, S., 2020). In this study, the number of cells were chosen from 50 to 300 on LSTM and DNN models shows that in the following figures (Figure 4.1 and Figure 4.2).

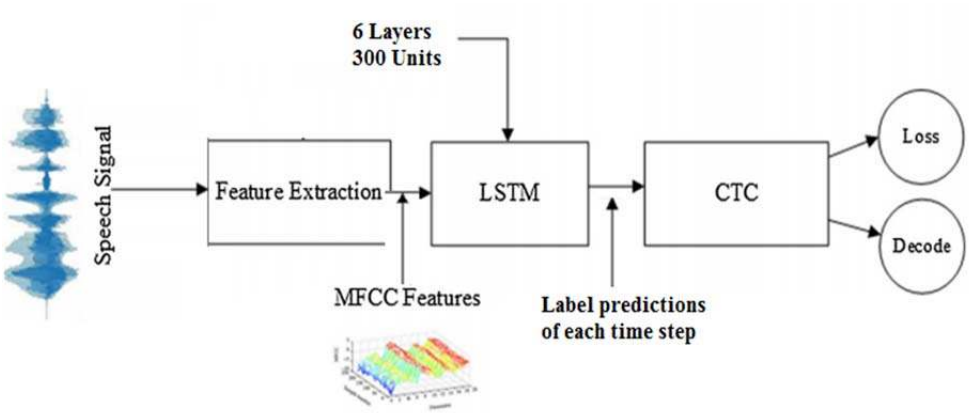


Figure 4.1. Utilized LSTM – CTC speech recognition method for this thesis

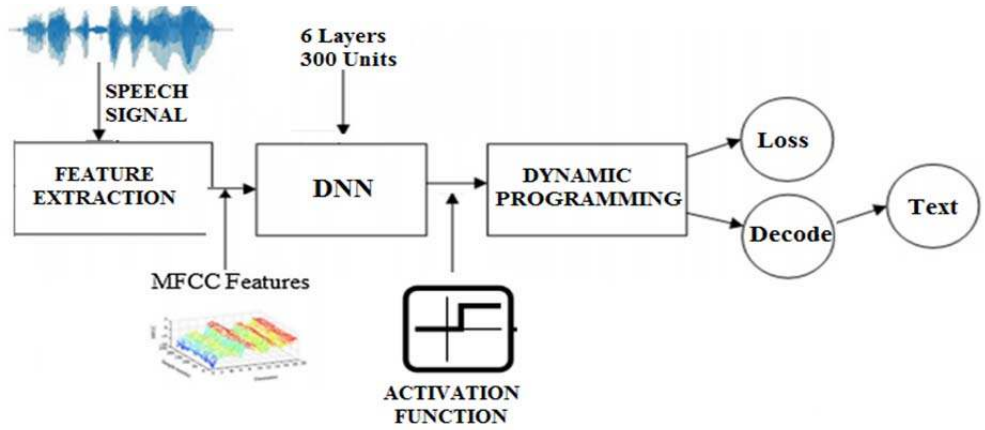


Figure 4.2. Utilized DNN speech recognition method for this thesis

• **Number of Epoch**: Epoch indicates the number of times the training process will be repeated. The number of times that all samples in the dataset will be trained is adjusted by making changes over this value. (Hızlısoy, S., 2020).

• **Learning Rate:** The learning rate is a hyperparameter that controls how much to change the model in response to the estimated error each time the model weights are updated. This hyperparamater value checks how rapidly the network model is adapted to the issue. While more training periods occur in small learning rates, larger learning rates cause rapid changes and require less training periods. Specifically, the learning rate is a configurable hyperparameter used in the training of neural networks that has a small positive value, often in the range between 0.0 and 1.0. The "learning rate" parameter (0.01) were employed in this study.

• **Number of Train and Test Dataset:** Number of training and validating dataset affects the success of a neural network. Out of 1039 sentences created in this thesis, 839 were used for train, 200 of which were used for validating.

Table 4.1. Number of training and validation sentences dataset

MODEL	#OF TOTAL SENTENCES	#OF TRAIN SENTENCES	#OF TEST SENTENCES
DNN	1039	839	200
LSTM	1039	839	200

• **Batch Size**: This value is used to get small groups of data to be trained. During model training, not all of the data is simultaneously inserted into the model. Because, when all data is added to the training at once, it requires very large data processing capacity, therefore the training is carried out in this way by taking as many parts as desired from the data set with this value. (Hızlısoy, S., 2020).

