ÇUKUROVA UNIVERSITY
INSTITUTE OF NATURAL AND APPLIED SCIENCES

PhD THESIS

Yeşim DOKUZ

EXPLORING MINI-BATCH SAMPLE SELECTION STRATEGIES FOR DEEP LEARNING BASED SPEECH RECOGNITION

DEPARTMENT OF COMPUTER ENGINEERING

ADANA-2020

ABSTRACT

PhD THESIS

EXPLORING MINI-BATCH SAMPLE SELECTION STRATEGIES FOR DEEP LEARNING BASED SPEECH RECOGNITION

Yeşim DOKUZ

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

Supervisor : Assoc. Prof. Dr. Zekeriya TÜFEKCİ
Year: 2020, Pages: 97
Juries : Assoc. Prof. Dr. Zekeriya TÜFEKCİ
: Assoc. Prof. Dr. Sami ARICA
: Assoc. Prof. Dr. Umut ORHAN
: Assoc. Prof. Dr. Gülay TEZEL
: Assoc. Prof. Dr. Serdar YILDIRIM

This thesis aims to propose mini-batch sample selection strategies for deep learning based speech recognition systems. Deep learning based speech recognition systems became more prevalent and state-of-the-art system for speech recognition domain with the popularity and success of deep learning architectures. Recurrent Neural Networks (RNN) and Long Short-Term Memory (LSTM) RNN are widely and successfully utilized for the applications of speech recognition. Mini-batch gradient descent algorithm is generally accepted algorithm for training deep learning based speech recognition systems. Mini-batch gradient descent algorithm is a successful algorithm, but one of the main problems in mini-batch gradient descent is that the training samples are selected randomly for each mini-batch. In this thesis, mini-batch sample selection strategies are proposed to improve speech recognition accuracy of deep learning based speech recognition systems. Proposed strategies use meta features of speech corpuses, i.e. gender and accent features. Three types of sample selection strategies are proposed, i.e. gender adjusted strategies, accent adjusted strategies, and hybrid strategies that combine gender and accent adjusted strategies. The experimental results show that proposed strategies are beneficial for improving performance of deep learning based speech recognition systems.

Keywords: Mini-batch Gradient Descent, Sample Selection Strategies, Deep Learning, Speech Recognition, RNN

ÖZ

DOKTORA TEZİ

DERİN ÖĞRENME TABANLI SES TANIMA İÇİN MİNİ-YIĞIN ÖRNEK SEÇİM STRATEJİLERİNİN ARAŞTIRILMASI

Yeşim DOKUZ

ÇUKUROVA ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ
BİLGİSAYAR MÜHENDİSLİĞİ ANABİLİM DALI

Danışman : Doç. Dr. Zekeriya TÜFEKCİ
Yıl: 2020, Sayfa: 97
Jüri : Doç. Dr. Zekeriya TÜFEKCİ
: Doç. Dr. Sami ARICA
: Doç. Dr. Umut ORHAN
: Doç. Dr. Gülay TEZEL
: Doç. Dr. Serdar YILDIRIM

Bu tez çalışması derin öğrenme tabanlı ses tanıma sistemleri için mini-yığın örnek seçim stratejileri önermeyi amaçlamaktadır. Derin öğrenme tabanlı ses tanıma sistemleri, derin öğrenme mimarilerinin popülaritesi ve başarıları ile birlikte çok daha yaygın ve ses tanıma alanında en modern sistem olmuştur. Özyinelemeli Sinir Ağları (ÖSA) ve Uzun Kısa Dönem Hafıza ÖSA ses tanıma uygulamaları için oldukça yaygın ve başarılı bir şekilde kullanılan mimarilerdir. Mini-yığın gradyan iniş algoritması derin öğrenme tabanlı ses tanıma sistemlerinde kabul görmüş bir eğitim algoritmasıdır. Mini-yığın gradyan iniş algoritması başarılı bir algoritmadır ancak bu algoritmanın problemlerinden biri her bir mini-yığın için eğitim örneklerini rastgele seçmesidir. Bu tez çalışmasında, derin öğrenme tabanlı ses tanıma sistemlerinin ses tanıma doğruluğunu artırmak için mini-yığın örnek seçim stratejileri önerilmiştir. Önerilen stratejiler ses veri kümelerinde bulunan cinsiyet ve aksan özellikleri gibi meta özellikleri kullanmaktadır. Cinsiyet tabanlı stratejiler, aksan tabanlı stratejiler ve cinsiyet ve aksan tabanlı stratejileri birlikte kullanan hibrit stratejiler olmak üzere üç tür örnek seçim stratejisi önerilmiştir. Deneysel sonuçlar önerilen stratejilerin derin öğrenme tabanlı ses tanıma sistemlerinin performansını artırmak için faydalı olduğunu göstermiştir.

Anahtar kelimeler: Mini-yığın Gradyan İniş, Örnek Seçim Stratejileri, Derin Öğrenme, Ses Tanıma, ÖSA

EXTENDED ABSTRACT

Speech recognition is an important domain which tries to convert speech signals into their corresponding text transcriptions. Speech recognition is essential for converting voice records, controlling devices with speech inputs and for searching in speech records. Speech recognition is beneficial and important for many application areas, such as mobile communication, search engines, voice input systems and personal digital assistance systems.

Speech recognition has several challenges. First of all, the datasets are present for few languages. Second, great variations occur in speech inputs based on gender, age, accent and local language differences. Finally, background noise, recording device quality and environmental factors affect speech recognition performance.

On the other hand, deep learning is a sub-field of machine learning which uses several algorithms to model high-level abstractions in the data by using graphs that consists of multiple processing layers that are composed of linear or non-linear transformations. Deep learning also provides efficient use and selection of the features in the datasets which are more relevant with the desired output. Deep learning gained attention in researchers after its applications achieved better performance than state-of-the-art methods in the literature. With the recent success and popularity of deep learning, it is being utilized in many application areas, such as image recognition, natural language processing, time series analysis and speech recognition.

Deep learning architectures are being utilized in many application areas in speech recognition, such as speech enhancement, speaker verification, and voice activity detection. Among the most popular deep learning architectures, Deep Neural Networks (DNN), Convolutional Neural Networks (CNN), and especially Recurrent Neural Networks (RNN) and Long Short Term Memory (LSTM) RNN, which is a special kind of RNN, are widely preferred for speech recognition. DNN is generally utilized in speech recognition as a hybrid method for Hidden Markov

Model (HMM) speech recognition systems. CNN is generally used for image processing; however, it can be used in speech recognition as well. In speech recognition, CNN is mostly utilized for visual speech inputs, which is called as *spectrograms*. RNN and LSTM are successful at analyzing sequential inputs, and thus, they are utilized for time series feature inputs of speech signals. Apart from these architectures, other deep learning architectures are also utilized for speech recognition tasks, such as Deep Belief Networks and Deep Autoencoders.

RNN and LSTM are the most preferred deep learning architecture in speech recognition domain to achieve state-of-the-art and satisfying performance with their ability to process sequential and time series inputs. Especially LSTM provides high performance for speech recognition tasks by using a specialized gate-based architecture. Also, training algorithms are specialized for these architectures in which generally a variant of gradient descent optimization is utilized.

Gradient descent optimization is one of the most popular approach for training and optimizing deep learning models. It is based on backpropagation algorithm that propagates network output errors backward and updates parameters. Gradient descent optimization has three variants, batch gradient descent, stochastic gradient descent, and mini-batch gradient descent which differ in the usage of the amount of the data before performing an update on the parameters of deep learning models. In deep learning based speech recognition studies, mini-batch gradient descent is preferred due to its benefits over other alternatives.

Deep learning based speech recognition systems are successful systems for recognizing a speech input with respect to the input features. The performance of the deep learning based speech recognition systems is based on several factors, including a comprehensive speech corpus which contains the different characteristics of user types, such as gender, accent, speaker and linguistic contents. A wide range of features in a speech corpus is essential to increase the recognition performance to have a more reliable and accurate speech recognition system. One important factor to increase the performances of deep learning based speech recognition systems is to train these systems with more data that have a wide range of accent, gender, and speakers.

In this thesis study, we aim to improve performance of deep learning based speech recognition systems by proposing new methods for mini-batch sample selection which is an important part in training deep learning systems. We proposed several strategies for selecting mini-batch samples for speech recognition tasks to increase model performance of deep learning based speech recognition. In particular, we proposed three types of mini-batch sample selection strategies, i.e. gender adjusted strategies, accent adjusted strategies, and hybrid strategies which selects mini-batch samples using combination of gender and accent adjusted proposed strategies. For each type of mini-batch sample selection strategy, two further categories are explored, i.e. homogeneous and heterogeneous categories. In homogeneous strategies, only one meta feature is considered in one mini-batch, such as only female or only English speakers are selected for one mini-batch. In heterogeneous strategies, proportional presence of each meta feature is used, such as half female and half male speakers are selected for one mini-batch.

Selected gender and accent meta features are important factors on the performance of speech recognition systems. Gender and accent presence in speech recognition models would yield good or poor performance with respect to the utilized dataset characteristics. For this reason, in this thesis study, we used gender and accent meta information for selection of mini-batch samples. When gender feature is not used in balanced between female and male speakers, the speech recognition system could not have good performance. Similarly, when different accents of a language are not present in the speech recognition system, then variable accents in test samples could not be recognized properly. Following these information, we proposed gender and accent based mini-batch sample selection strategies which could increase deep learning based speech recognition performance.

The proposed three types of mini-batch sample selection strategies are developed to investigate the performance of deep learning based speech recognition systems based on mini-batch sample features and to monitor the effect of gender and accent information on training of deep learning systems. For deep

learning architecture, LSTM is preferred which is the most popular architecture in speech recognition systems. All of the proposed strategies are evaluated on the same deep learning system to fairly compare the strategies with each other and to determine the best strategy.

The experimental results of the proposed mini-batch sample selection strategies show that, homogeneous gender – heterogeneous accent hybrid strategy, homogeneous gender strategy and heterogeneous accent strategy outperform other alternative strategies in terms of train and test metrics. The reason for this result could be that, selecting homogeneous gender as one mini-batch samples provide deep learning systems to better model speech inputs with output texts. Homogeneous gender strategy selects samples of one mini-batch only with one gender. This provides closer gender features within a mini-batch and the deep learning based speech recognition systems could better learn the character utterances. Also, heterogeneous accent strategy had good performance which selects mini-batch samples with the presence of all accents. When all accents are present in every mini-batch, the deep learning based speech recognition system becomes more successful, because it could learn every possible accent and all varieties of the utterances of the characters within each mini-batch. Homogeneous gender – heterogeneous accent hybrid strategy has the best performance in comparison with the proposed mini-batch sample selection strategies. The reason for this result is, both homogeneous gender and heterogeneous accent strategies have good performance, and using a hybrid of these two strategies provide better speech recognition performance.

When all of the results are evaluated, we can conclude that, when gender and accent meta information are used for mini-batch sample selection, deep learning based speech recognition models could achieve better performance. Especially when using hybrid of successful mini-batch sample selection strategies further improves the performance of the systems.

GENİŞLETİLMİŞ ÖZET

Ses tanıma, ses sinyallerini uygun metin karşılıklarına çevirmeyi amaçlayan önemli bir alandır. Ses tanıma konuşma kayıtlarının dönüştürülmesi, cihazların ses girdileriyle kontrol edilmesi ve ses kayıtları içerisinde arama yapılması için gereklidir. Ses tanıma, mobil haberleşmeler, arama motorları, konuşma girdi sistemleri ve kişisel dijital asistan sistemleri gibi çeşitli uygulama alanları için faydalı ve önemlidir.

Ses tanımanın çeşitli zorlukları vardır. Birincisi, az sayıda dil için veri kümeleri mevcuttur. İkincisi, ses girdileri arasında cinsiyet, aksan, yaş ve yerel dil değişimlerden dolayı büyük değişkenlikler oluşmaktadır. Son olarak, arka plan gürültüsü, kayıt cihaz kalitesi ve çevresel faktörler ses tanıma performansını etkileyen faktörlerdir.

Diğer yandan, derin öğrenme, lineer veya lineer olmayan dönüşümlerden oluşan çok sayıda işlem katmanı içeren ağlar kullanarak veri içerisindeki yüksek seviyeli soyutlamaları modellemek için çeşitli algoritmalar kullanan makine öğrenmesinin bir alt alanıdır. Ayrıca, derin öğrenme hedeflenen çıktıyla en yakın ilişkide olan özelliklerin verimli kullanımı ve seçimini de sağlamaktadır. Derin öğrenme uygulamalarının literatürdeki en iyi yöntemlere göre daha iyi performans göstermesiyle birlikte derin öğrenme araştırmacılar arasında önem kazanmıştır. Derin öğrenme algoritmalarının son yıllardaki başarısı ve popülaritesi ile derin öğrenme görüntü işleme, doğal dil işleme, zaman serisi analizi ve ses tanıma gibi çeşitli uygulama alanlarında kullanılmaya başlamıştır.

Derin öğrenme mimarileri, ses iyileştirme, konuşmacı tanıma ve ses aktivite tespiti gibi ses tanımanın çeşitli uygulama alanlarında kullanılmaya başlanmıştır. En popüler derin öğrenme mimarileri arasında Derin Sinir Ağları (DSA); Evrişimli Sinir Ağları (ESA) ve Özyinelemeli Sinir Ağları (ÖSA) ve ÖSA'nın özelleşmiş bir versiyonu olan Uzun Kısa Dönem Hafıza (UKDH) ÖSA ses tanıma için yaygın olarak tercih edilmektedir. DSA genellikle Saklı Markov Model (SMM) tabanlı ses

tanına sistemleri ile hibrit bir şekilde kullanılmaktadır. ESA genellikle görüntü işlemede kullanılmaktadır; ancak, aynı zamanda ses tanımada da kullanılabilmektedir. Ses tanımada ESA genellikle *spektrogram* olarak adlandırılan görsel ses girdileri için tercih edilmektedir. ÖSA ve UKDH sıralı girdilerin analizinde başarılı mimarilerdir ve bu nedenle ses sinyallerinin zaman serisi girdi özellikleri üzerinde kullanılmaktadırlar. Bu mimarilerden ayrı olarak ses tanıma görevleri için Derin İnanç Ağları ve Derin Otokodlayıcılar gibi diğer derin öğrenme mimarileri de kullanılmaktadır.

ÖSA ve UKDH, sıralı ve zaman serisi girdilerini işlemedeki yeteneklerinden dolayı ses tanıma alanında en yüksek başarımı ve tatmin edici performansı sağlamak için tercih edilen en yaygın olarak kullanılan derin öğrenme mimarileridir. Özellikle UKDH özelleşmiş kapı-tabanlı mimarisiyle ses tanıma görevlerinde yüksek performans sağlamaktadır. Ayrıca bu mimariler için öğrenme algoritmaları gradyan iniş optimizasyonunun varyantları olarak özelleşmiştir.

Gradyan iniş optimizasyonu, derin öğrenme modellerinin eğitimi ve optimizasyonu için en popüler yaklaşımlardan biridir. Gradyan iniş optimizasyonu ağ çıktı hatalarını geriye doğru yayan ve parametreleri buna göre güncelleyen geriye yayılım algoritmasına dayanmaktadır. Gradyan iniş optimizasyonun yığın gradyan iniş, stokastik gradyan iniş ve mini-yığın gradyan iniş olmak üzere üç varyantı bulunmaktadır. Bu varyantlar derin öğrenme modellerinin parametre güncellemeden önce kaç adet örnek kullanılacağı konusunda ayrılmaktadır. Derin öğrenme tabanlı ses tanıma çalışmalarında mini-yığın gradyan iniş diğer alternatiflerine göre avantajlarından dolayı tercih edilmektedir.

Derin öğrenme tabanlı ses tanıma sistemleri ses girdilerinin girdi özelliklerine bağlı olarak tanınmasında başarılı sistemlerdir. Derin öğrenme tabanlı ses tanıma sistemlerinin performansı, cinsiyet, aksan, konuşmacı ve dilbilimsel içerik gibi farklı karakteristikteki kullanıcı türlerini içeren kapsamlı ses veri kümeleri başta olmak üzere çeşitli faktörlere bağlıdır. Ses tanıma sistemlerinin ses tanıma performansını artırabilmek için geniş kapsamlı bir ses veri kümesi

zorunludur. Derin öğrenme tabanlı ses tanıma sistemlerinin performanslarını artırabilmek için gerekli olan önemli bir faktör, bu sistemleri geniş aksan, cinsiyet ve konuşmacıya sahip olan çok fazla sayıda veriyle eğitmektir.

Bu tez çalışmasında, derin öğrenme sistemlerinin eğitimi için önemli bir yeri olan mini-yığın örnek seçimi için yeni yöntemler önererek derin öğrenme tabanlı ses tanıma sistemlerinin performansını artırmak amaçlanmaktadır. Derin öğrenme tabanlı ses tanıma sistemlerinin model performansını artırabilmek için mini-yığın örneklerini seçmek için çeşitli stratejiler önerdik. Özellikle, cinsiyet tabanlı, aksan tabanlı ve cinsiyet ve aksan tabanlı stratejileri birlikte kullanan hibrit stratejiler olmak üzere üç tür mini-yığın örnek seçim stratejisi önerilmiştir. Her bir tür mini-yığın örnek seçim stratejisi için homojen ve heterojen olmak üzere iki ayrı kategori araştırılmıştır. Homojen stratejilerde bir mini-yığında sadece bir meta özellik değerlendirilmiştir. Örneğin, sadece bayan veya sadece İngiliz aksanlı kayıtlar bir mini-yığına alınmıştır. Heterojen stratejilerde bir mini-yığında tüm özellikler orantısal olarak bulunmaktadır. Örneğin, mini-yığınlar yarı bayan yarı erkek kayıtlardan oluşmaktadır.

Seçilen cinsiyet ve aksan meta özellikleri ses tanıma sistemlerinin performansı için önemli faktörlerdir. Ses tanıma modellerinde cinsiyet ve aksan temsili kullanılan veri setinin karakteristiğine bağlı olarak iyi veya zayıf performansa sebep olmaktadır. Bu nedenle, bu tez çalışmasında mini-yığın örneklerinin seçiminde cinsiyet ve aksan meta özellikleri kullanılmıştır. Cinsiyet özelliği kadın ve erkek konuşmacılar arasında dengeli bir şekilde dağılmadığı zaman ses tanıma sistemleri iyi performans gösteremezler. Benzer şekilde, bir dile ait farklı aksanlar ses tanıma sistemi içerisinde bulunmuyor ise test örneklerindeki değişik aksanlar düzgün bir şekilde tanınamayacaktır. Bu bilgileri takip ederek, derin öğrenme tabanlı ses tanıma sistemlerinin performansını artırabilmek için cinsiyet ve aksan tabanlı mini-yığın örnek seçim stratejileri önerilmiştir.

Önerilen üç tür mini-yığın örnek seçim stratejileri derin öğrenme tabanlı ses tanıma sistemlerinin performansını mini-yığın örnek özelliklerine göre

değerlendirmek ve cinsiyet ve aksan bilgilerinin derin öğrenme sistemlerinin eğitimindeki etkisinin gözlemlenmesi için geliştirilmiştir. Derin öğrenme mimarisi olarak ses tanıma sistemlerinde oldukça popüler olarak kullanılan UKDH (LSTM) tercih edilmiştir. Önerilen tüm stratejiler, birbirleriyle adil bir şekilde karşılaştırabilmek ve en iyi stratejiyi belirleyebilmek için aynı derin öğrenme sistemi üzerinde değerlendirilmiştir.

Deneysel sonuçlar, homojen cinsiyet – heterojen aksan hibrit mini-yığın örnek seçim stratejisi, homojen cinsiyet ve heterojen aksan stratejileri eğitim ve test ölçülerine göre diğer alternatif stratejilerden daha başarılı olduğunu göstermiştir. Bu sonucun elde edilmesinin sebebi, bir mini-yığın için homojen cinsiyet seçilmesi derin öğrenme sistemlerinin ses girdilerini ile çıktı metinler arasındaki ilişkiyi daha iyi modelleyebilmesini sağlaması olduğu düşünülmektedir. Homojen cinsiyet stratejisi bir mini-yığındaki örnekleri sadece bir cinsiyetle seçmektedir. Bu da birbirine yakın cinsiyet özelliklerinin bir mini-yığında bulunması ve derin öğrenme tabanlı ses tanıma sistemlerinin karakter söylenişlerini daha iyi öğrenmesini sağlamaktadır. Ayrıca, bir mini-yığında tüm aksanların temsil edildiği heterojen aksan stratejisi de iyi performans göstermektedir. Her mini-yığında tüm aksanlar bulunduğu zaman derin öğrenme tabanlı ses tanıma sistemi daha başarılı olmaktadır, çünkü tüm mini-yığınlarda olası tüm aksanları ve bu aksanlardaki karakterlerin söylenişlerini öğrenmektedir. Homojen cinsiyet – heterojen aksan hibrit stratejisi önerilen tüm mini-yığın örnek seçim stratejileri arasında en iyi performansı göstermiştir. Bu sonucun sebebi ise, hem homojen cinsiyet hem de heterojen aksan stratejileri iyi performans göstermişlerdir ve bu iki stratejinin hibrit olarak kullanılması da daha iyi ses tanıma performansı sağlamıştır.

Tüm sonuçlar değerlendirildiği zaman, cinsiyet ve aksan meta bilgileri mini-yığın örnek seçimi için kullanıldığı zaman derin öğrenme tabanlı ses tanıma modellerinin daha iyi performans gösterdikleri sonucuna ulaşılabilir. Özellikle, başarılı mini-yığın örnek seçim stratejileri hibrit olarak kullanıldığı zaman bu sistemlerin performansları daha da artırılabilmektedir.

ACKNOWLEDGEMENT

I would like to present my gratitude to my supervisors Assoc. Prof. Dr. Zekeriya TÜFEKCİ for his endless support, patience, guidance, and motivation for the successful completion of this thesis.

I wish to express my special thanks to Assoc. Prof. Dr. Sami ARICA and Assoc. Prof. Dr. Umut ORHAN for their suggestions, corrections and their valuable help. I would like to thank members of the PhD thesis jury Assoc. Prof. Dr. Gülay TEZEL and Assoc. Prof. Dr. Serdar YILDIRIM for their constructive suggestions and corrections.

Finally, I would like to thank my husband Ahmet Şakir DOKUZ, my son Ömer Avni DOKUZ, my daughter Ebrar Bilge DOKUZ, my mother Filiz TORUN, my father Hayati TORUN and other family members for their limitless support and patience.

CONTENTS **PAGE**

LIST OF TABLES **PAGE**

LIST OF FIGURES **PAGE**

LIST OF ABBREVIATIONS

GMM	: Gaussian Mixture Model
HMM	: Hidden Markov Model
DL	: Deep Learning
DNN	: Deep Neural Networks
CNN	: Convolutional Neural Networks
DBN	: Deep Belief Networks
DAE	: Deep Autoencoder
RNN	: Recurrent Neural Networks
LSTM	: Long Short-Term Memory
RBM	: Restricted Bolzmann Machine
ReLU	: Rectified Linear Units
CTC	: Connectionist Temporal Classification
LER	: Label Error Rate

1. INTRODUCTION

Speech recognition is the process of converting speech signals into text transcription. With the increase of computational power and availability of communication with computers by voice, speech recognition systems gained more attention. Text-to-speech and speech-to-text systems are widely used in many technologies, such as mobile communication, search engines, and personal digital assistance systems.

However, speech recognition has several challenges. First of all, currently the speech corpuses for few languages are available for speech recognition systems. There is a lack of datasets and systems to include many languages. Second, speech recognition systems should handle variations in the datasets, such as accent, gender, age and other variations of speakers. Third, speech signals have many distorting factors, such as background noise, microphone quality, and environmental factors. To overcome some of these challenges, deep learning architectures are getting more attention in speech recognition researches.

Deep learning is a branch of machine learning that uses a set of algorithms that attempt to model high-level abstractions in data by using a deep graph with multiple processing layers, composed of multiple linear and non-linear transformations (Goodfellow et al., 2016; Deng et al., 2014). Deep learning provides automatic selection and ranking of features in the datasets using efficient algorithms. Recently, deep learning achieved great attention and tremendous success in application areas, such as image processing, natural language processing, sequence alignment, and speech recognition.

Deep learning systems use datasets and training algorithms for developing their models. One of the most popular training algorithms in deep learning systems is gradient descent optimization method (Goodfellow et al., 2016; Ruder, 2016). Gradient descent optimization method takes into account first-order derivative when performing updates of the parameters. The algorithm updates model

parameters by the opposite direction of the gradient of the objective function and goes until it finds a local minimum. There are three variants of gradient descent algorithm that differ in using the amount of the data, batch gradient descent, stochastic gradient descent, and mini-batch gradient descent (Ruder, 2016). Batch gradient descent computes the gradients of the cost function and updates the parameters for all of the dataset, while stochastic gradient descent computes the gradients and updates parameters for each data instance of the dataset separately. Each of these variants has several weaknesses and for this purpose mini-batch gradient descent is proposed (Goodfellow et al., 2016).

In mini-batch gradient descent, gradient computation and parameter update are performed after a batch of *n* data instances are traversed (Ruder, 2016). With this strategy, fluctuations in stochastic gradient descent and slow and redundant computations in batch gradient descent could be prevented. Standard mini-batch gradient descent selects data instances for each mini-batch randomly from non-selected data of the corpus. However, when the data instances are randomly selected, the deep learning model and parameters are updated based on the data instances that are in current mini-batch. Especially in speech recognition, there are many factors that affect performance of deep learning models, such as accent, gender, age, and speaker. The performance of the model could be increased by selecting appropriate instances for each mini-batch based on these factors.

Deep learning has several deep architectures, such as Deep Neural Networks (DNNs), Convolutional Neural Networks (CNNs), and Recurrent Neural Networks (RNNs) which are widely utilized to be used in speech recognition systems (Yu and Deng, 2016). DNNs and CNNs are feed-forward architectures that contain multiple layers of transformations and nonlinearity with the output of each layer that are feeding subsequent layer. RNNs is a recurrent architecture that has both forward pass which transfers information to subsequent layers and recurrent pass that processes past information and current input together.

With the popularity and high performance of deep learning systems with respect to other candidates, these systems are also being utilized in speech recognition problems. Among the most popular deep learning architectures, Deep Neural Networks (DNN), Convolutional Neural Networks(CNN), and especially Recurrent Neural Networks(RNN) and Long Short Term Memory (LSTM) RNNs, which is a special kind of RNN, are preferred for speech recognition. CNN is mostly utilized for visual speech inputs, which is called as *spectrograms*, while RNN and LSTM are utilized for time series feature inputs of speech signals.

In the literature, there are many studies present that use deep learning architectures for speech recognition. In the studies that use DNN for speech recognition, Dahl et al. (2011) proposed a context-dependent pre-trained DNN approach for acoustic model of speech recognition and compared with Gaussian Mixture Model (GMM) and DNN-HMM approach outperformed GMM-HMM approach. Yu et al. (2013) proposed Kullback–Leibler divergence (KLD) regularization adaptation technique for context-dependent DNN-HMM for better speech recognition performance. Seltzer et al. (2013) investigated the performance of DNN-based acoustic models and proposed three methods to improve accuracy for noise robust speech recognition. Jaitly et al. (2012) proposed to use deep belief networks (DBN) for pre-training DNNs for DNN-HMM hybrid speech recognition systems and reported that proposed approach outperformed GMM-HMM baseline. Dahl et al. (2013) investigated the behaviour of DNNs using rectified linear units (ReLU) and dropout and reported that using ReLU and dropout improves performance of speech recognition systems. Han et al. (2014) proposed to use DNNs to extract high level features for speech emotion recognition. Lalitha et al. (2019) proposed a DNN based perceptual speech feature extraction approach for emotion recognition. Lei et al. (2014) proposed a framework for speaker recognition using i-vector models and DNNs. Snyder et al. (2016) proposed an end-to-end speaker verification system that consists of DNNs that take variable length speech segments and maps into a speaker embedding, and they reported that

proposed system outperformed i-vector based baseline in equal error-rate (EER) by 13%, average. Variani et al. (2014) investigated the use of DNNs for a small footprint text-dependent speaker verification. Zen et al. (2013) proposed a speech synthesis scheme that is based on DNNs and they reported that DNN based speech synthesis system outperform HMM based baseline.

In the studies that use CNN for speech recognition, Sainath et al. (2013) explored configurations of CNN, such as convolution layer count, optimal number of hidden units, best pooling strategy, and best input feature type, for obtaining better performance than DNN for large vocabulary continuous speech recognition. Sercu et al. (2016) proposed a very deep CNN approach that consists of 14 weight layers and applied multilingual speech recognition task on the generated deep architectures. Abdel-Hamid et al. (2014) proposed a CNN based speech recognition system to reduce error rate using limited-weight-sharing scheme to better model speech features. Qian et al. (2016) proposed a very deep CNN model for noise robust speech recognition and investigated best configurations for proposed deep CNN model for noise robust speech recognition. Zhang et al. (2017) proposed an end-to-end speech recognition system that is based on CNNs with Connectionist Temporal Classification (CTC) approach without using a recurrent layer for obtaining computationally efficient model and competitive results. Palaz et al. (2015) investigated the use of CNNs to large vocabulary speech recognition which takes raw speech signals as inputs and the proposed approach outperformed classical DNN based speech recognition system. Mao et al. (2014) utilized CNN for learning affect-salient features for speech emotion recognition using two-phase learning. Badshah et al. (2017) used CNN for extracting discriminative features for speech emotion recognition using spectrograms of input speech signals. Thomas et al. (2014) utilized CNNs as acoustic models for speech activity detection (SAD) in mismatched acoustic conditions using noisy radio communication channels data. Swietojanski et al. (2014) used CNNs for large vocabulary distant speech recognition using the data from single distant microphone and multiple distant

microphones and CNN outperformed DNN and GMM in terms of Word Error Rate (WER). Fu et al. (2016) proposed two signal-to-noise-ratio (SNR) aware algorithms for modelling CNN for speech enhancement and the proposed model outperformed DNN for denoising performance. Torfi et al. (2018) proposed a 3D CNN model for adaptive feature learning for text independent speaker verification.

In the studies that use RNN/LSTM for speech recognition, Graves et al. (2013) investigated the use of deep LSTM RNNs for speech recognition to achieve state-of-the-art results and reported that their deep LSTM model achieved best phoneme error rate. Graves et al. (2013b) investigated the use of deep bidirectional LSTM architecture as an acoustic model to NN-HMM hybrid speech recognition system and achieved equal performance with previous studies. Graves and Jaitly (2014) proposed an end-to-end speech recognition system that do not require phonetic representation using a combination of LSTM and CTC objective function. Sak et al. (2015) proposed techniques that improve performance of LSTM RNNs as acoustic models for LVSR and resulted that stacking frames and reducing frame rate provides more accurate models and faster decoding. Li and Wu (2015) proposed a deep LSTM to obtain performance improvement and applied on large vocabulary telephone speech recognition task and resulted that deep LSTM strategy provide better performance. Miao et al. (2015) proposed an end-to-end speech recognition system that uses weighted finite-state transducers (WFSTs) and bidirectional LSTM deep architecture. Sak et al. (2014) proposed a distributed training for LSTM using stochastic gradient descent on a cluster of machines and reported that their proposed system outperformed DNN. Lu et al. (2016) proposed an efficient learning rate schedule method that improves the accuracy of large vocabulary speech recognition. Mirsamadi et al. (2017) investigated using RNNs for automatically extracting emotion-related features for speech emotion recognition by using both short-time frame-level emotional features and temporal aggregation of such features. Maas et al. (2012) investigated the use of deep recurrent auto encoder neural network for noise reduction in automatic speech

recognition. Weninger et al. (2015) proposed an LSTM RNN framework that are trained by an optimal speech reconstruction objective for speech enhancement in noise robust speech recognition. Weninger et al. (2014) investigated the use of LSTM RNNs on training, network architecture and representation of features for regression based single-channel speech separation. Hughes and Mierle (2013) proposed a multi-layer RNN model for voice activity detection that outperforms larger baseline GMM with a hand-tuned state machine (SM) system. Sun et al. (2015) investigated the use of Deep Bidirectional LSTM RNN (DBLSTM RNN) for voice conversion which is able to model temporal correlations between speech frames. Zen and Sak (2015) proposed a unidirectional LSTM with recurrent output layers for low-latency speech synthesis system.

Although one deep learning architecture is sufficient for gaining good performance for speech recognition systems and applications, some studies consider using a hybrid of two or more deep learning architectures to achieve better performance. Trigeorgis et al. (2016) proposed a framework that combines CNNs and LSTMs to automatically learn best feature representation from raw speech signals for speech emotion recognition. Lim et al. (2016) investigated the use of concatenated architecture from CNNs and RNNs for extracting better features than hand-crafted features for speech emotion recognition. Zhao et al. (2018) proposed an end-to-end CNNs and RNNs based model for catching local variations in both time and frequency domains for speech enhancement. Hori et al. (2017) proposed an end-to-end model using CNNs as encoder and LSTMs as language model for speech recognition which reduced the error rate. Chan et al. (2015) utilized RNNs and DNNs for increasing speech recognition performance on embedded devices by building a large RNNs acoustic model and pass this model to DNNs for speech recognition. Chen et al. (2018) proposed a 3D attention-based CRNN deep learning architecture which takes MFCC with deltas and delta-deltas as input for speech emotion recognition. Wu et al. (2016) proposed a deep model in which CNNs and DNNs extract visual cues and acoustic features, and BiLSTMs model higher level

dependencies among features and visual information. Sainath et al. (2015) combined CNNs, LSTMs, and DNNs into a unified deep learning system, which is named as CLDNN, for taking advantage of each architecture and achieved better performance than LSTM which is considered as strongest architecture of these three alternatives in speech recognition. Wang et al. (2019) proposed CNN-BLSTM-CTC deep learning hybrid model for Mandarin speech recognition. They employed CNN for learning of local speech features, BLSTM for learning past and future dependencies, and CTC for decoding purposes and claim that their proposed method outperformed best existing model.

These literature studies show that there are many studies present which use deep learning architectures for speech recognition. Especially RNN and LSTM are highly preferred among the most popular deep learning architectures with the powerful and successful modelling capability of sequential inputs. Also, hybrid architectures which combine more than one deep learning architectures are getting more attention in deep learning based speech recognition studies that have the potential to improve performance of speech recognition systems.

In addition to the literature review of deep learning architectures for speech recognition, several studies are performed to increase performance of mini-batch gradient descent optimization algorithm. Zhang et al. (2017, 2018) proposed a determinantal point processes and a repulsive point processes based sample selection scheme for mini-batch stochastic gradient descent algorithm to obtain presence of diverse points in a batch and get improved convergence speed instead of randomly selecting samples. Loshchilov and Hutter (2015) proposed an online batch selection strategy that takes into account loss values of each data instance and the selection probability of these data instances for training is decreased with respect to the loss values. Doetsch et al. (2017) proposed a batch construction strategy that produces large variability of sequences and reduces irrelevant computations at the same time. Peng et al. (2019) proposed typicality sampling scheme for mini-batch gradient descend to select high representative samples

instead of standard random selection. Park et al. (2018) proposed paired mini-batch sample selection system for image forensics. Their proposed mini-batch system takes images as pairs, in comparison with standard random strategy.

The studies in the literature have several limitations. First of all, the studies do not take into account the variability of speech datasets and do not propose solutions which collapse various speech recognition datasets. Second, many of these studies focus on different areas and do not focus on speech recognition problems. Third, these studies do not use meta information of speech datasets to propose solutions to mini-batch sample selection.

1.1. Contribution of the Thesis

In this study, we propose strategies for selecting mini-batch samples for speech recognition tasks to increase model performance of deep learning based speech recognition. These strategies are constructed to better represent each meta feature in the mini-batches. For this purpose, gender adjusted strategies, accent adjusted strategies and hybrid strategies which combine gender and accent adjusted strategies are proposed. In particular, homogeneous and heterogeneous strategies are proposed for gender and accent information of the speech corpuses. In homogeneous strategies, only one meta value, i.e. only female speaker or only English accent, is used for one mini-batch. In heterogeneous strategies, proportional meta values, i.e. half female and half male speakers, are used for one mini-batch. The experiments showed that our proposed strategies for representing each meta property in the mini-batch provide better performance in comparison with randomly selected mini-batch samples.

1.2. Organization of the Thesis

The rest of this thesis is organized as follows. In Section 2, basic concepts of automatic speech recognition are introduced, feature extraction, acoustic and language models, and hypothesis search are explained. Deep learning, Recurrent

Neural Networks, and Long-Short Term Memory RNNs are explained, and gradient descent optimization algorithm and mini-batch gradient descent are investigated.

In Section 3, proposed mini-batch sample selection strategies are presented. First, standard mini-batch sample selection strategy is explained which selects mini-batch samples randomly, and then gender and accent adjusted strategies are proposed. Finally, gender-accent hybrid mini-batch sample selection strategies are presented.