• **Activation Function:** Activation functions are mathematical equations that determine the output of a neural network. The function is attached to each neuron in the network, and determines whether it should be activated or not, based on whether each neuron's input is relevant for the model's prediction. The most used activation functions are softmax, sigmoid, rectified linear unit (Relu), and hyperbolic tangent (tanh). Sigmoid, which is the activation function frequently used in classification problems in artificial neural networks, converts incoming inputs into outputs between 0 and 1. Generally, the sigmoid function is used in studies with two classes, while the softmax function is used in studies with more classes. The hyperbolic tangent activation function converts incoming inputs into outputs between -1 and 1. The relu is a function that converts negative values to 0, and keeps other values unchanged. (Hızlısoy, S., 2020).

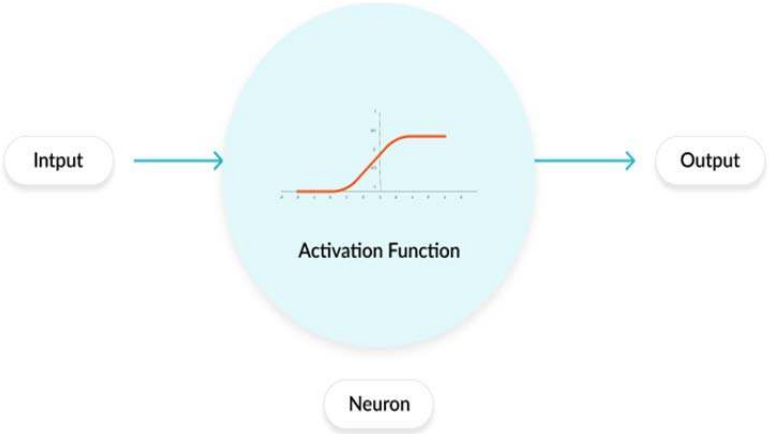


Figure 4.3. Activation Function Within Neural Networks.

4.2. Experimental Results

4.2.1. Effect of Number of Layers on LSTM

In this proving of thesis part, effect of number of layers on speech recognition performance were evaluated on LSTM. Number of cells (units), batch size, and number of samples were set to 200, 32, and 4096, respectively. Number of layers were selected from 1 to 6, and Label Error Rate (LER) and Test Cost were calculated. The effect of number of layers were presented in Figure 4.4.

As can be seen in Table 4.2, when number of layers increase in our setup, the deep networks become more accurate for both LER and Test cost. However, after 4 layers, the increase of number of layers decreases the performance of the deep network. The best performances were observed at 4 layers followed with 3 layers, and 1 layer and 6 layers LSTM perform worst for speech recognition task.

Table 4.2. Effect of number of layers on test datasets on LSTM Model.

Number of Layers	Test Cost	Test LER
1	58.116	49.40%
2	49.725	42.25%
3	44.438	38.30%
4	**41.985**	**37.40%**
5	52.174	44.10%
6	55.206	47.90%

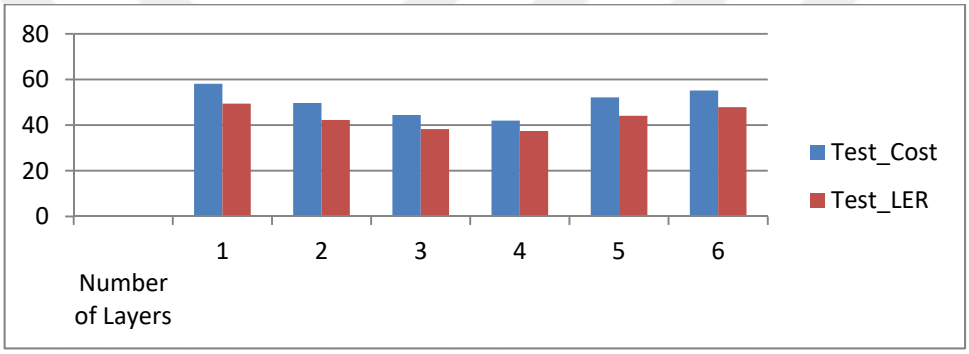


Figure 4.4. Effect of number of layers on test datasets on LSTM Model.