In Section 4, the dataset, utilized deep learning system, and experimental results of proposed mini-batch sample selection strategies are presented. Gender, accent and hybrid strategies are evaluated and the best strategy among the proposed strategies is investigated. Also, the results of proposed mini-batch sample selection strategies are compared with the literature studies and a discussion about the results of proposed methods are presented.

In Section 5, conclusions of the thesis and possible future studies are presented.

2. BASIC CONCEPTS

In this section, first speech recognition problem and the steps of speech recognition are presented. Then, deep learning and Recurrent Neural Networks (RNN), and Long-Short Term Memory (LSTM) RNN are introduced. Finally, gradient descent optimization algorithm and mini-batch gradient descent are explained.

2.1. Speech Recognition

Speech recognition is the task of producing a text transcription of the audio signal from a speaker (Yu and Deng, 2014). Speech recognition gained attention recently, with the help of computation power of computing devices, presence of more data, and increasing success rates of speech recognition systems. Speech recognition has several possible applications, such as voice search, personal digital assistance, smart home environments, and mobile communications.

Formally, speech recognition problem can be explained as given in Equation (2.1) (Yu and Deng, 2014). Given a sequence of t vectors of acoustic information $X = x_1 \dots x_t$ that we assume encodes a sequence of T words $w = w_1 \dots w_T$. The aim of speech recognition is to find the best transcription hypothesis w according to some learned scoring function $P(w, X)$ (Yu and Deng, 2014):

$$\hat{w} = \underset{w}{argmax} P(w, X) \tag{2.1}$$

Because the Equation (2.1) is difficult to model directly, Bayes' rule is used to generate an equivalent equation which is more easily modelled, as shown in Equation (2.2) (Yu and Deng, 2014).

$$\hat{w} = \underset{w}{argmax} \{P(X|w)P(w)\} \tag{2.2}$$

In Equation (2.2), $P(X|w)$ is the acoustic model, and $P(w)$ is the language model (Yu and Deng, 2014).

A typical speech recognition system consists of four modules, namely, signal processing and feature extraction, acoustic model, language model, and hypothesis search (Yu and Deng, 2014). Signal processing and feature extraction module takes audio signal as input and performs several processing steps, such as noise removal, signal to feature domain conversion, and feature extraction from the audio signal. The acoustic model takes features as input and phonetic knowledge and generates an acoustic model score for the variable-length feature sequence. The language model estimates the probability of a hypothesized word sequence by using the correlation between words in a training corpus. The performance of language model could be improved with providing domain knowledge to the model. The hypothesis search component combines acoustic model and language model scores and the hypothesized word sequence, and outputs the word sequence with the highest score as the recognition result. Basic speech recognition flow is presented in Figure 2.1 with respect to these four modules.

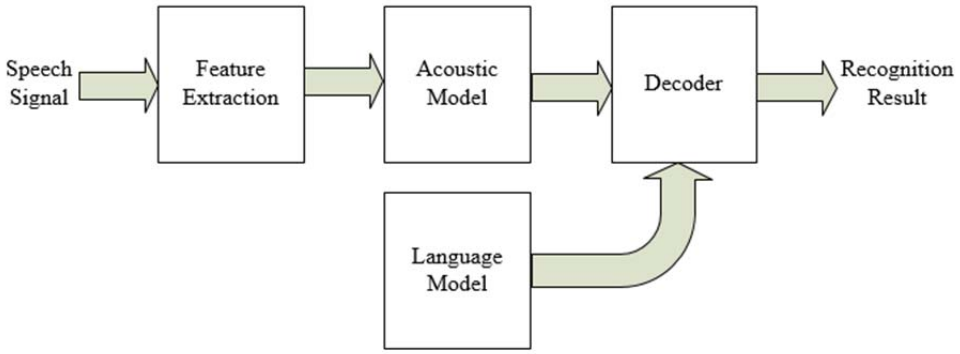

Figure 2.1. Basic flowchart of speech recognition

Each modules of speech recognition are explained below in detail.

2.1.1. Feature Extraction

In speech recognition, several feature extraction techniques are used for extracting features from speech signals for recognition purposes. The main feature extraction techniques are Linear Predictive Analysis (LPC), Perceptual Linear Predictive (PLP), Relative Spectra Filtering of Log Domain Coefficients (RASTA) and Mel-Frequency Cepstral Coefficients (MFCC) (Salomons and Havinga, 2015; Benba et al., 2015; Narang and Gupta, 2015).

In this thesis study, MFCC is selected as a feature extraction technique due to its robust and dynamic nature for speech feature extraction. MFCC are coefficients of Mel-Frequency Cepstrum (MFC) which is a short-term power spectrum of a sound, based on a linear cosine transform of a log power spectrum on a nonlinear mel scale of frequency. Figure 2.2 presents the main steps of MFCC feature extraction (Salomons and Havinga, 2015; Benba et al., 2015).

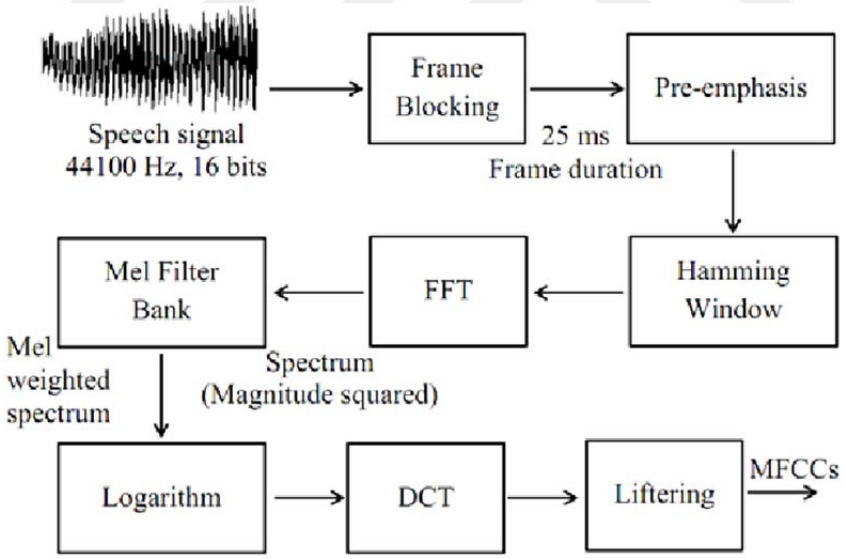

Figure 2.2. Block diagram of MFCC feature extraction

As can be seen in Figure 2.2, first of all, the speech signal is divided into small subsets to obtain stationary signals from long non-stationary speech signals

which is called as frame blocking. For this purpose, 10 ms to 30 ms frame sizes are used to obtain stationary signal input. In frame blocking, the speech input is divided into frames which have same length, i.e. 20 ms. A step size is used for dividing input speech into frames which is smaller than frame size and a proportion of the frames are desired to be overlapped to obtain more adequate and representative features.

At the second step, pre-emphasis is performed on each frame to increase energy of speech waveforms. In pre-emphasis, first order derivative is applied on speech signal to reduce effect of distorting factors in speech signals. A coefficient $0 \leq k < 1$ is used, which is generally between 0.9 and 1, to remove noises and to amplitude speech signals.

At the third step, a windowing process is applied on the speech signal to reduce the discontinuity of speech signals and combine starting and ending points of the signal frames. Several windowing techniques are present for speech signals, including rectangular windowing and Hamming windowing. Rectangular windowing gives equal weight to all samples of the signal, while Hamming windowing gives most weight to middle parts of the frames and reduces weights sharply for starting and ending points.

At the fourth step, Fourier Transform (FT) is applied on windowed speech signal to convert each frame of the signal from temporal domain to frequency domain. Also, Fourier Transform provides magnitude of each frequency in the speech signal. In general, Fast Fourier Transform (FFT) which is a fast implementation of Discrete Fourier Transform (DFT) is preferred for speech signal processing.

At the fifth step, filterbank analysis is performed. Because human hearing perception is not in a linear form, different pitches are measured on the Mel-scale for each frequency. Figure 2.3 presents a general form of a Mel-scale filterbank. As can be seen in the figure, the Mel-scales are in triangular form and first Mel-frequencies are linear while others increase logarithmically.

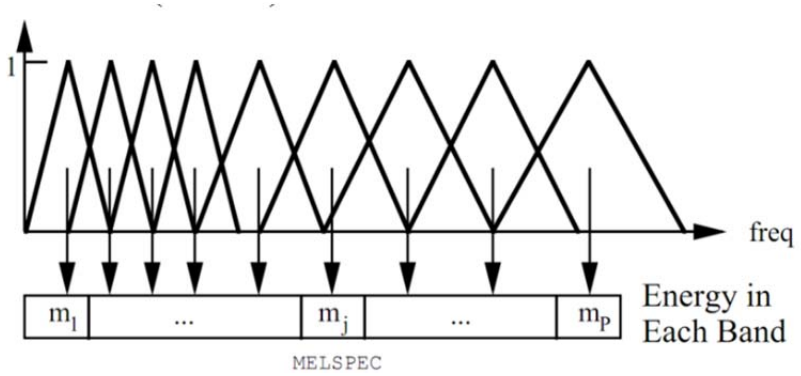

Figure 2.3. Mel-scale filterbank

At the sixth step, Discrete Cosine Transform (DCT) is performed on log filterbank amplitudes to obtain Mel-scale cepstral coefficients. An extra log operation could be performed with DCT for extracting finer MFCC features and for obtaining better performance. After DCT is applied, delta and delta-delta operations could be performed for extraction of more MFCC features.

At the seventh step, liftering is performed to rescale the MFCC features to have increased magnitudes using cepstral sine lifter parameter.

2.1.2. Acoustic Model

Acoustic model is one of the essential parts of speech recognition which is used for matching input speech features with their corresponding phonetic hypothesis words (Yu and Li, 2017; Aggarwal and Dave, 2011; Gavat and Militaru, 2015). In acoustic modeling, sub-word units are preferred to better project input speech features with the hypothesis words, which are called phonemes. To produce a successful acoustic model, a comprehensive training procedure should be performed with enough input data. The training procedure for acoustic model is important because a speech recognition system could only model what it has learned during training. In acoustic model training procedure, a common pattern is generated from the patterns of speech sounds which have the same class label. The generated common patterns of each phoneme are used for classification of

unknown test speech sounds with a class label by computing a similarity between common patterns with test pattern.

For acoustic models generation, two candidate approaches are present, i.e. Dynamic Time Warping (DTW) (Permanasari et al., 2019), and Hidden Markov Model (HMM) (Gales and Young, 2008). In DTW, the distance between training acoustic information and test acoustic information are calculated and the best match (smallest distance) between acoustic information of training and test units are calculated. HMMs are treated as state-of-the-art acoustic models for speech recognition for many years due to their success at modeling input features with class labels.

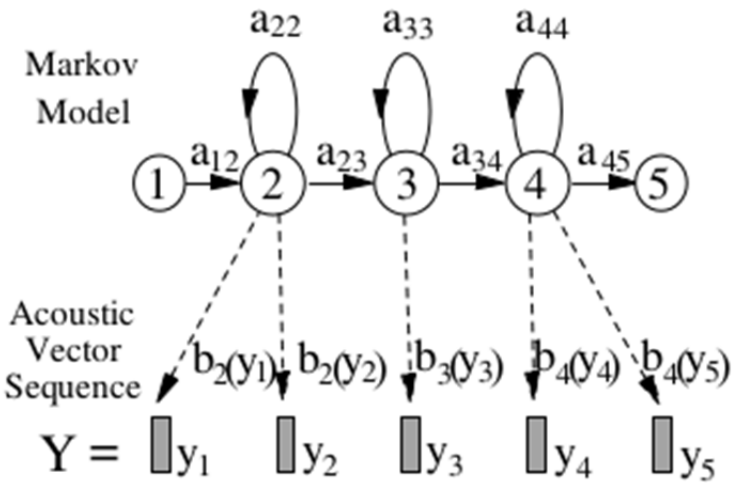

Figure 2.4. HMM-based phone model

In a basic HMM acoustic model, each spoken unit is conveyed on each other to generate an output sequence as presented in Figure 2.4. In the figure, a_{ij} are transition probabilities, and b_j are output observation distributions. An HMM makes a transition from its current state to one of its connected states for each time step. The probability of making a transition between states are given by transition

probability a_{ij}. The transition and output probabilities of HMM are trained by using a large dataset and with forward-backward, or Baum-Welch algorithms. Also Gaussian Mixture Model (GMM) is widely used for modeling state-output distribution.

With the advances in deep learning technologies, new acoustic modeling techniques are proposed which are based on deep learning methods. Several deep architectures are proposed as acoustic models to reduce error rate of speech recognition systems with respect to state-of-the-art GMM-HMM acoustic models. The main deep learning architectures that are used as acoustic model generation are Restricted Bolzmann Machine (RBM), Convolutional Neural Network (CNN), Deep Autoencoder (DAE), and Deep Belief Network (DBN).

2.1.3. Language Model

In general, *n*-gram language models are used for speech recognition systems (Deng and Yu, 2014; Zhu et al., 2016). In other words, to correctly predict likelihood of *n*th word, previous n-1 words are used. In language model, the probability of a word sequence W is calculated using the probabilities of each word in a sequence which has m words as given in Equation (2.3).

$$P(W) = P(w_1, w_2, w_3, \ldots, w_m) \tag{2.3}$$

When the probability of each word are separated, Equation (2.3) is transformed to Equation (2.4).

$$P(W) = P(w_1)P(w_2|w_1)P(w_3|w_1w_2) \ldots P(w_m|w_{m-n+1} \ldots w_{m-1}) \tag{2.4}$$

A common value of n is generally selected as 3 (n=3) which is also known as tri-grams. In tri-gram, $P(w_3|w_1w_2)$ is modeled with using words of w_1, w_2, and w_3.

Tri-gram language models are widely preferred because many words in sentences are strongly associated with their previous one and two words. Another reason of preference of tri-gram language models is that generating a language model with three word sequences are reasonable in terms of search space and computational infrastructures. Also, a moderate sized dataset could provide an efficient language model for tri-grams.

With the advances of deep learning technologies, in line with the acoustic model, deep learning based language models are proposed which provides better performance in comparison with *n*-gram language models. First of all, deep learning based language models do not treat words as exact sequences, but they rather see them as a projection of words onto the *n*th word. This provides less parameters which should be trained. But on the other hand, deep learning based language models have a larger computational complexity with respect to *n*-gram language models. For deep learning based language models, Recurrent Neural Networks (RNN) is the uttermost preferred deep learning architecture which has the dependency modeling capability among the sequential datasets.

2.1.4. Decoder

The decoding process in speech recognition is the final step which takes into account acoustic model and language model and produces a recognition result of a given input speech sound (Aubert, 2002; Watanabe et al., 2017). In this process, Equation (2.1) is used for decoding an input speech. A typical decoder in a large vocabulary speech recognition system should satisfy four main points, i) it should generate hypothetical word sentences, ii) it should score the generated hypotheses using knowledge sources, iii) it should discard redundant sequences, and iv) it should go backwards to retrieve the best possible outcome.

Several state-of-the-art algorithms are used for decoding, i.e. Viterbi decoding, Beam Search decoding, and Weighted Finite State Transducers (WFST)

decoding algorithms. All of these algorithms are proposed to reduce computational complexity of search space and to better provide a recognition result.

Recently, new decoding methods are proposed that are based on deep learning techniques. Connectionist Temporal Classification (CTC) and Attention-based methods are generally preferred for decoding deep learning-based speech recognition systems.

2.2. Deep Learning

Deep learning is a sub-section of machine learning that uses a set of algorithms that attempt to model high-level abstractions in data by using a deep graph with multiple processing layers, composed of multiple linear and non-linear transformations (Goodfellow et al., 2016; Deng et al., 2014). Deep learning provides automatic selection and ranking of features in the datasets with efficient algorithms. Deep learning architectures have tremendous success in many areas, such as speech recognition, image processing, natural language processing, and sequence prediction.

Figure 2.5 presents the relationship of deep learning with artificial intelligence and machine learning. As can be seen in the figure, deep learning is under machine learning category, which is a sub-field of artificial intelligence. Artificial intelligence is the general term of the methods that provide intelligence to the machines. Machine learning is the term of a set of algorithms and statistical models which accomplishes the tasks in a sustainable manner without explicitly introduced. In that sense, deep learning is one of the machine learning methods which uses big datasets and employs neural networks to gain insights and to perform many non-trivial tasks without being controlled.

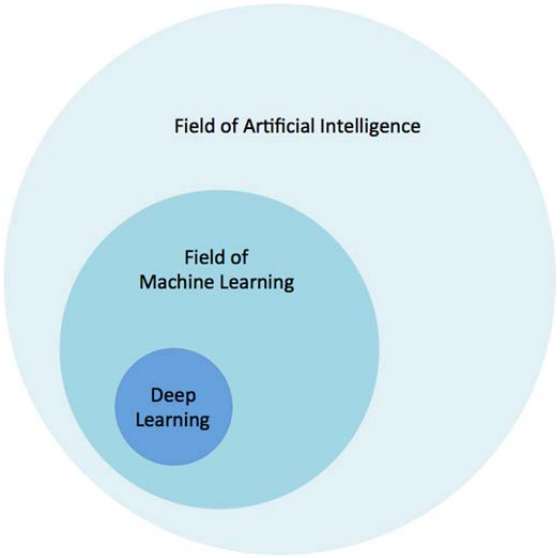

Figure 2.5. The relationship of deep learning with artificial intelligence and machine learning

Most recent deep learning architectures consider the structure of state-of-the-art artificial neural networks to perform deep learning tasks. The difference of deep learning is at the learning algorithms and with the level-wise arrangement and simplification of problems. At each level, the features become more abstract and far from the original formation, but also eases the learning of the network. For example, in Convolutional Neural Networks, each layer is responsible of one elementary operation and evaluates the results of the layer.

Deep learning gained huge attention with the increase of the size of the datasets and with the advancements of new data processing hardware. The most important factor that deep learning achieves good performance is the amount of the data to train the deep learning architectures. As the size of the data increases, the accuracy of the deep architecture increases.

Deep learning has several deep architectures, such as Deep Neural Networks (DNN), Convolutional Neural Networks (CNN), Deep Belief Networks (DBN), and Recurrent Neural Networks (RNN) (LeCun et al., 2015). DNN and CNN are

feed-forward architectures that contain multiple layers of transformations and nonlinearity with the output of each layer that are feeding subsequent layer. RNN is a recurrent architecture that has both forward pass which transfers information to subsequent layers and recurrent pass that processes past information and current input together.

Deep Neural Networks (DNN) are a type of Artificial Neural Networks (ANN) with multiple layers between input and output layers (Dahl et al., 2011; Dahl et al., 2013). The main aim of DNN is to find a proper mathematical modelling for a given input to obtain the output. In exploration of mathematical explanation, DNN considers both linear and non-linear relations of input and hidden vectors to achieve the desired output. Figure 2.6 presents an example of a DNN structure.