4.2.2. Effect of Number of Layers on DNN

In this proving of thesis part, the effect of number of layers on speech recognition performance were evaluated on DNN. Number of cells (units), batch size, and number of samples were set to 200, 32, and 4096, respectively. Number of layers were selected from 1 to 6, and LER and Test Cost were calculated. The effect of number of layers is presented in Table 4.3 and Figure 4.5.

As can be seen in Table 4.3, as the number of layers increases in our setup, the deep networks become more accurate for both LER and Test cost. However, after 5 layers, the increase of number of layers decreases the performance of the deep network. The best performance was observed at 5 layers followed with 4 and 3 layers, and 1 layer and 2 layers DNN perform worst for speech recognition task.

Table 4.3. Effect of number of layers on test datasets on DNN Model.

Number of Layers	Test Cost	Test LER
1	72.524	62.35%
2	63.682	55.45%
3	56.741	50.90%
4	54.954	49.30%
5	**53.035**	**48.25%**
6	60.782	53.45%

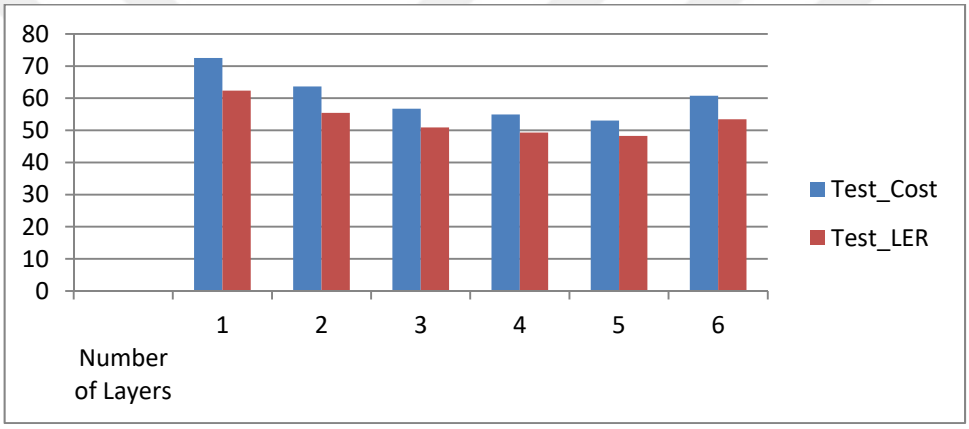


Figure 4.5. Effect of number of layers on test datasets on DNN Model.

4.2.3. Effect of Number of Units on LSTM

In this proving of thesis part, the effect of number of units on speech recognition performance was evaluated on LSTM. Number of layers, batch size, and number of samples were set to 4, 32, and 4096, respectively. Number of units were selected as 50, 100, 150, 200, 250 and 300, and LER and Test Cost were calculated. The effect of number of units is presented on Table 4.4 and Figure 4.6.

As can be seen on Table 4.4, as the number of units increases in our setup, the accuracy of the deep networks increases too for both LER and test cost. However, after 250 units, the accuracy keeps constant and does not increase. The best performances are observed for the number of units of 250 and 300.

Table 4.4. Effect of number of units on test datasets on LSTM Model.

Number of Units	Test Cost	Test LER
50	64.233	53.75%
100	53.415	45.30%
150	47.744	41.60%
200	41.985	37.40%
250	37.248	**34.25%**
300	**36.312**	**34.25%**