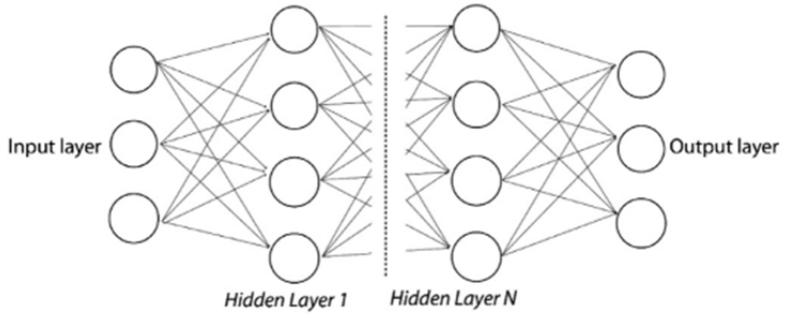

Figure 2.6. Example of a DNN structure

DNN are powerful for modelling complex non-linear relationships of input-output pairs. Because the DNN have multiple hidden layers and have many neurons in the layers, the DNN have the ability to model and process multiple features. DNN are strong alternatives to ANN, however have two main challenges. First of all, the error is propagated to first layers and the effect of the error to other layers is minor. Second, the learning process of DNN is slow due to complex and high-order matrix multiplications.

Convolutional Neural Networks (CNN) are a type of deep learning architectures that is specialized to be mostly used in analyzing visual datasets (Abdel-Hamid et al., 2014). In CNN, layers are utilized to perform one specific job, i.e. convolution, or sub-sampling, and then the network is connected to a fully connected deep architecture to produce an output. In each convolution and sub-sampling layers, the higher level features are extracted from the input images and the output becomes more accurate. A sample CNN architecture is presented in Figure 2.7.

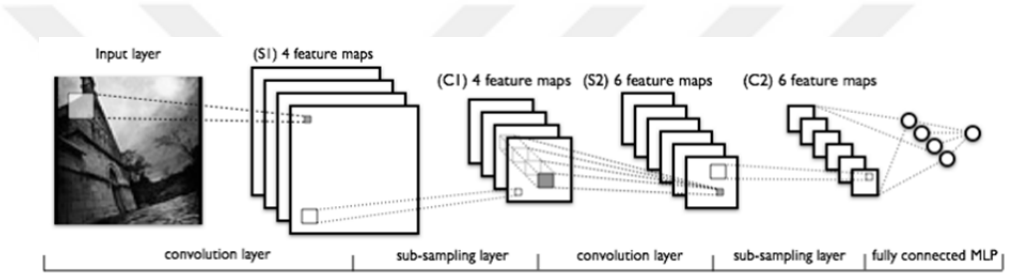

Figure 2.7. Example of a CNN structure

CNN are successful architectures in image analysis and processing, because they have the ability to exploit spatial and temporal correlation in the datasets. Convolution layer explores useful local features from the input data, and sub-sampling layer gets the features and summarizes the results. With the help of these steps, CNN have the ability to extract features automatically. However, CNN have some challenges based on their operation and result extraction. First of all, the process of CNN is black-box and the interpretation and explanation are hard. Second, selecting appropriate hyper-parameter values are important for the performance of CNN. Third, efficient training of CNN requires powerful computational resources. Fourth, CNN perform poorer performance with noisy and un-labelled datasets.

Deep Belief Networks (DBN) are a type of deep learning architectures that are probabilistic generative graphical models composed of multiple layers of

hidden units (Testolin et al., 2020; Ou et al., 2019). These hidden units take binary values and are also called as feature detectors. The top two layers have undirected connections while lower layers are directed in a top-down structure. The hidden unit states at the lowest layer presents input data vector. Typically, DBN employs Restricted Boltzmann Machines (RBM) for connecting layers with each other. The main idea of DBN is to use unsupervised learning method to train each RBM layer by layer and then fine-tune the network with supervised learning. Figure 2.8 presents the basic structure of DBN.

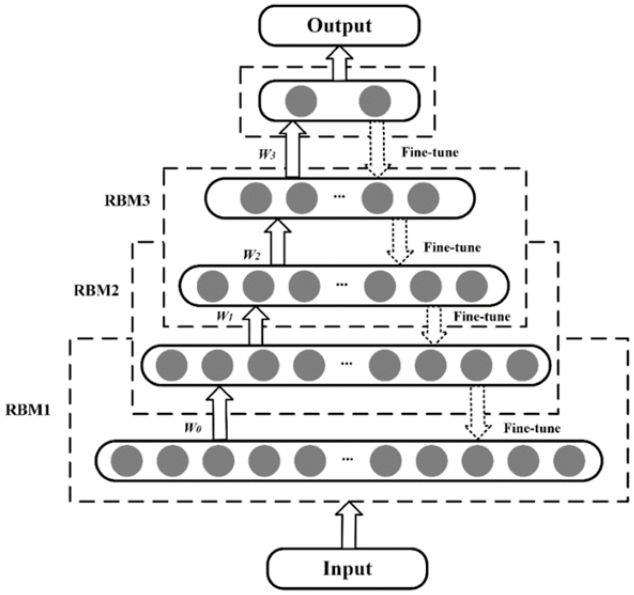

Figure 2.8. Example of a DBN structure

DBN construct a strong correlation between layers. The learning from the layers in a top-down approach is efficiently performed and generative weights determine how the variables in one layer are connected with the variables in the connected layer. Also, after the learning phase, a fine-tune phase is performed to better model the connections between the layers. DBN are widely used as a pre-training systems which determines the weights of deep architectures.

Recurrent Neural Networks (RNN) is a type of deep learning architectures which is capable of handling large sequential inputs (Graves et al., 2013). Main idea behind RNN is to extract outputs of current time step based on current input and previous inputs with weighted manner. This approach is beneficial for several tasks which needs information about previous inputs, such as speech recognition, natural language processing. The weights of input-to-hidden, hidden-to-hidden, and hidden-to-output do not change along the network. A sample RNN architecture is presented in Figure 2.9.

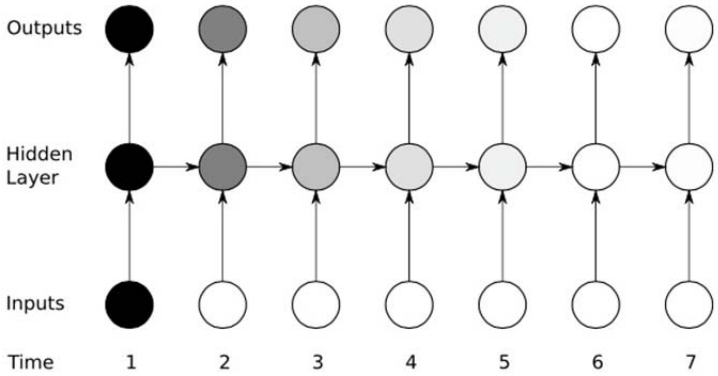

Figure 2.9. Example of a RNN structure

In this thesis study, we used LSTM that is a variant of RNN as a deep learning system. We selected LSTM because of its popularity and success in speech recognition domain over other candidate deep architectures. The details of RNN and LSTM are provided in Section 2.2.1 and 2.2.2.

2.2.1. Recurrent Neural Networks (RNN)

Recurrent neural networks (RNN) are a type of deep learning architectures that are used to model and process sequential data (Graves et al., 2013, Graves et al., 2013b). The RNN takes $x = x_1, x_2, \ldots, x_T$ as input and computes hidden vector

of $h = h_1, h_2, \ldots, h_T$ and output vector of $y = y_1, y_2, \ldots, y_T$ by using Equation (2.5) and Equation (2.6) (Graves et al., 2013, Graves et al., 2013b).

$$h_t = H(W_{xh}x_t + W_{hh}h_{t-1} + b_h) \tag{2.5}$$

$$y_t = G(W_{hy}h_t + b_y) \tag{2.6}$$

where W is the weight matrices of input to hidden, hidden to hidden and hidden to output vectors, b is the bias, H and G are the activation functions for hidden layer and output, and t is the time step indicator of the parameters.

Figure 2.10 presents the basic structure of a general RNN and the structure of RNN after it is unfolded over time. As can be seen in the figure, the weights of inputs (W_{xh}), hidden vectors (W_{hh}) and outputs (W_{hy}) are same for each time step t, but the weights of inputs, hidden vectors and outputs could be different from each other. In other words, all input weights are same for the whole network, but these weights are different from weights of hidden vectors and outputs.

RNN has a good performance on time-dependent or sequential data because of its dependency based modelling capacity. As shown in Equation (2.5) and Equation (2.6), current hidden value is based on current input and previous hidden value. Current output of RNN changes with the change of current input and previous hidden values. This dependency provides RNN to efficiently process sequential inputs and generate satisfying results.

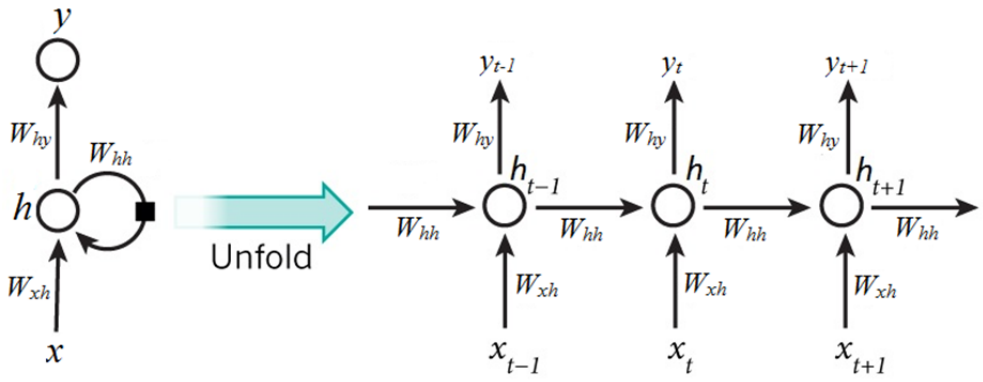

Figure 2.10. The basic structure and unfolding of RNN

There are several variants of RNN which have difference in their network structure. The most popular and known variants of RNN are Elman networks, Jordan networks, Hopfield network, Bidirectional associative memory network, Echo state network, Gated recurrent unit, and Long-Short-Term Memory (LSTM) networks. All of these architectures are based on RNN, but have several differences and additions with respect to classical RNN.

RNN is a good candidate for processing sequential data such as speech data, however, has several limitations. First, it has vanishing and exploding gradient problem. When there is a long sequence of deep model, the training errors in RNN vanishes or explodes while updating the weights which causes inefficient learning of the deep model. Second, because of vanishing and exploding gradient problem, it cannot process and correlate longer sequences. To overcome these limitations Long-Short-Term Memory (LSTM) networks are proposed (Graves et al., 2013b).

2.2.2. Long-Short-Term Memory (LSTM) RNN

LSTM networks are a variant of RNN that propose to use gates for better modelling of input data sequences (Graves et al., 2013b). In a standard LSTM network, there exists three layers, input, LSTM and output layers. Input layers

carries information of each time step to LSTM layer. LSTM layer produces outputs for every input instance and provides to the output layer.

Three gates are proposed in a custom LSTM network, namely input, forget, and output gates. Input gate controls the proportion of the input. Forget gate controls the proportion of the previous steps that will be allowed to the next step. Output gate controls the proportion of the cell output that will be given as output. The equations of the cell state, input, forget and output gates and output of the cell are provided in Equation (2.7) to Equation (2.11) (Graves et al., 2013b).

$$i_t = \sigma(W_{xi}x_t + W_{hi}h_{t-1} + W_{ci}c_{t-1} + b_i) \tag{2.7}$$

$$f_t = \sigma\left(W_{xf}x_t + W_{hf}h_{t-1} + W_{cf}c_{t-1} + b_f\right) \tag{2.8}$$

$$c_t = f_t c_{t-1} + i_t tanh(W_{xc}x_t + W_{hc}h_{t-1} + b_c) \tag{2.9}$$

$$o_t = \sigma(W_{xo}x_t + W_{ho}h_{t-1} + W_{co}c_t + b_o) \tag{2.10}$$

$$h_t = o_t tanh(c_t) \tag{2.11}$$

In Equation (2.7) to Equation (2.11), σ is the sigmoid function, i, f, o, and c are input gate, forget gate, output gate, and cell state activation vectors, h is the hidden vector, and W are the weight matrices. The schematic view of LSTM is presented in Figure 2.11.

LSTM could handle long-term dependencies with the help of Equation (2.7) to Equation (2.11) and could produce more reliable models with respect to RNN. As can be seen in these equations, LSTM selects which proportion of the information will be transferred to new cell state and which proportion of the input

will be processed. Also, output activation mechanism provides proportional extraction of output state.

LSTM have several hyper-parameters which should be selected based on the application requirements. These parameters are number of layers, number of hidden units, mini-batch size, optimizer, learning rate and number of iterations. These parameters should be selected carefully based on application of LSTM on different datasets.

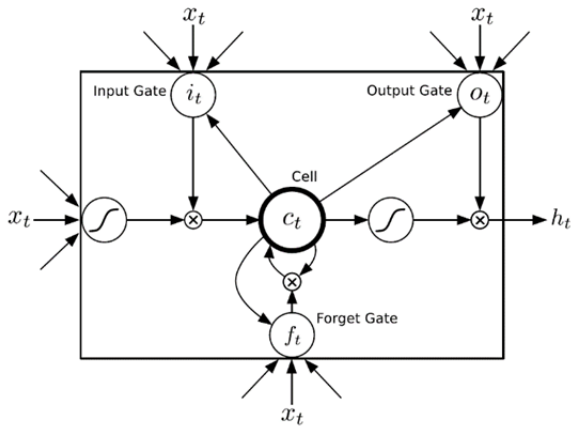

Figure 2.11. LSTM network overview (Graves et al., 2013b)

2.2.2.1. LSTM Variants

There are many variants of LSTM, each of which differ from and propose advancements to traditional LSTM at different aspects. The variants of LSTM advance architecture or training of the deep networks to obtain better performance. These proposed variants of LSTM provide extra performance for some focused application areas or deep learning domain itself.

One variant of LSTM is to add weighted peephole connections to the gates of the LSTM memory blocks (Gers et al., 2002). These peephole connections provide the gates to investigate the current cell state and decide which proportion of the information should be passed. With this modification, LSTM could learn the distinctions between sequences of spikes.

Other popular variant of LSTM is bidirectional LSTM which connects the units both forward and backwards (Graves et al., 2013b). Bidirectional LSTM (BiLSTM) provides more flexible approach to train the network using the data of both behind and ahead of the current time step.

Another variant is Gated Recurrent Unit (GRU) LSTM which combines input and forget gates into an update gate (Cho et al., 2014). Also, cell state and hidden output are combined into a hidden state layer. With these modifications GRU became a simpler LSTM model and requires less input and computations.

Fourth variant is multiplicative LSTM (mLSTM) which combines LSTM and multiplicative recurrent neural networks architecture (Krause et al., 2017). mLSTM model has different recurrent transition functions for each input. This variant is widely used in especially natural language processing applications.

2.3. Connectionist Temporal Classification (CTC)

Connectionist Temporal Classification (CTC) is an output layer for recurrent neural networks which was introduced by Graves et al. (2006). CTC is proposed for temporal classification tasks where the input and output do not overlap and the alignment between the input and output is not known in advance, such as sequence labelling, keyword spotting, handwriting recognition and speech recognition. CTC is proposed to overcome the drawbacks of NN-HMM hybrid methods which require a pre-segmentation on input data and post-processing to transform network outputs into label sequences.

CTC provides two important benefits with respect to classical NN-HMM hybrid speech recognition systems. First one is that, NN-HMM hybrid methods require pre-segmented training data which should be performed and provided as the input to neural networks. In other words, neural networks require the inputs as segmented and labelled. Segmentation and labelling the time steps causes the network to learn local classifications and global aspect is missed. Second, a post-processing is required to transform the network outputs into the label sequences

which is not a trivial task. CTC achieves to handle these problems with allowing the network to make predictions at each time step of the input sequence. Also, CTC prevents the post-processing step by providing probabilities of complete label sequences. The difference between CTC and framewise label classification of speech input signals is presented in Figure 2.12.

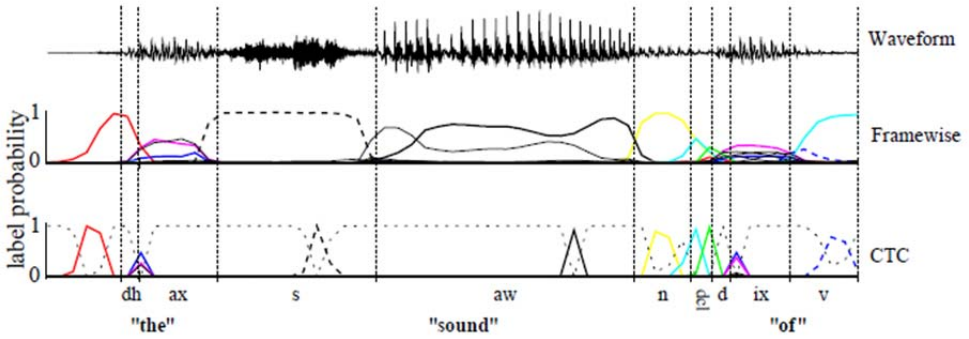

Figure 2.12. Overview of CTC in comparison with standard framewise RNN (Graves et al., 2006; Graves, 2012)

As can be seen in Figure 2.12, framewise RNN tries to both predict probabilities of each phoneme, and align the sequences with respect to the manual segmentation which is shown in vertical lines. Because of alignment forcing in framewise RNN, it produces errors for misaligned segment boundaries. However, CTC predicts only the sequence of phonemes without requiring to align the predicted phonemes with the manual segmentation by using a series of spikes separated by blank labels. The outputs of CTC could be observed by following the spikes in the sequence, however, the outputs of framewise RNN should be post-processed to decide what phoneme is predicted.

In speech recognition, CTC is applied on top of neural network layers to perform alignment between output of the network and the target sequences and to compute loss and error rates of the network models. CTC layer has a softmax output layer which has one more label than the number of labels in the input labels

list *L*. The activations of the first *L* labels are predicted by using the corresponding labels at particular time steps, and the activation of the extra one label is the probability of *blank* where none of the labels are predicted. The final sequence of labels is used to define a distribution over all possible sequences of length up to the input label sequence.