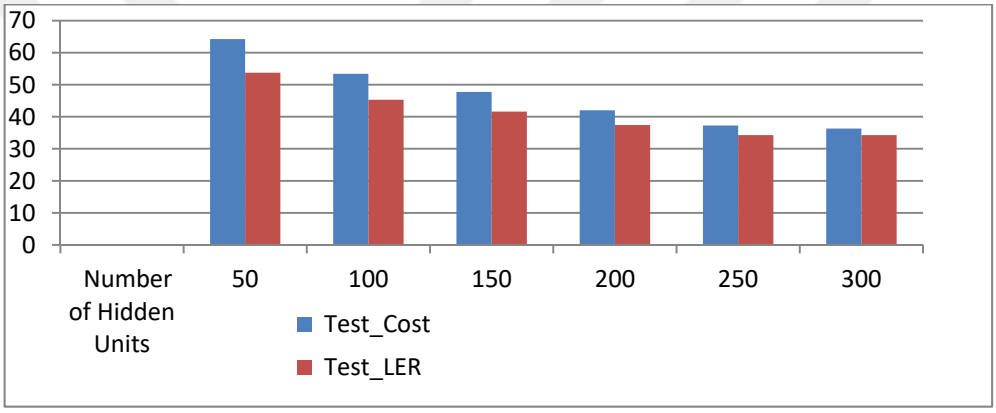


Figure 4.6. Effect of number of units on test datasets on LSTM Model.

4.2.4. Effect of Number of Units on DNN

In this proving of thesis part, the effect of number of units on speech recognition performance were evaluated on DNN. Number of layers, batch size, and number of samples are set to 5, 32, and 4096, respectively. Number of units were selected as 50, 100, 150, 200, 250 and 300, and LER and Train Cost were calculated. The effect of number of units were presented in Figure 4.7.

As can be seen on Table 4.5, as the number of units increases in our setup, the accuracy of the deep networks increases too for both LER and test cost. However, after 250 units, the accuracy decreases. The best performances were observed for the number of units of 200 and 250.

Table 4.5. Effect of number of units on test datasets on DNN Model.

Number of Units	Test Cost	Test LER
50	78.524	68.35%
100	69.682	61.45%
150	59.741	53.90%
200	53.035	48.25%
250	**49.854**	**46.70%**
300	55.603	51.45%

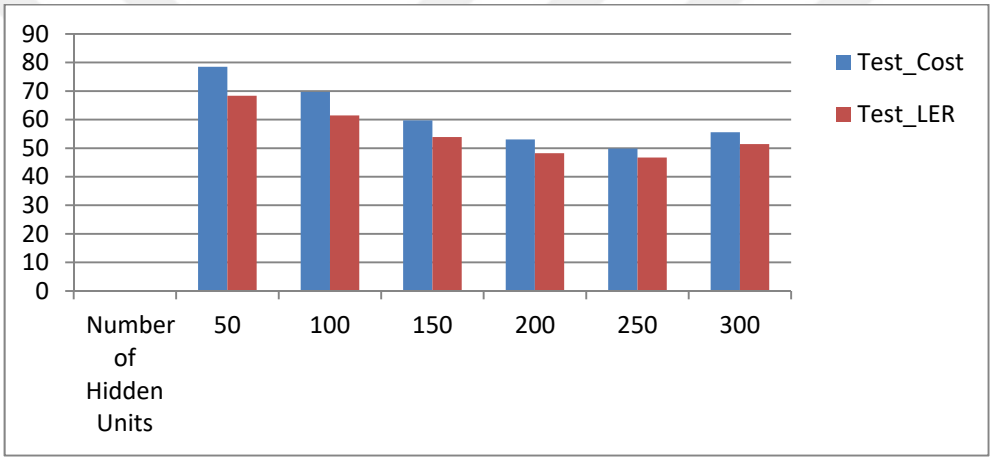


Figure 4.7. Effect of number of units on test datasets on DNN Model.

4.2.5. Comparison of LSTM and DNN models in terms of number of layer effect

In this proving of thesis part, the effect of number of layers on speech recognition performance were compared with LSTM and DNN. Number of units, batch size, and number of samples are set to 200, 32, and 4096, respectively both two models. Number of layers were selected from 1 to 6, and LER and Test Cost were calculated. The effect of number of layers were presented in Table 4.6 and Figure 4.8.

When number of layers were increased on LSTM in our setup, the performance increases until 4 layers. After the 4 layers, the performance gets worse. When number of layers were increased on DNN, the performance increases until 5 layers. After the 5 layers, the performance gets worse in our setup. It has been observed that number of layers has been effective in both models, the most effective number of layers is different. While lowest LER is % 37.40 for LSTM model, on DNN model it is % 48.25. LSTM is proportionally about 11 points better than DNN.

Table 4.6. Comparison of LSTM and DNN models in terms of number of layer effect

	LSTM		**DNN**	
Number of Layers	**Test Cost**	**Test LER**	**Test Cost**	**Test LER**
1	58.116	49.40%	72.524	62.35%
2	49.725	42.25%	63.682	55.45%
3	44.438	38.30%	56.741	50.90%
4	**41.985**	**37.40%**	54.954	49.30%
5	52.174	44.10%	**53.035**	**48.25%**
6	55.206	47.90%	60.782	53.45%

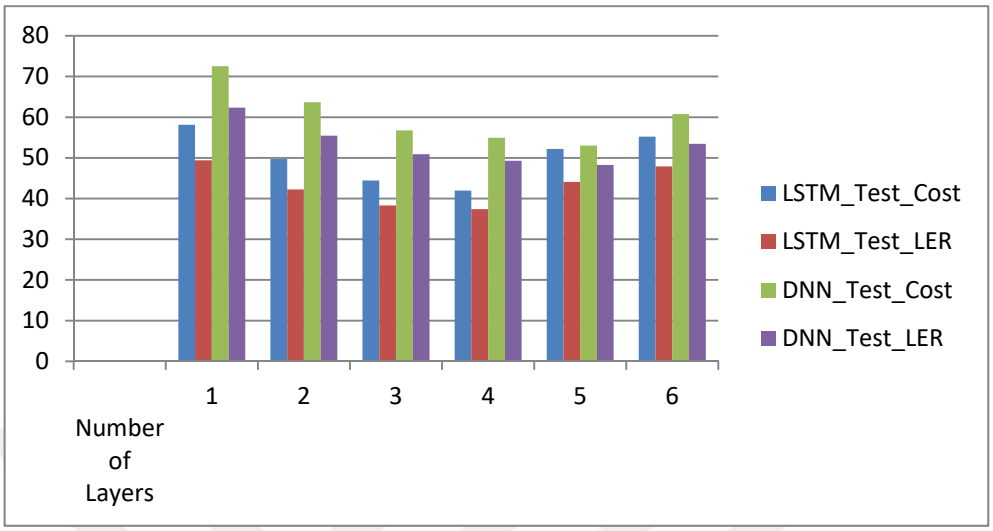


Figure 4.8. Comparison of LSTM and DNN models in terms of number of layer effect.

4.2.6. Comparison of LSTM and DNN models in terms of number of units effect

In this proving of thesis part, the effect of number of units on speech recognition performance were compared with LSTM and DNN. Number of batch size, and number of samples are set to 32, and 4096, respectively and number of layers set to models which they were most successful for our setup. Number of units were selected as 50, 100, 150, 200, 250 and 300, and LER and Test Cost were calculated. The effect of number of units were presented on Table 4.7 and Figure 4.9.

When number of cells were increased with fixed layer number on LSTM in our setup, the error rate of the system (LER) gets decrease from % 37.40 to % 34.25 with rate of % 8.4, later on error rate fixed at % 34.25. When number of cells were increased with fixed layer number on DNN in our setup, the error rate of the system (LER) gets decrease from % 48.25 to % 46.70 with rate of % 3.2, and later on performance decreased. For our setup, While best performance were observed rate at % 34.25 LER, with number of layers 4 and number of cells 250

on LSTM model, best performance were observed rate at % 46.70 LER, with number of layers 5 and number of cells 250 on DNN model. When the evaluation results of this study was investigated, number of units have effect on the performance of LSTMs and DNN models for speech recognition based systems.

Table 4.7. Comparison of LSTM and DNN models in terms of number of units effect.

	LSTM		**DNN**	
Number of Units	**Test Cost**	**Test LER**	**Test Cost**	**Test LER**
50	64.233	53.75%	78.524	68.35%
100	53.415	45.30%	69.682	61.45%
150	47.744	41.60%	59.741	53.90%
200	41.985	37.40%	53.035	48.25%
250	37.248	**34.25%**	**49.854**	**46.70%**
300	**36.312**	**34.25%**	55.603	51.45%