Formally, for a given input sequence *x* of length *T*, let us the outputs of the network is $y = N_W(x)$ and y_k^t is the predicted label *k* at time *t*. Then, if we assume that the output probabilities are conditionally independent based on *x* input sequences, we can produce Equation (2.12) (Graves et al., 2006; Graves, 2012).

$$p(\pi|x) = \prod_{t=1}^{T} y_{\pi_t}^t, \forall \pi \in L \tag{2.12}$$

In Equation (2.12), the sequences of π are referred as *paths* that are produced with the outputs of the network. After, the output *paths* and possible labellings of *x* are used in a many-to-one function *F* to produce predicted labels. In this process, first, the repetitions and blank labels are removed in the *path*. For example, if the blank is symbolized as '-' and for an output *path* "cc--aa-a-a---tt-t-t" is treated as "cat". Also other possible outputs are linked with "cat", such as "ccc-a-a--a---t-t-t" or "cc---a-aa-tt-ttt-t" (Graves et al., 2006; Graves, 2012).

The probabilities of labels *l* for given input sequence *x* is calculated by summing the probabilities of all *paths* that are provided by *F* as given in Equation (2.13) (Graves et al., 2006; Graves, 2012).

$$p(l|x) = \sum_{\pi \in F^{-1}(l)} p(\pi|x) \tag{2.13}$$

The conditional probability of $p(l|x)$ should be calculated in an efficient way since it requires too many calculations and there are many combinations. CTC uses dynamic programming algorithm like forward-backward algorithm. Figure

2.13 presents an example of label sequence “cat” in terms of forward-backward algorithm. The black nodes present *blank* outputs and white nodes present the labels. As can be seen in the figure, forward variables are updated in the direction of arrows and backward variables are updated in the opposite direction of the arrows. The process is efficiently computed using recursive computations of forward and backward variables.

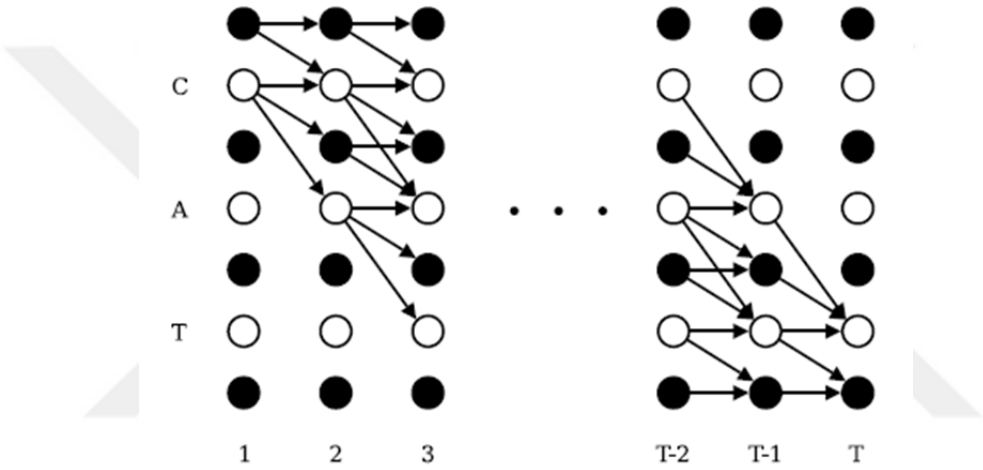

Figure 2.13. CTC forward-backward algorithm example of “cat” (Graves et al., 2006; Graves, 2012)

CTC based neural networks use CTC derived objective function to calculate loss and train the networks based on the loss of the network outputs with respect to the actual label sequences using gradient descent. CTC loss $L(S)$ over a training set S is presented in Equation (2.14) (Graves et al., 2006; Graves, 2012).

$$L(S) = -ln \prod_{(x,z) \in S} p(z|x) = -\sum_{(x,z) \in S} \ln p(z|x) \tag{2.14}$$

Because the Equation (2.14) is differentiable, gradient descent algorithm variants can be used to train CTC based neural networks using CTC loss as an

objective function over training set and network weights. CTC loss is used for calculation of training and test costs of the neural network with respect to the given training and test datasets. The loss calculation of CTC is the multiplication of each label sequences for each training sample and the addition of multiplication results of all of the training samples in the dataset. By this way, CTC learns which paths are closer to the target text and which paths are far from the required target text.

In CTC-based speech recognition systems, generally Label Error Rate (LER) is used to measure the efficiency of the constructed model. LER calculates edit distance between recognized text and the target text and produces a similarity score. If LER is smaller, then the network learns to recognize input speech signals better. The formulation of LER is provided in Equation (2.15) (Graves et al., 2006; Graves, 2012).

$$LER(h,S) = \sum_{(x,z) \in S} EditDist(h(x), z) \tag{2.15}$$

In Equation (2.15), h is the classifier, S is the test samples, x is input features, and z is the target text with respect to x.

After the network is trained, any unknown input sequence x can be labelled using the most probable label sequence l^*, which is generally known as decoding. For decoding purposes, two popular methods are present in CTC, i.e. best path decoding and prefix search decoding (Graves et al., 2006; Graves, 2012). Also, in decoding step, language model could be adapted to improve performance of end-to-end speech recognition.

2.4. Gradient Descent Optimization

Gradient descent optimization is one of the most popular approach for training and optimizing deep learning models. It is based on backpropagation algorithm that propagates network output errors backward and updates parameters.

Gradient descent algorithm uses first-order derivative and is more successful than higher-order derivative based algorithms for training deep learning models. Gradient descent algorithm aims to minimize objective function which is generated by network weights by updating the network's weights in the opposite direction of gradient of the objective function until the desired learning rate is achieved.

Several optimization algorithms are present in gradient descent optimization method, such as Momentum, Nesterov Accelerated Gradient (NAG), Adagrad, Adaptive Moment Estimation (Adam), Adadelta, and RMSProp. These algorithms differ on the calculation of gradient and selection of the next point. Each algorithm is successful at some specific problems. However, Adam optimizer is generally seen as the best optimizer among the gradient descent based optimization algorithms (Ruder, 2016).

There are three main key points of gradient descent to be carefully taken into account. First of all, learning rate is an important parameter that should be selected carefully and based on the dataset characteristics. Secondly, local minimum points should be carefully evaluated. Thirdly, same learning rate might not yield good performance and adaptive learning rates based on gradient of the data could be used.

Gradient descent optimization has three variants, batch gradient descent, stochastic gradient descent, and mini-batch gradient descent (Goodfellow et al., 2016; Ruder, 2016). Batch gradient descent calculates the gradient and performs weight updates after it traverses on all of the data points in the dataset. Stochastic gradient descent calculates the gradient and updates the weights after each data point in the dataset. Both of these methods have strong and weak aspects and to handle their weaknesses, mini-batch gradient descent is proposed. In mini-batch gradient descent, the calculation of the gradients and the updates of the weights are performed after a number of data instances are traversed. In other words, each of the gradient descent variants use same formulation, however differ in the usage of the amount of the data before performing an update on the parameters.

Batch gradient descent, in other words *vanilla* gradient descent, algorithm performs updates based on Equation (2.16) (Bottou, 2010; Ruder, 2016).

$$w_{t+1} = w_t - \eta_t \frac{1}{TotalNo.ofData} \sum_{i=1}^{TotalNo.ofData} \nabla_w Q(z_i, w_t) \quad (2.16)$$

In Equation (2.16), w_t is model weights, η is the learning rate of the algorithm, ∇_w is the gradient, and Q is the loss function.

As can be seen in Equation (2.16), to calculate gradients and update model parameters, batch gradient descent needs to loop over entire dataset. Batch gradient descent provides stable convergence to the neural networks, however, the convergence rate and the speed of batch gradient descent is slow. For this reason, batch gradient descent is slow and impractical for datasets that do not fit into memory. Furthermore, it does not allow to update model parameters online using new examples.

2.4.1. Mini-Batch Gradient Descent

Mini-batch gradient descent is a combination of batch gradient descent and stochastic gradient descent. Mini-batch gradient descent performs an update to the model parameters after every mini-batch of *n* training samples are traversed. Mini-batch gradient descent is proposed to overcome weakness of each approach on updating the weights and calculation of gradients. The formula of mini-batch gradient descent algorithm is given in Equation (2.17) (Bottou, 2010; Ruder, 2016).

$$w_{t+1} = w_t - \eta_t \frac{1}{n} \sum_{i=1}^{n} \nabla_w Q(z_i, w_t) \quad (2.17)$$

As can be seen in Equation (2.17), mini-batch gradient descent calculates gradients and updates weights when *n* data instances are traversed (Goodfellow et al., 2016; Bottou, 2010). By this strategy, mini-batch gradient descent reduces

variance in the model parameter update and provides more stable convergence than stochastic gradient descent and is much more efficient on computation of gradients than batch gradient descent.

Mini-batch gradient descent optimization provides more efficient and stable convergence to the networks, however it has a main problem. One of the main problems in mini-batch gradient descent is that it selects mini-batch samples randomly from the data instances. This could have the potential to provide some of the data instances more important that their actual importance by selecting these instances in the same mini-batch. Especially in speech recognition, there are several factors that affect speech recognition performance, such as accent, gender, and age. When the data instances are selected randomly, these features, especially accent feature, could not be selected properly based on their actual presence and this could result in wrongly training of the speech recognition system. If data instances of mini-batches could be selected with respect to the presence of these features and when proper mini-batches could be generated, the performances of deep learning based speech recognition could be improved and better speech recognition models could be constructed based on input dataset.

2.5. Speech Recognition using Deep Learning

With the emerging popularity and success of deep learning architectures on non-trivial tasks, it is also considered to be used in speech recognition which is a difficult problem for machines. Earlier studies related to speech recognition with deep learning focus on using deep learning architectures as a supportive part of different speech recognition systems to increase recognition performance. First of all, GMM-HMM state-of-the-art speech recognition systems evolved to DNN-HMM speech recognition systems because DNN performed better than GMM in terms of frame classification with the help of big speech datasets and efficient training algorithms. Several deep architectures are also used with DNN-HMM

speech recognition systems to further increase performance, such as Context Dependent DNN (CD-DNN) and Deep Belief Network trained DNN.

The advances in deep learning architectures and higher performance ability of these networks led speech recognition systems to use new deep architectures for speech recognition tasks, especially CNN and RNN. CNN is used in speech recognition for feature extraction and taking spectral powers as image inputs to speech recognition systems which is called *spectrograms* and to extract features implicitly within the deep learning system to improve feature relations and to monitor unknown associations of the features. RNN is used in speech recognition due to its high success rate on sequential data processing ability. RNN could extract relations between time steps of input features and could model long-term dependencies better than any other deep learning architectures.

With these powerful architectures, deep learning based speech recognition studies transformed to end-to-end deep learning systems for speech recognition to benefit the power of deep learning systems and to efficiently process huge speech datasets. In end-to-end deep learning based speech recognition systems, each step of speech recognition is performed by one or more deep learning architectures. For end-to-end speech recognition task, CNN and RNN are seen more successful deep architectures with respect to other candidates. Especially RNN outperformed other deep learning systems in terms of end-to-end speech recognition. RNN and its more powerful variant LSTM are highly popular on utilization of speech recognition tasks.

Figure 2.14 presents the utilized end-to-end speech recognition system of this thesis study. As can be seen in the figure, all recognition processes are performed using deep learning systems and operations. First of all, the speech signal is taken as input and the MFCC features are extracted from the raw speech signal. Second, the MFCC features are taken into an LSTM network which is composed of 5-layers and 600 units. LSTM network extracts label predictions for each time step of the input MFCC features. CTC layer takes the predictions of each

time step and converts these predictions to the output of the speech recognition system. In CTC layer, both loss of the neural network is calculated and decoding is performed which performs alignments of temporal predictions into actual predictions. CTC loss is used for calculating the gradients. Decoding is used for aligning and reducing the count of the neural network outputs with respect to the target text.

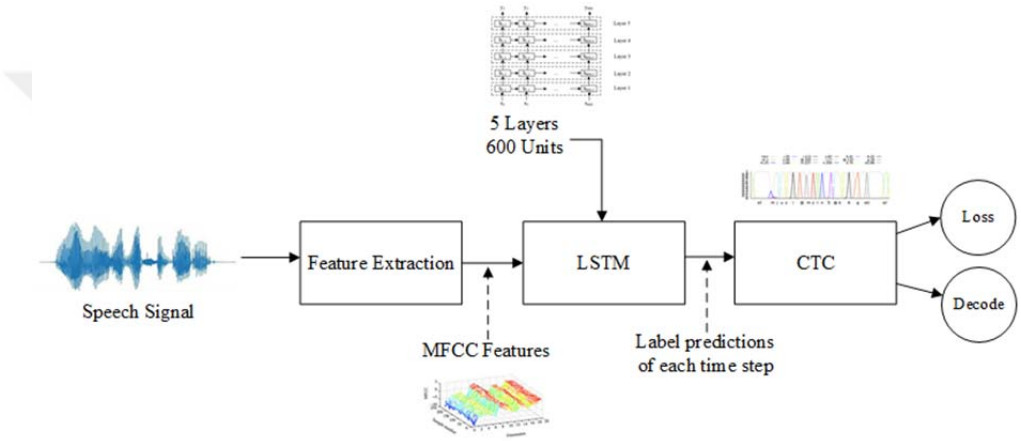

Figure 2.14. Utilized LSTM-CTC speech recognition method of this thesis

3. MINI-BATCH SAMPLE SELECTION STRATEGIES

In this section, we present mini-batch sample selection strategies of deep learning systems for speech recognition applications which are used in this thesis. First, standard random mini-batch sample selection strategy is presented, and then proposed gender and accent adjusted strategies are presented. Finally, hybrid mini-batch sample selection strategies which combine proposed gender and accent adjusted strategies are presented.

Two types of strategies are proposed for gender and accent adjusted strategies, namely homogeneous and heterogeneous sample selection strategies. In homogeneous sample selection strategies, only one gender or one accent is present in one mini-batch. In heterogeneous sample selection strategies, equal number of gender or equal ratio of accents are present in one mini-batch. In other words, homogeneous strategies contain only one type of meta parameter, contrarily, heterogeneous strategies contain all of the meta parameters in one mini-batch. In hybrid strategies, proposed homogeneous gender, heterogeneous gender, homogeneous accent, and heterogeneous accent strategies are combined with each other and homogeneous gender – homogeneous accent, homogeneous gender – heterogeneous accent, heterogeneous gender – homogeneous accent, heterogeneous gender – heterogeneous accent strategies are proposed.

Figure 3.1 presents the overview of proposed method. First of all, the corpus is read and the speech features are extracted from the corpus. Then, the dataset is divided into training and testing sets. The training set is further split into mini-batches based on the strategies proposed in this study. As shown in the figure, two types of strategies are used, i.e. unified strategies and hybrid strategies. In unified strategies, gender and accent adjusted strategies are present. For each strategy, homogeneous and heterogeneous sample selection strategies are proposed. In hybrid strategies, proposed hybrid strategies are present which are developed using gender and accent adjusted strategies. After, the mini-batches are given to deep

learning based speech recognition model which uses LSTM architecture and speech recognition model is generated using these mini-batches. Then, test data is applied on the model. Finally, the results are extracted for identification of the proposed strategies.

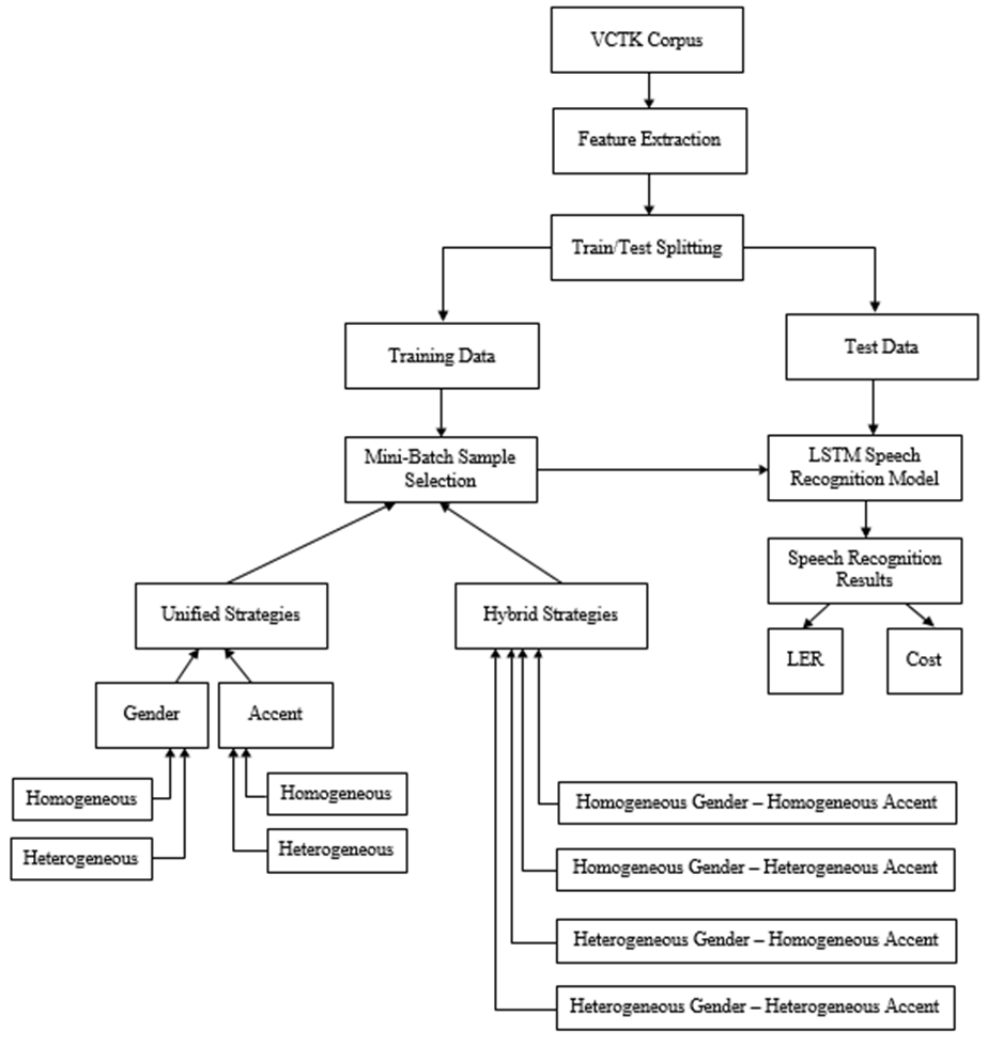

Figure 3.1. Overview of proposed method

3.1. Standard Mini-Batch Sample Selection Strategy

Standard mini-batch sample selection strategy is the most straightforward way of selecting mini-batch samples from the training dataset. In standard mini-batch sample selection, the speech dataset is shuffled and the mini-batch samples are selected randomly from the shuffled speech dataset. There is no intelligent sample selection strategy used for mini-batches to increase performance of speech recognition system by selecting appropriate samples. The usage of standard mini-batch sample selection is because of its easy and fast implementation to the deep learning systems. The pseudo code for standard mini-batch sample selection strategy is given in Algorithm 1.