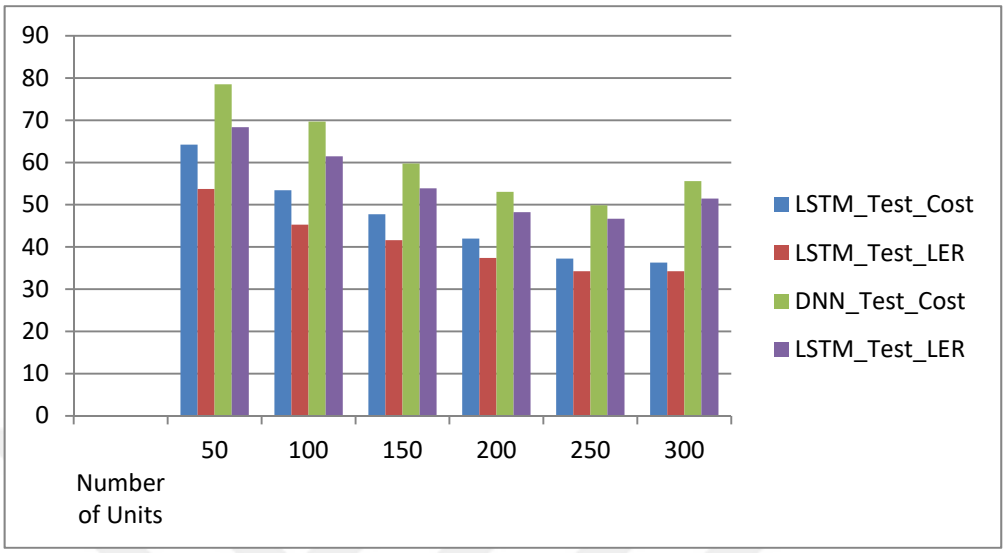


Figure 4.9. Comparison of LSTM and DNN models in terms of number of units effect.

4.3. Experimental Comparions With Other Studies

Table 4.8. Results of different recognition approach with Turkish Speech Datasets

Name Of Authors	Recognition Models	Dataset_Name	Dataset Size (h)	Label Error Rate
Arısoy and Saraçlar (2018)	TDNN-LSTM	Turkish Broadcast News Transcription and Retrieval (March 2006-May 2007)	194 hours	9.83 %
Arısoy and Saraçlar (2018)	DNN	Turkish Broadcast News Transcription and Retrieval (March 2006-May 2007)	194 hours	13.42 %
Kimanuka and Buyuk (2018)	GMM-HMM	Turkish Mobile Device Speech Dataset (2016)	6.1 hours	13.21 %
Kimanuka and Buyuk (2018)	DNN-HMM	Turkish Mobile Device Speech Dataset (2016)	6.1 hours	8.47 %
Arslan and Barışçı (2019)	LSTM	METU 1.0 Dataset	8.3 hours	24.82 %

When the studies applied with different Turkish speech datasets were examined on; Arısoy - Saraçlar (2018) using the 194-hours Turkish Broadcast News Transcription and Retrieval dataset, Kimanuka - Buyuk (2018) using the approximately 6-hours Turkish Mobile Device Speech Dataset, and Arslan - Barışçı (2019) using the approximately 8-hours METU 1.0 Dataset, It has been observed in the studies on Table 4.8 shows that they had very successful results on LSTM and DNN models (low label error rates) by using large datasets. Information on datasets used in these studies was given in section 2.1.

Comparing with other studies, one of the most important reasons for the high word error rates in our study; currently, our models have less dataset for train and test than those of the studies that obtain speech recognition based systems performances. The size of the available dataset is very important in speech recognition based studies. When the size of the dataset increases, hyperparameter values such as the number of layers and the number of units can also be more increased and better results can be obtained.

5. CONCLUSION AND FUTURE WORK

Creating a speech recognition database for Turkish language and to share this dataset in the web environment as a source for other speech recognition based studies are aimed in this thesis. For this purpose, approximately 2 hours of Turkish Broadcasts News Speech Dataset from a total of 1039 sound sentence files with their corresponding text transcripts were created. Also 8491 words (4101 different words), mono-phoneme and three-phoneme structures were created from all sentences.

Additionally, we investigated and compared the effect of the number of layers and number of cells on RNNs LSTMs and DNN hyperparameters on Turkish Broadcasting News Speech Dataset that we created. LSTM and DNN have several hyperparameters that affect the performance of deep learning systems regardless of the area of application. These hyperparameter values that change LSTMs and DNN structure, and can help achieve better performance with the same dataset.

When number of layers were increased for LSTM, the performance increased until 3 and 4 layers, after 3 and 4 layers, the performance gets worse in our setup. When number of layers were increased for DNN, the performance increased until 4 and 5 layers, after 4 and 5 layers, the performance gets worse in our setup. In both models, the success rate decreases after a certain layer due to the situation of memorization.

When number of units were increased with fixed layer number for LSTM, the performance of the system gets better and later on performance rate were fixed. When number of units were increased with fixed layer number for DNN, the performance of the system gets better and later on performance rate were decreased.

For our setup, While best performance were observed rate at % 34.25 Label Error Rate, with number of layers 4 and number of cells 250 on LSTM model, best performance were observed rate at % 46.70 Label Error Rate, with number of layers 5 and number of cells 250 on DNN model. LSTM produced more successful results than DNN. When previous studies on the effect of LSTM model on speech recognition systems were examined, it was expected that it gave better results and therefore LSTM would give more successful results in this study too.

Experimental results show that each parameter has its specific values for the selected number of training instances to provide lower error rates and better speech recognition performance. It is shown in this study that before selecting appropriate values for each LSTM and DNN parameters, there should be several experiments performed on the speech corpus to find the most eligible value for each parameter.

When the evaluation results of this study is investigated, all hyperparameters that we applied have effect on the performance of LSTMs and DNN for speech recognition systems.

For future work, larger dataset and better modelling strategies and hyperparameters can be used to obtain better performance. Currently, our models have less dataset for train and test than those of the studies that obtain speech recognition based models performances. Secondly the number of speech databases created for Turkish is very little. Prepared Speech databases are extremely important for researchers who are trying to develop a Turkish Speech Recognition System. Their number and diversity should increase.

As a result, this study set a precedent for creating a database for studies in Turkish speech recognition based areas.

The created dataset can be accessed from the following web extensions;

http://cusesveri.com/

https://gitlab.com/serhatok/cusesveri/-/tree/master

Figure 5.1. Çukurova University-Turkish Speech Recognition Database

REFERENCES

Abdulkarim, M. D., 2015. Noise Robust Speaker Recognition Under Unknown Noise Environment.

Arısoy, E., ve Saraçlar, M. (2018). Türkçe haber programları için konuşma Tanımanın Tekrar Gözden Geçirilmesi. In Proceedings of the IEEE 28. Sinyal İşleme ve İletişim Uygulamaları Konferansı (SİU), İzmir, Turkey

Asefisaray, B., 2018. Uçtan-Uca Konuşma Tanıma Modeli: Türkçe' deki Deneyler. p. 53-55

Bengio, Y., 2009. "Learning Deep Architectures for AI" (PDF). Foundations and Trends in Machine Learning. 1–127.

Chan, W., Jaitly, N., Le, Q. V., and Vinyals, O., 2015 "Listen, Attend and Spell". CoRR, 2015. abs/1508.01211.

Çömez, M. A., 2003. Large Vocabulary Continuous Speech Recognition For Turkish using HTK.

Dokuz, Y., 2020. Exploring Mini – Batch Sample Selection Strategies for Deep Learning Based Speech Recognition

Gaikwad, S., Gawali, B. W., & Yannawar, P. 2010. A review on Speech Recognition Technique. , pp. 16-24.

Graves, A., Fernandes, S., Gomez, F., 2006. "Connectionist temporal classification: Labelling unsegmented sequence data with recurrent neural networks". In Proceedings of the International Conference on Machine Learning,

Graves, A., Fernandes, S., Gomez, F., and Schmidhuber, J., 2006. Connectionist temporal classification: labelling unsegmented sequence data with recurrent neural networks. in Proceedings of the 23rd international conference on Machine learning. 2006. ACM.

Graves, A., Jaitly, N., & Mohamed, A. R. (2013b, December). Hybrid speech recognition with deep bidirectional LSTM. In 2013 IEEE workshop on automatic speech recognition and understanding (pp. 273-278). IEEE.

Graves, A., Mohamed, A. R., & Hinton, G. (2013, May). Speech recognition with deep recurrent neural networks. In 2013 IEEE international conference on acoustics, speech and signal processing (pp. 6645-6649). IEEE.