Algorithm 1. Standard strategy

```
1.  initialization: Minibatch_samples = null
2.  all_files = read-dataset-files(Dataset)
3.  shuffle(all_files)
4.  batch_count = num_examples / batch_size
5.  for i in batch_count
6.     batch_samples = null
7.     batch_samples.add(all_files [0:batch_size])
8.     all_files.remove[0:batch_size]
9.     Minibatch_samples.add(batch_samples)
10. end for
11. return Minibatch_samples
```

In Algorithm 1, in step 1, *Minibatch_samples* array is initialized to fill with standard strategy. In step 2, all of the files in the dataset are read into *all_files* array. In step 3, *all_files* array is shuffled to get samples randomly from the dataset. In step 4, batch count is calculated using number of examples and batch size

variables to identify the number of mini-batch samples. In steps between 5 and 10, mini-batch samples are selected as standard random sample selection approach. In step 6, *batch_samples* array is initialized for filling the samples in a mini-batch. In step 7, batch samples are sequentially selected from the randomized *all_files* array and added to *batch_samples* array. The sequential selection of mini-batch samples would not be a problem because *all_files* array is shuffled. In step 8, selected files are removed from *all_files* array to prevent re-selection of same file as a mini-batch sample. In step 9, *batch_samples* array is added to global *Minibatch_samples* array. In step 11, *Minibatch_samples* array is returned as the output of standard mini-batch sample selection algorithm.

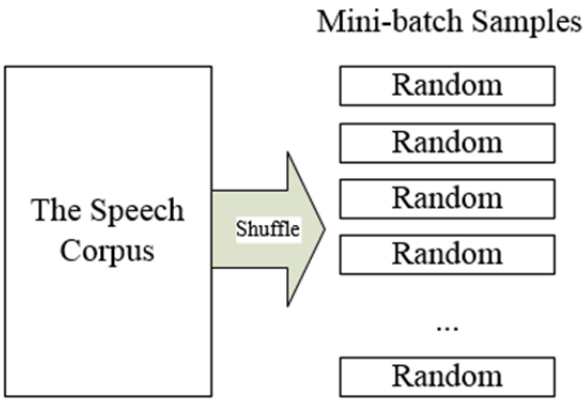

Figure 3.2. Standard mini-batch sample selection strategy

Figure 3.2 illustrates the standard mini-batch sample selection strategy. As can be seen in the figure, standard mini-batch sample selection strategy has no clever selection method. Firstly, it reads the speech corpus and shuffles the instances in the corpus. Then, it randomly selects samples for each mini-batch and fills the batches. Standard mini-batch sample selection strategy is simple and easy to implement. However, due to its random selection of mini-batch samples, it could produce unbalanced mini-batches in terms of selected samples. Especially for the

corpuses that have various accents and different speaker profiles, using standard mini-batch sample selection strategy would not yield good speech recognition performance and reduce accuracy of speech recognition systems. For these applications, instead of standard mini-batch sample selection, advanced sample selection algorithms should be preferred and employed.

3.2. Gender Adjusted Strategies

Gender adjusted mini-batch sample selection strategies use gender information of speech datasets to select mini-batch samples. Gender is an important factor to achieve good speech recognition performance. Thus, in this thesis study, gender adjusted strategies are proposed to investigate the effect of gender on deep learning based speech recognition performance.

In gender adjusted mini-batch sample selection, the speech dataset is divided into two separate categories, namely, female speakers and male speakers, and the utterances of the speakers are categorized based on their gender. Then, the mini-batch samples are selected from these categories with respect to the selected strategy. Two strategies are proposed which use these two categories, i.e. homogeneous and heterogeneous gender strategies.

In homogeneous gender strategy, a mini-batch is filled with speakers of only one gender, i.e. female or male speakers. Sequential mini-batches are constructed with different gender. Contrarily, in heterogeneous gender strategy, each mini-batch is filled with equal size of utterances of female and male speakers. Furthermore, the samples for each mini-batch are selected randomly and after the mini-batch samples are selected, the mini-batch is re-shuffled to prevent wrong learning of the model based on gender ordering.

The difference in the strategies could reveal the performances of deep learning based speech recognition with respect to gender information presence in mini-batches and would reveal gender related performance comparison of the

strategies. Algorithms 2 and 3 presents homogeneous and heterogeneous gender adjusted mini-batch sample selection strategies.

Algorithm 2. Homogeneous gender strategy

```
1.  initialization: Minibatch_samples = null
2.  [female_speakers, male_speakers] = get-dataset-info(Dataset_Meta)
3.  [female_filenames,male_filenames]=get-files(female_speakers,
    male_speakers,Dataset)
4.  shuffle(female_filenames)
5.  shuffle(male_filenames)
6.  batch_count = num_examples / batch_size
7.  for i in batch_count
8.    batch_samples = null
9.    if i%2 == 0
10.     batch_samples.add(female_filenames[0:batch_size])
11.     female_filenames.remove[0:batch_size]
12.   else
13.     batch_samples.add(male_filenames[0:batch_size])
14.     male_filenames.remove[0:batch_size]
15.   end if
16.   Minibatch_samples.add(batch_samples)
17. end for
18. return Minibatch_samples
```

Algorithm 2 presents homogeneous gender mini-batch sample selection strategy. In Algorithm 2, in step 1, *Minibatch_samples* array is initialized to fill with homogeneous gender strategy. In step 2, the dataset meta information is read with get-dataset-info function and female and male speakers taken into separate

arrays. In step 3, the file names of female and male speakers are read from the dataset using female and male speaker information and stored into the arrays of *female_filenames* and *male_filenames*, respectively. In steps 4 and 5, the female and male file names are shuffled to get files randomly from each category. In step 6, batch count is calculated using *num_examples* and *batch_size* parameters. In steps between 7 and 17, mini-batch samples are selected according to the gender of the speakers. In step 8, *batch_samples* array is initialized for filling the samples in a mini-batch. In step 9, the consecutive batches are controlled to be in the order of female and male speakers using the mode operation. Female speakers are identified as 0 and male speakers are identified as 1 from the mode 2 operation. If female speaker will be used in current batch, steps 10-11 are executed. In step 10, the samples are selected from *female_filenames* array for the amount of *batch_size*. In step 11, to prevent selection of same file name twice, the selected file names are removed from *female_filenames* array. A similar process is executed for the male speakers in steps 13-14. *male_filenames* array is used in these steps to get files of male speakers instead of *female_filenames* array of steps 10-11. In step 16, the *batch_samples* array is added to the array of *Minibatch_samples*. Finally, in step 18, *Minibatch_samples* array is returned as the output of homogeneous gender mini-batch sample selection algorithm.

Figure 3.3 presents homogeneous gender mini-batch sample selection strategy. As can be seen in the figure, first of all, the speech corpus is read from the memory. Then the speakers are divided into two groups, i.e. female speakers and male speakers. The mini-batch samples are selected based on their gender which are divided and categorized. For the first mini-batch, random files are selected from only one gender class as the count of *batch_size*. In this study, first mini-batch is filled with files of female speakers. For the second mini-batch, random files are selected from other gender class as the count of *batch_size*. In this study, second mini-batch is filled with files of male speakers. Successive mini-batches are filled with different genders. The files are randomly selected form female and male

speakers lists to reduce sequence and speaker dependency of deep learning systems training using mini-batches. Finally, the selected mini-batch samples are fed into deep learning system to generate a speech recognition model.

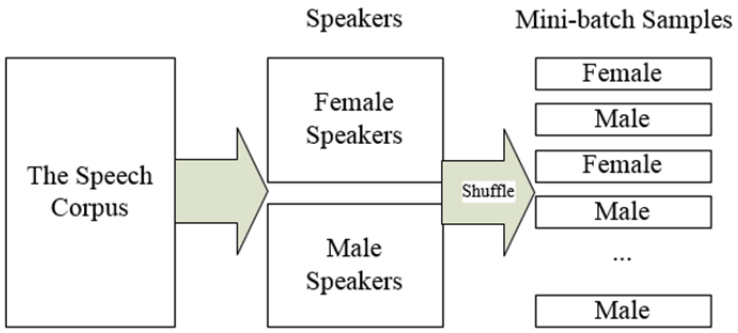

Figure 3.3. Homogeneous gender mini-batch sample selection strategy

With using homogeneous gender mini-batch sample selection strategy, the aim is to train deep learning based speech recognition system with the features of one gender in one mini-batch. In other words, the deep learning based speech recognition system would be trained with the files of only one gender before updating the weights. By this approach, the effect of unified gender on training of deep learning based speech recognition systems at mini-batches are evaluated by this strategy.

Algorithm 3 presents heterogeneous gender mini-batch sample selection strategy. Algorithm 3 is similar to Algorithm 2 in context of preparation of files. First of all, the speakers are divided into two categories, i.e. female and male speakers. Then, the files of each gender are extracted and stored in two arrays, i.e. *female_filenames* and *male_filenames*. The filenames are shuffled to select randomized speaker files from the categories. The difference of Algorithm 3 with Algorithm 2 is the loop of batch sample selection approach which is the steps between 7 and 15. In steps between 7 and 15, mini-batch samples are selected in heterogeneous manner using gender information. In step 8, *batch_samples* array is

initialized for filling the samples in a mini-batch. In steps 9 and 10, equal size of female and male speaker file names are added to *batch_samples* array that are half of the *batch_size* parameter to provide equal presence of each gender in one mini-batch. The selected and added files are removed from *female_filenames* and *male_filenames* arrays to prevent re-selection of the same file and wrongly biased mini-batch samples. In step 14, the generated *batch_samples* array is added to *Minibatch_samples* array. Finally, in step 16, *Minibatch_samples* array is returned as the output of heterogeneous gender mini-batch sample selection algorithm.

Algorithm 3. Heterogeneous gender strategy

1. initialization: *Minibatch_samples* = **null**
2. [female_speakers, male_speakers] = get-dataset-info(*Dataset_Meta*)
3. [female_filenames,male_filenames]=get-files(female_speakers, male_speakers,*Dataset*)
4. shuffle(female_filenames)
5. shuffle(male_filenames)
6. batch_count = *num_examples* / *batch_size*
7. **for** i **in** batch_count
8. batch_samples = **null**
9. batch_samples.add(female_filenames[0:*batch_size*/2])
10. batch_samples.add(male_filenames[0:*batch_size*/2])
11. female_filenames.remove[0:*batch_size*/2]
12. male_filenames.remove[0:*batch_size*/2]
13. shuffle(batch_samples)
14. Minibatch_samples.add(batch_samples)
15. **end for**
16. **return** Minibatch_samples

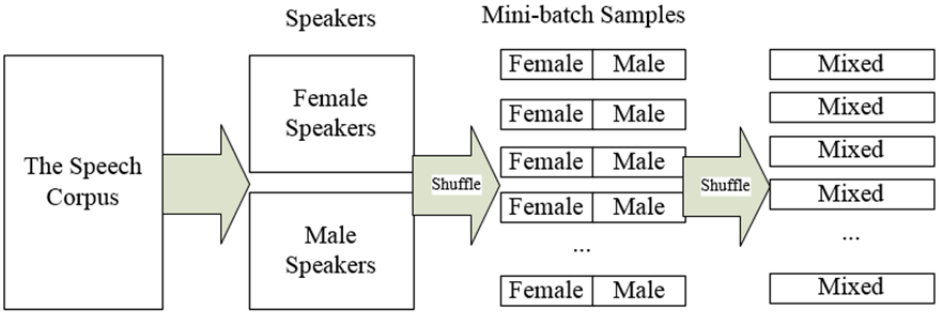

Figure 3.4. Heterogeneous gender mini-batch sample selection strategy

Figure 3.4 presents heterogeneous gender mini-batch sample selection strategy. As can be seen in the figure, heterogeneous gender mini-batch sample selection strategy has similar preparation procedures with homogeneous gender strategy on reading the speech corpus and dividing and categorizing the speakers based on their gender. However, the main difference is that, in heterogeneous gender strategy, one mini-batch is filled with equal size of different genders. Half of the mini-batches are filled with using files of female speakers and the other half of the mini-batches are filled with using files of male speakers. Then, the selected files are further shuffled and a mix of female and male speaker files are extracted for each mini-batch. The files are randomly selected from female and male speakers file lists to reduce sequence and speaker dependency of deep learning systems in mini-batches.

With using heterogeneous gender mini-batch sample selection strategy, the aim is to train deep learning system with the features of both genders in one mini-batch. In other words, deep learning based speech recognition system would see the features of both female and male speakers and build a model based on mixed features of both genders. By mixing the genders in mini-batches, the effect of using both genders on training of deep learning based speech recognition systems at mini-batches could be evaluated.

3.3. Accent Adjusted Strategies

Accent adjusted mini-batch sample selection strategies use accent information of speech datasets to select and gather mini-batch samples. Accent is an essential information on speech recognition systems which would impact the performance of the systems. Thus, in this thesis, accent adjusted mini-batch sample selection strategies are proposed to select mini-batch samples based on accent information and to possibly increase deep learning based speech recognition performance.

In accent adjusted sample selection, the speech dataset is divided into the categories of accents of speakers and the speaker utterances are categorized based on their accents. Similar to gender adjusted strategies, two strategies are proposed in accent adjusted strategies using accent categories, i.e. homogeneous and heterogeneous accent strategies.

In homogeneous accent strategy, the mini-batches are filled with only one accent. Each accent has their proportional presence in training dataset in compatible proportion with the original dataset. For example, if English accent is 30% of the original dataset, then 30% of all mini-batches are filled with only English accent. Contrarily, in heterogeneous accent strategy, each mini-batch is filled with all accents that are present in the original dataset in compatible proportion with the original dataset, similar to homogeneous accent strategy. The samples of each mini-batch are selected randomly, and also the mini-batches are re-shuffled to prevent sequential ordering of speakers and wrongly learning of deep learning systems based on sequence of the input files would be prevented.

With using accent adjusted strategies, the aim is to obtain uniformly distributed accents on all mini-batches and to generate mini-batches that have only one accent or with the presence ratio of the accents. Algorithms 4 and 5 presents homogeneous and heterogeneous gender adjusted mini-batch sample selection strategies.

Algorithm 4. Homogeneous accent strategy

```
1.  initialization: Minibatch_samples = null
2.  all_accents = get-dataset-info(Dataset_Meta)
3.  all_accents_filenames = get-accent-files(all_accent, Dataset)
4.  shuffle(all_accents_filenames)
5.  all_accents_batch_counts=count-
    batches(all_accents,all_accents_filenames)
6.  for accent_batch_count in all_accents_batch_counts
7.    for j in accent_batch_count
8.      batch_samples = null
9.      batch_samples.add(all_accents_filenames[j, 0:(batch_size)])
10.     all_accents_filenames.remove[j, 0:(batch_size)]
11.     Minibatch_samples.add(batch_samples)
12.   end for
13. end for
14. return Minibatch_samples
```

Algorithm 4 presents homogeneous accent mini-batch sample selection strategy. In Algorithm 4, in step 1, *Minibatch_samples* array is initialized to fill with homogeneous accent strategy. In step 2, all accents in the dataset are read with get-dataset-info function. In step 3 the files of each accent are read using *all-accents* list. In step 5, batch count of each accent is calculated using count-batches-for-all-accents function. In this step, the presence ratio of accents in the original dataset is calculated and the number of batches for each accent is determined. In steps between 6 and 13, mini-batch samples are prepared using homogeneous accent strategy. In steps between 7 and 12, each accent is handled and batch samples for the accent are selected. In step 8, *batch_samples* array is initialized for filling the samples in a mini-batch. In step 9, *batch_samples* array is filled with file

names of selected accent. In step 10, selected file names are removed to prevent re-selection of the same file. In step 11, *batch_samples* array is added to the array of *Minibatch_samples*. The steps between 6 and 13 are processed for each accent with the number of *accent_batch_count* to correctly represent each accent. Finally, in step 14, *Minibatch_samples* array is returned as the output of homogeneous accent mini-batch sample selection algorithm.

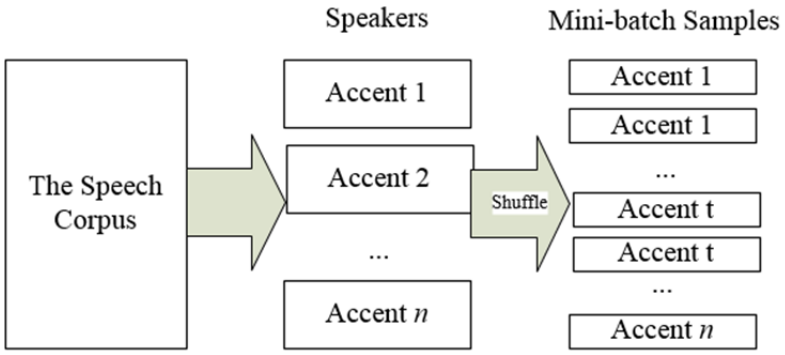

Figure 3.5. Homogeneous accent mini-batch sample selection strategy

Figure 3.5 presents homogeneous accent mini-batch sample selection strategy. As can be seen in the figure, first of all, the speech corpus is read from the memory. Then, the files of the speakers are separated based on their accents and each accent files are categorized. Finally, each mini-batch samples are selected from only one accent. Successive mini-batches are filled with the same accent until the accent presence ratio is finished. Each accent has its own accent presence ratio to fill a mini-batch. As the number of mini-batches are filled for the count of presence ratio, new accent is handled and the mini-batches are filled with the new accent. The process continues until all accents are used and all mini-batches are filled with only one accent. The mini-batch samples are selected randomly from accent files to prevent sequence and speaker dependency of deep learning systems.

The aim of homogeneous accent mini-batch sample selection strategy is to evaluate the performance of deep learning based speech recognition systems for one accent in one mini-batch and in a sequential order. With homogeneous accent strategy, the effect of unified accent on training of deep learning based speech recognition systems could be explored and studied whether it is a suitable method for mini-batch sample selection to improve the performance of speech recognition systems.

Algorithm 5 presents heterogeneous accent mini-batch sample selection strategy. Algorithm 5 is similar to Algorithm 4 in context of preparation of files and categorization of the accents. All accents are read from the dataset and file names of each accent is stored in *all_accents_filenames* matrix using *all_accents* array. In step 5, batch count is calculated using *num*_examples and *batch_size* parameters. The difference between Algorithm 5 and Algorithm 4 is at the selection of mini-batch samples at steps between 6 and 15. In step 7, *batch_samples* array is initialized for filling the samples in a mini-batch. Steps between 8 and 12 are used to select proportional accents for the mini-batches. In step 9, accent sample counts are calculated for each accent. In step 10, by using *accent_sample_count* variable, mini-batch samples are selected from the accent categories and added into *batch_samples* array. In step 11, selected and added files are removed from *all_accents_filenames* array to prevent re-selection of the same file. In step 13, *batch_samples* array is re-shuffled. In step 14, *batch_samples* array is added to the array of *Minibatch_samples*. Finally, in step 16, *Minibatch_samples* array is returned as the output of heterogeneous accent mini-batch sample selection algorithm.

Algorithm 5. Heterogeneous accent strategy

```
1.  initialization: Minibatch_samples = null
2.  all_accents = get-dataset-info(Dataset_Meta)
3.  all_accents_filenames = get-accent-files(all_accent, Dataset)
4.  shuffle(all_accents_filenames)
5.  batch_count = num_examples / batch_size
6.  for i in batch_count
7.    batch_samples = null
8.    for accent in all_accents
9.      accent_sample_count=calculate-accent-sample-count(accent,
      all_accents_filenames)
10.     batch_samples.add(all_accents_filenames[accent,  0:accent_sample-
      _count])
11.     all_accents_filenames.remove[accent, 0:accent_sample_count]
12.   end for
13.   shuffle(batch_samples)
14.   Minibatch_samples.add(batch_samples)
15. end for
16. return Minibatch_samples
```

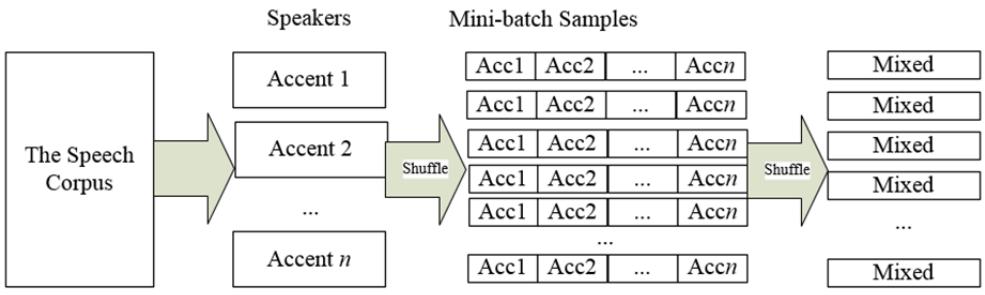

Figure 3.6. Heterogeneous accent mini-batch sample selection strategy

Figure 3.6 presents heterogeneous accent mini-batch sample selection strategy. As can be seen in the figure, reading the corpus into the memory and extraction and separation of different accents from the corpus are similar with homogeneous accent strategy. The difference of heterogeneous accent strategy is the selection of mini-batch samples from the separated accents lists. In heterogeneous accent mini-batch sample selection strategy, each mini-batch is filled with all accents that are present in the speech corpus with their presence ratio. When selecting files from each accent list, random selection is performed to prevent sequential selection and speaker dependent file selection. All accents are represented in one mini-batch. Finally, the selected files in mini-batches are further shuffled to mix sequence of each accent in each mini-batch to prevent wrong dependency modelling. The aim of heterogeneous accent strategy is to represent each accent in one mini-batch and to investigate the performance of deep learning based speech recognition on presence of all accents in one mini-batch.

With using heterogeneous accent mini-batch sample selection strategy, each mini-batch has all accents and the deep learning systems are trained with presence of all accents. The performance of deep learning based speech recognition systems could be evaluated when all accents are present at one mini-batch by using heterogeneous accent mini-batch sample selection strategy.

3.4. Hybrid Strategies

In this thesis study, hybrid strategies are also proposed which are generated by combined usage of gender and accent adjusted mini-batch sample selection strategies. The aim of hybrid mini-batch sample selection strategies is to investigate whether combining proposed gender and accent adjusted strategies would further improve accuracy and performance of deep learning based speech recognition.

The mini-batch sample selection approach in proposed hybrid strategies provide mixed and equally involved meta features. This could increase the

performance of deep learning based speech recognition in context of mini-batch samples which selects the samples and updates the weights based on these samples. Hybrid approaches use the main idea of gender and accent adjusted strategies and combines these ideas with each other. One mini-batch is divided into two parts and each part is filled with gender and accent adjusted strategies. Combination of each strategy would provide different results and could help increase performance of deep learning systems for speech recognition applications.

The proposed four hybrid mini-batch sample selection strategies combine successful gender and accent adjusted strategies and could provide better performance than using only one meta feature for mini-batch sample selection. Using both gender and accent information, more eligible mini-batch samples could be selected and mini-batches could be uniformly composed. This would provide presence of each information in all mini-batches and would increase the performance of deep learning systems.

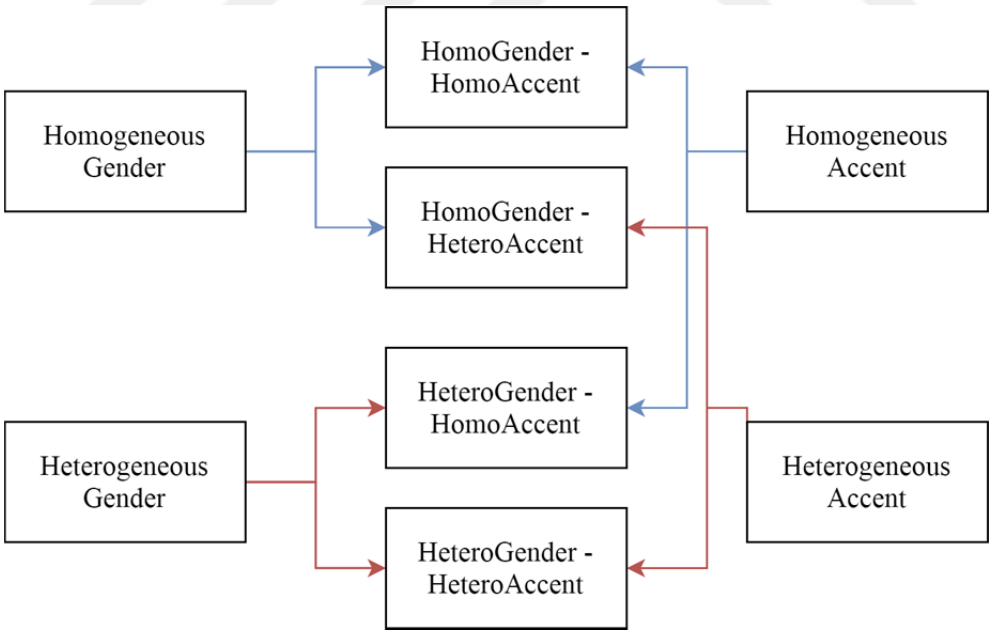

Figure 3.7. Proposed hybrid strategies

In hybrid sample selection strategies, gender and accent adjusted strategies that are explained in Section 3.2 and 3.3 are used in a hybrid manner. The main aim of hybrid strategies is to achieve better speech recognition performance and to further reduce error rates of proposed strategies. In particular, homogeneous gender, heterogeneous gender, homogeneous accent, and heterogeneous accent strategies are used to extract homogeneous gender – homogeneous accent, homogeneous gender – heterogeneous accent, heterogeneous gender – homogeneous accent, and heterogeneous gender – heterogeneous accent strategies. By this way, both gender and accent strategies are present in one mini-batch within an equal presence. The proposed hybrid strategies are presented in Figure 3.7.

As can be seen in Figure 3.7, the proposed gender and accent adjusted strategies are used to propose hybrid mini-batch sample selection strategies. In homogeneous gender – homogeneous accent hybrid strategy, only one gender and one accent are present in one mini-batch. In homogeneous gender – heterogeneous accent strategy, the mini-batches are filled with only one gender and with compatible proportion of accents. In heterogeneous-gender – homogeneous accent strategy, half of the mini-batches are filled with one accent and with male speaker and other half of the mini-batches are filled with same accent and with female speaker. In heterogeneous gender – heterogeneous accent strategy, both female and male speakers are equally present in the mini-batches and also compatible proportion of accents are selected for these genders.

3.4.1. Homogeneous Gender – Homogeneous Accent Strategy

Homogeneous gender – homogeneous accent hybrid strategy is constructed by using homogeneous gender and homogeneous accent mini-batch sample selection strategies which are explained in Section 3.2 and Section 3.3. In this hybrid mini-batch sample selection strategy, the speech samples are selected as homogeneous gender and homogeneous accent for one mini-batch. For example, female and English accent speakers are filled to one mini-batch and male and

English accent speakers are filled to the other mini-batch. The accents are distributed to the mini-batches in compatible with their presence ratio in the original dataset.

Homogeneous gender – homogeneous accent hybrid strategy would provide information and insights into the performance and success of deep learning based speech recognition systems when only one gender and only one accent is present in one mini-batch.

3.4.2. Homogeneous Gender – Heterogeneous Accent Strategy

Homogeneous gender – heterogeneous accent hybrid strategy is constructed by using combination of homogeneous gender and heterogeneous accent mini-batch sample selection strategies. In this hybrid mini-batch sample selection strategy, the mini-batch samples are selected as homogeneous gender and with the presence of all accents in one mini-batch. For example, for one mini-batch, only female speakers are used but all accents are selected which are uttered by female speakers. Gender is homogeneous, but all accents are represented in all mini-batches. The accent presence in mini-batches are selected as compatible with their presence in the original dataset.

With using homogeneous gender – heterogeneous accent mini-batch sample selection strategy, the main aim is to investigate the performance of deep learning based speech recognition system with using only one gender and with the presence of all accents whether it is more eligible strategy with respect to other sample selection strategies.

3.4.3. Heterogeneous Gender – Homogeneous Accent Strategy

Heterogeneous gender – homogeneous accent hybrid strategy is constructed by using heterogeneous gender and homogeneous accent mini-batch sample selection strategies. In this hybrid sample selection strategy, the accent is kept constant in mini-batches, however, the files of speakers are selected from each

gender equally. For example, one mini-batch is filled with English accent and half of the mini-batch is selected from English-accent male speakers and other half is selected from English-accent female speakers. The accents are distributed to the all mini-batches as compatible with their presence in the original dataset.

Heterogeneous gender – homogeneous accent mini-batch sample selection strategy is generated to investigate the performance of the speech recognition system with the presence of only one accent and with both genders in one mini-batch.

3.4.4. Heterogeneous Gender – Heterogeneous Accent Strategy

Heterogeneous gender – heterogeneous accent hybrid strategy is constructed by using heterogeneous gender and heterogeneous accent mini-batch sample selection strategies. In this hybrid strategy, both gender and accent information are selected in heterogeneous manner to fill mini-batches. For example, one mini-batch is filled with the utterances of both female and male speakers while accents of these genders are selected in compatible with their presence in the mini-batches.

Heterogeneous gender – heterogeneous accent hybrid mini-batch sample selection strategy is developed whether the performance of deep learning based speech recognition system could be increased with the presence of each meta feature on all mini-batches at the same time.

3.5. Overview of Proposed Mini-Batch Sample Selection Strategies

In this thesis study, three types of mini-batch sample selection strategies are proposed for investigation of training performances of deep learning based speech recognition systems. The first type of the strategies is gender adjusted mini-batch sample selection strategies. In gender adjusted strategies, mini-batch samples are selected based on gender of the speakers, i.e. female speakers and male speakers. In gender adjusted strategies, two sub-strategies are proposed, i.e. homogeneous gender, and heterogeneous gender. In homogeneous gender mini-batch sample

selection strategy, mini-batches are filled with only one gender. In heterogeneous gender mini-batch sample selection strategies, mini-batches are filled with equal count of both genders.

The second type of proposed strategies is accent adjusted mini-batch sample selection strategies. In accent adjusted strategies, mini-batch samples are selected based on their accent information. Also, in accent adjusted strategies, two sub-strategies are proposed, i.e. homogeneous accent, and heterogeneous accent. In homogeneous accent mini-batch sample selection strategy, mini-batches are filled with using only one accent. In heterogeneous accent mini-batch sample selection strategy, mini-batches are filled with the presence of all accents.

The third type of proposed strategies is hybrid mini-batch sample selection strategies. Four hybrid mini-batch sample selection strategies are proposed with the combinations of gender and accent adjusted strategies, i.e. homogeneous gender – homogeneous accent, homogeneous gender – heterogeneous accent, heterogeneous gender – homogeneous accent, and heterogeneous gender – heterogeneous accent. Hybrid mini-batch sample selection strategies use the concept of gender and accent adjusted strategies in compatible with their names.

The proposed three types of mini-batch sample selection strategies are developed to investigate the performance of deep learning based speech recognition systems and to monitor the effect of gender and accent information on training of deep learning systems. All of the proposed strategies are evaluated on the same deep learning system to compare with each other and to determine the best strategy.

4. RESULTS AND DISCUSSIONS

This section presents the results of proposed mini-batch sample selection strategies. First of all, the speech corpus that is used in this thesis study is presented. Second, the deep learning system configurations are explained. Third, the results of the proposed mini-batch sample selection strategies, i.e. gender adjusted, accent adjusted and hybrid strategies, are presented. Finally, a discussion of all investigated results is performed and comparison with the literature studies are presented. The experimental results are presented as follows:

- The effect of gender adjusted strategies on deep learning based speech recognition performance,
- The effect of accent adjusted strategies on deep learning based speech recognition performance,
- The effect of hybrid strategies on deep learning based speech recognition performance

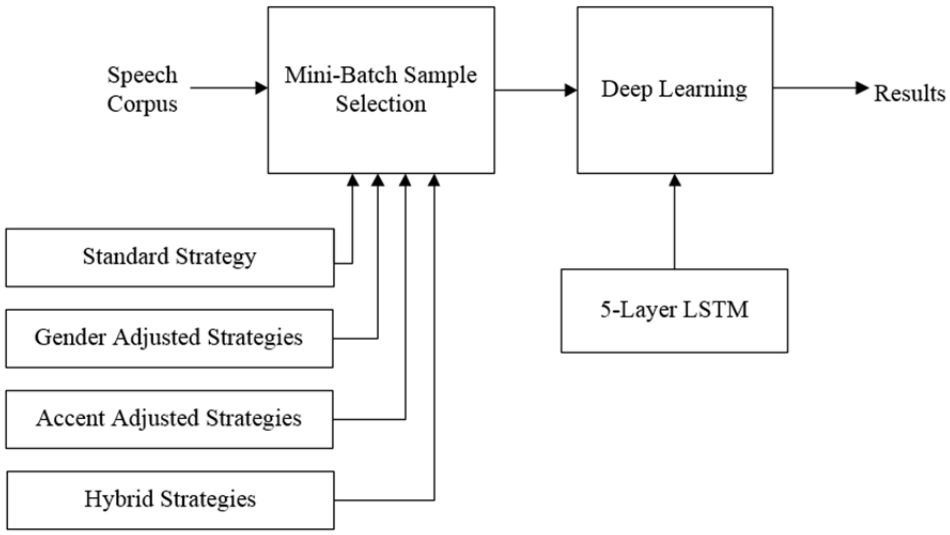

Figure 4.1. Basic architecture of proposed speech recognition system

Figure 4.1 presents the basic architecture of proposed speech recognition system of this thesis. As can be seen in the figure, first of all, the speech corpus is read into memory and then mini-batch sample selection strategies are processed on the speech corpus. In this step, four types of strategies are present, i.e. standard random selected mini-batch samples and proposed strategies. Finally, 5-layer LSTM deep learning system is run with selected mini-batch samples and the results are evaluated.

4.1. VCTK Speech Corpus

In this study, we used VCTK corpus as the speech dataset (Veaux et al., 2019). VCTK corpus is generated by University of Edinburgh. The Centre for Speech Technology Research (CSTR). Gender, age, accent, and region information are present in the meta files of the corpus that are suitable for our mini-batch sample selection strategies. The Corpus includes speech utterances read by 109 native speakers of English with different accents. Each speaker reads approximately 400 sentences. Most of the sentences were from a newspaper, plus the Rainbow Passage and an elicitation paragraph is present to identify the speaker's accent.

We selected VCTK Corpus because of its rich meta information about speakers. The distributions of number of speakers for different gender and accents are provided in Figures 4.2 and 4.3. As can be seen in the figures, each gender has reasonable proportion in the corpus. However, for some accents, there are many speakers but for some accents few speakers are present. For example, for English, American, and Scottish accents, there are many speakers present. But for Australian, New Zealand, and Welsh, few speakers are present. The reason for this distribution could be because of the preferences and less native speaker presence for these accents.

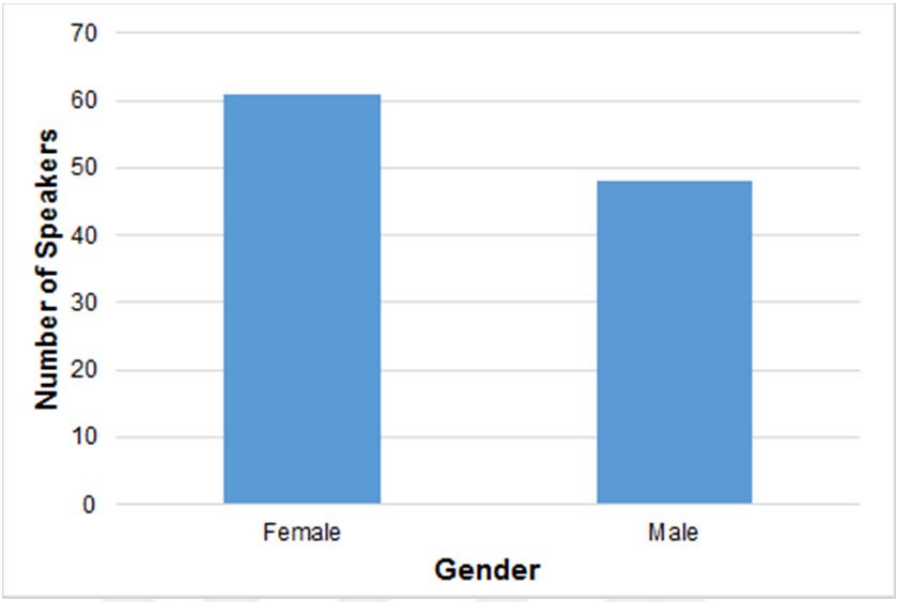

Figure 4.2. Distribution of the dataset based on gender information

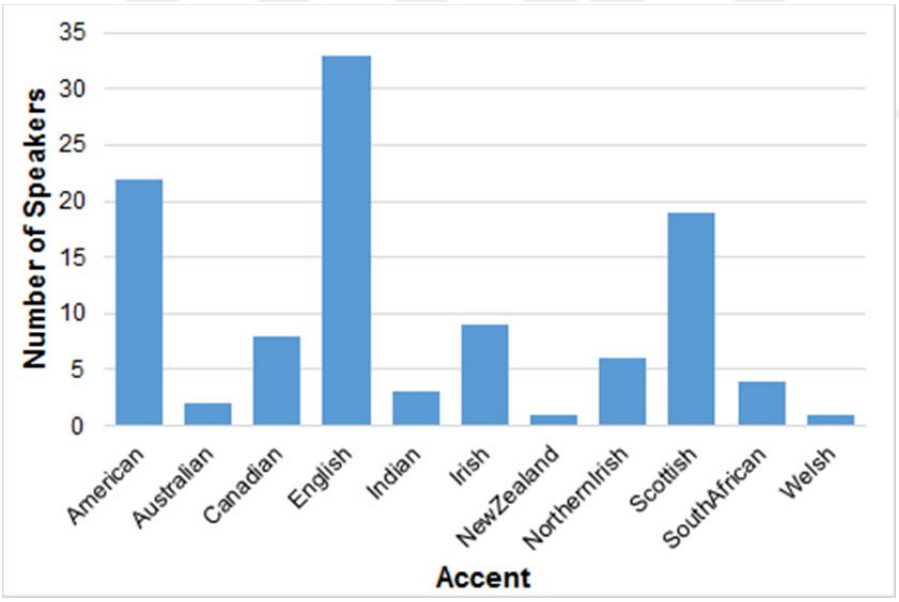

Figure 4.3. Distribution of the dataset based on accent information

We used quarter of the original corpus, that is around 11.000 sentences, by tuning *num_examples* parameter in the algorithms due to the insufficiency of our

computing infrastructures. We selected these sentences with respect to the sample selection strategy that is being used. The length of our selected corpus is around 12 hours of training dataset. Additionally, we randomly selected 1.024 sentences from all of the speakers that are used as test set. We removed test sentences from original corpus to test real performances of mini-batch sample selection strategies.