Hizlisoy, S., 2020. Music Emotion Recognition Using Convolutional Long Short Memory Deep Neural Networks.

Hochreiter, S. and Schmidhuber, J., 1997. Long short-term memory. Neural computation, 9(8): p. 1735-1780.

Hof, R. D., 2018. "Is Artificial Intelligence Finally Coming into Its Own?". MIT Technology Review.

Liwicki, M., Graves, A., and BUNKE, H., and SCHMIDHUBER, J. 2007. "A novel approach to on-line handwriting recognition based on bidirectional long short-term memory networks". In Proceedings of the 9th International Conference on Document Analysis and Recognition.

Patlar, F., 2009. A Continuous Speech Recognition System For Turkish Language Based On Triphone Model.

R. S. Arslan And N. BARIŞÇI, "Development of Output Correction Methodology for Long Short Term Memory-Based Speech Recognition, “Sustainability, vol. 11, 2019

Sepp Hochreiter; Jürgen Schmidhuber (1997). "LSTM can Solve Hard Long Time Lag Problems". Advances in Neural Information Processing Systems 9. Advances in Neural Information Processing Systems. Wikidata Q77698282.

Szegedy, C., and Toshev, A., and Erhan, D., 2013. "Deep neural networks for object detection". Advances in Neural Information Processing Systems: 2553–2561.

Tüfekci, Z., and Dokuz, Y., 2020. Investigation of the Effect of LSTM Hyperparameters on Speech Recognition Performance , European Journal of Science and Technology: p. 165.

U. A. Kimanuka And O. Büyük, "Turkish Speech Recognition Based on Deep Neural Networks," Süleyman Demirel Üniversitesi Fen Bilimleri Enstitüsü Dergisi , vol.22, pp.319-329, 2018

Yalçin, N., 2006. Konuşma Tanıma Teknolojisi Yardımıyla İlköğretim Birinci Sınıf Öğrencilerine İlkokuma Yazma Öğretimi İçin Bir Yazılım Geliştirme, Doktora Tezi, Gazi Üniversitesi Fen Bilimleri Enstitüsü, Ankara, 18-19

Yu, D., & Deng, L. (2016). Automatic Speech Recognition: A Deep Learning Approach. Springer

Zuo, Z., Shuai, B., Wang, G., Liu, X., Wang, X., Wang, B. (2016). Learning Contextual Dependence with Convolutional Hierarchical Recurrent Neural Networks. IEEE Transactions on Image Processing, 25, 2983-2996.

KALDI, http://kaldi asr.org/doc/examples.html

LDC, https://catalog.ldc.upenn.edu/byyear

LDC, https://catalog.ldc.upenn.edu/LDC2012S06

LDC, https://catalog.ldc.upenn.edu/LDC2004S01

LDC, https://catalog.ldc.upenn.edu/LDC98T29

LDC, https://catalog.ldc.upenn.edu/LDC2006S42

LDC, https://catalog.ldc.upenn.edu/LDC97S44

LDC, https://catalog.ldc.upenn.edu/LDC2007T19

LDC, https://catalog.ldc.upenn.edu/LDC2006S46

LDC, https://catalog.ldc.upenn.edu/LDC2017S25

LDC, https://catalog.ldc.upenn.edu/LDC2006S33

FlvTo, https://www.flvto.biz/tr/youtube-downloaded

VOXFORGE, http://www.voxforge.org/home/dev/acousticmodels/linux/create/htkjulius/tutorial/data-prep/step-2

CUSESVERİ, http://cusesveri.com/

GİTLAB, https://gitlab.com/serhatok/cusesveri/-/tree/master

CURRICULUM VITAE

Serhat OK was born in Adana, on 17 May 1993. He completed his elementary education at Barbaros İlkokulu and Süreyya Nihat Oral Ortaokulu and highschool education at 19 Mayıs Lisesi. He completed his Bachelor at department of Computer Engineering of Çukurova University Adana/TURKEY between 2012 and 2017. Since 2018, he has been working as a software engineer at Ziraat Teknoloji (Ziraat Bank) in Istanbul / TURKEY.

Mail Address: serxadok47@gmail.com