Some pre-processing steps are performed before deep learning system is applied on the corpus. First, all audio files are read from the corpus using a sample rate of 8000, and 13 MFCC features are extracted from the audio files. While extracting MFCC features, we used 25 milliseconds of window length and 10 milliseconds of window step. We removed all punctuation marks in the text files to better model uttered labels.

4.2. Deep Learning System Configurations

In this study, we used TensorFlow deep learning framework (Abadi et al., 2016) which is developed by Google Brain. TensorFlow is one of the popular deep learning frameworks which has many built-in deep learning architectures and provides several functionalities for developers. Tensorflow is based on Python programming language which is one of the most popular programming languages. It is shared as an open-source library for numerical computations and machine learning tasks, including deep learning systems. Tensorflow provides data flow graphs that provides information about how the data will flow on the graph on processing units.

TensorFlow is widely preferred for deep learning based speech recognition studies due to its several benefits. With the powerful and flexible mathematical ability of Python and the implementation of deep learning architectures, speech recognition problems could be efficiently studied. Preprocessing and post-processing steps could be handled by Python libraries and the deep learning step is effectively performed by TensorFlow on Python computational infrastructure.

We selected LSTM deep learning architecture. Number of layers are selected as 5 and number of hidden units for each layer are selected as 600. Adam optimizer is used as an optimization algorithm. We selected 64 for mini-batch sizes for performance evaluation of mini-batch sample selection strategies. We used Connectionist Temporal Classification (CTC) loss function to calculate training and test costs and Label Error Rates (LERs). CTC calculates LER of a transcribed audio using edit distance of transcribed text and original text that takes into account insertion, deletion, and substitution of characters. The details of CTC and LER calculation are provided in Section 2.3.

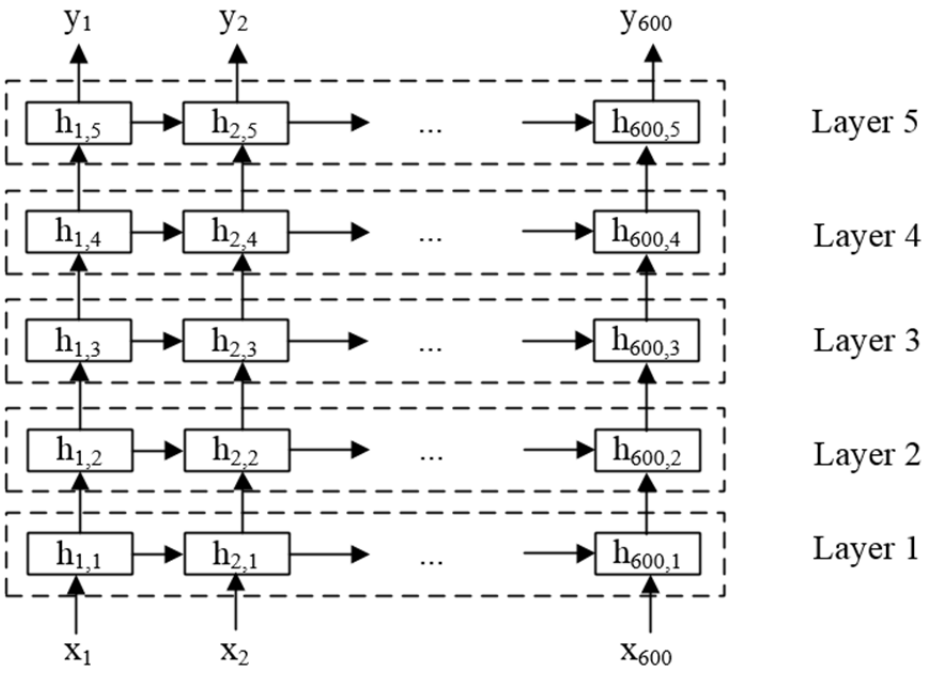

Figure 4.4. Illustration of the proposed deep learning system design

Figure 4.4 presents the detailed illustration of proposed deep learning system design for speech recognition. As can be seen in the figure, each rectangle represents one LSTM cell. Five layers are connected with each other for each time

step, and after the processing of five layers with the input x_t, the output y_t is produced for each time step. For extraction of current output, processing of 600 hidden units and 5 layers should be performed. With using multi-layer architecture, we aim to increase the performance of deep learning based speech recognition system and to better explore the differences of mini-batch sample selection strategies.

We used one 4 GB Nvidia GeForce GTX 1050 Ti graphical card with Intel i7 CPU and 16 GB of RAM as a computing infrastructure and all of the experiments are performed on this device.

4.3. Experimental Results

In this section, we evaluated performances of the proposed gender adjusted, accent adjusted and gender-accent hybrid mini-batch sample selection strategies in comparison with standard mini-batch sample selection that selects samples randomly from the dataset. The experiments are conducted to reveal performance comparison. In the experiments, training and test costs and Label Error Rates (LERs) of the strategies are compared. The details of calculation of LER is explained in Section 4.2. The experiments are repeated 10 times and average results of these experiments are presented with their error band in 95% confidence interval. The results of gender and accent adjusted strategies are presented, and then hybrid strategies are presented.

4.3.1. Results of Gender Adjusted Strategies

In this experiment we evaluated the effect of gender adjusted strategies, i.e. homogeneous gender and heterogeneous gender, with respect to standard mini-batch sample selection strategy. Batch size is selected as 64, Adam optimizer is selected, and learning rate of optimizer is selected as $1e^{-3}$. We ran deep learning systems for 1000 iterations and the results are presented. Figures 4.5 and 4.6 presents training cost and training LER of gender adjusted strategies and standard

strategy and Table 4.1 presents train and test costs and LERs of gender adjusted strategies after 1000 iterations.

As can be seen in Figures 4.5 and 4.6, learning rate of the strategies decayed around 500th iteration. As can be seen in the figures, both train cost and train LER follow the same trend for all of the strategies. Also, homogeneous gender strategy outperforms heterogeneous gender strategy and standard strategy after 1000 iterations in terms of train cost and LER. Heterogeneous gender strategy could not improve deep learning based speech recognition performance in comparison with standard randomly selected mini-batch sample selection strategy.

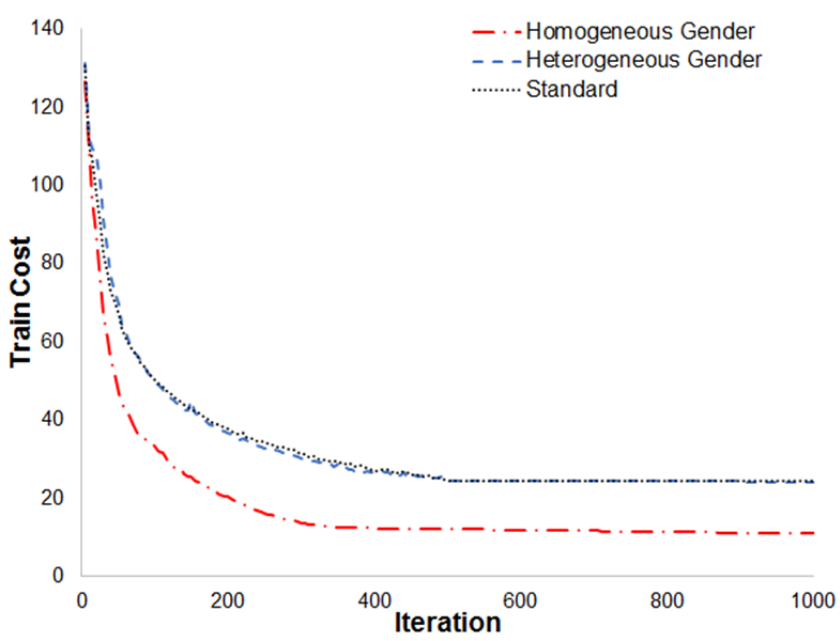

Figure 4.5. Train cost of gender adjusted strategies

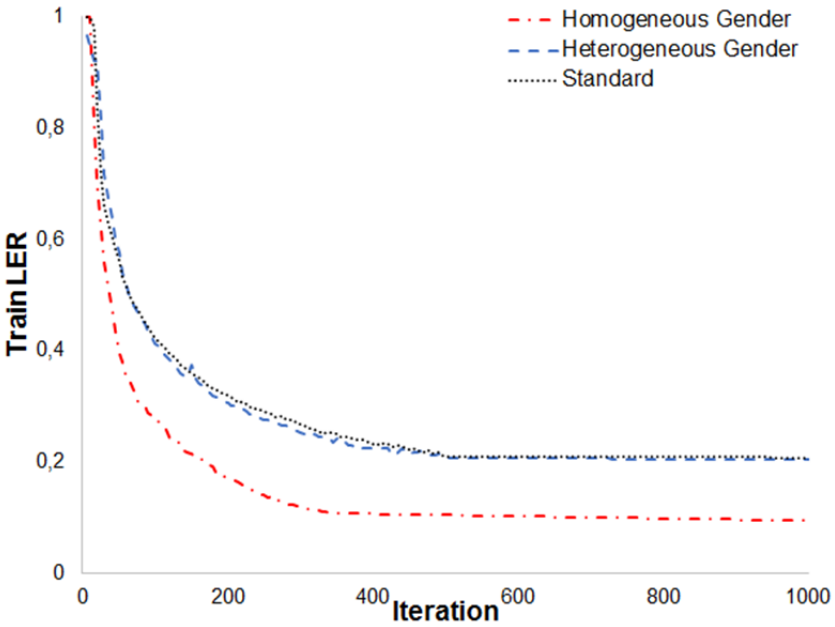

Figure 4.6. Train LER of gender adjusted strategies

Table 4.1. Train and test cost and LERs of gender adjusted strategies

	Standard	Homogeneous Gender	Heterogeneous Gender
Train Cost	24.31	11.04	24.22
Train LER	20.77%(±0.1)	9.77%(±0.05)	20.42%(±0.1)
Test Cost	32.42	20.73	30.05
Test LER	25.70%(±0.12)	15.32%(±0.08)	24.20%(±0.12)

(The difference between LERs is significant at 95% confidence interval (alpha=0.05))

As can be seen in Table 4.1, train and test costs of homogeneous gender strategy is much better than other alternatives. Homogeneous gender strategy could reduce train cost to 11.04 and test cost to 20.73 from 24.31, and 32.42, respectively in comparison with standard strategy. Also, train and test LERs of homogeneous

gender strategy outperform heterogeneous gender and standard strategies which could reduce to 9.77%, and 15.32%, respectively in comparison with other strategies. Overall, homogeneous gender strategy could reduce train and test LER around 10% more than other alternatives and the difference is statistically significant at 95% confidence interval.

The results of gender adjusted strategies show that selecting mini-batch samples randomly or with equal gender would not yield good performance. However, the selection of mini-batch samples with homogeneous gender strategy could improve deep learning based speech recognition performance with a high proportion. When we select one gender for one mini-batch, the deep learning model could learn better than providing equal number of samples for each mini-batch or selecting samples randomly. The reason for this is the utterances of words for male and female speakers are different and deep learning system could model this difference better. Selecting only one gender for one mini-batch provides closer gender features within a mini-batch and the deep learning based speech recognition systems could better learn the character utterances.

4.3.2. Results of Accent Adjusted Strategies

In this experiment, we evaluated the effect of accent adjusted strategies, i.e. homogeneous accent and heterogeneous accent, with respect to standard strategy. Batch size is selected as 64, Adam optimizer is selected, and learning rate of optimizer is selected as 1e-3. We ran deep learning systems for 1000 iterations and the results are presented. Figures 4.7 and 4.8 presents train cost and train LER of accent adjusted strategies and standard strategy and Table 4.2 presents train and test costs and LERs of accent adjusted strategies after 1000 iterations.

As can be seen in Figures 4.7 and 4.8, the learning rate of the strategies decayed at around 400th iteration. As can be seen in the figures, both train cost and train LER follow the same trend for all of the strategies. Also, heterogeneous accent strategy outperforms other two alternative strategies, homogeneous accent

strategy and standard strategy, after 1000 iterations in terms of both train cost and train LER. Homogeneous accent strategy could not improve deep learning based speech recognition performance in comparison with standard randomly selected mini-batch sample selection strategy.

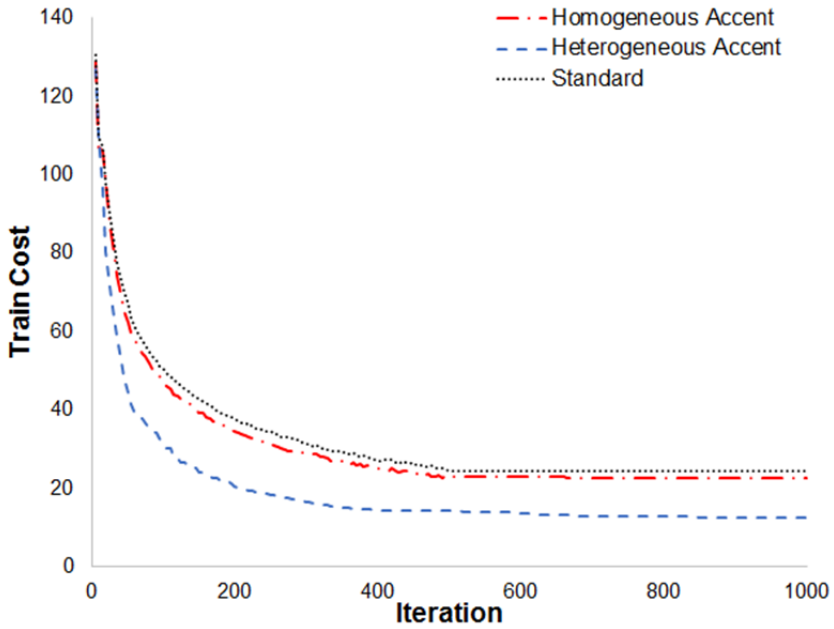

Figure 4.7. Train cost of accent adjusted strategies

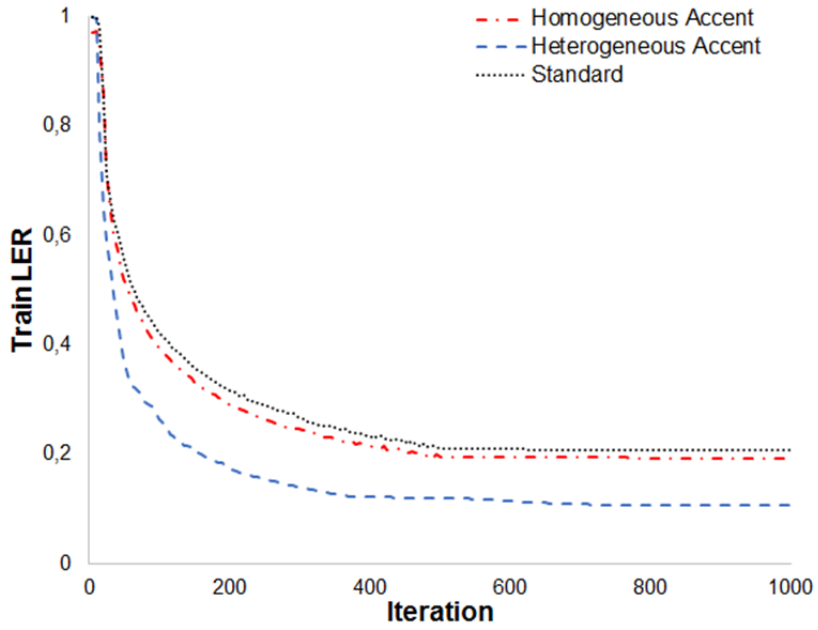

Figure 4.8. Train LER of accent adjusted strategies

Table 4.2. Train and test cost and LERs of accent adjusted strategies

	Standard	Homogeneous Accent	Heterogeneous Accent
Train Cost	24.31	22.67	12.68
Train LER	20.77%(±0.1)	19.28%(±0.09)	10.67%(±0.06)
Test Cost	32.42	32.15	21.95
Test LER	25.70%(±0.12)	25.56%(±0.12)	16.49%(±0.08)

(The difference between LERs is significant at 95% confidence interval (alpha=0.05))

As can be seen in Table 4.2, heterogeneous accent strategy is the best among these three strategies for both train and test costs. Heterogeneous accent strategy could reduce train and test cost to 12.68, and 21.95 from 24.31, and 32.42, respectively in comparison with standard strategy. Also, heterogeneous accent

strategy provides less train and test LERs with respect to other alternatives. It could reduce train and test LERs to 10.67% and 16.49%, respectively in comparison with homogeneous accent and standard strategies and the difference between LERs is statistically significant at 95% confidence interval. After these results are evaluated, heterogeneous accent strategy could reduce train and test costs more than 10% and train and test LERs up to 9%.

The results of accent adjusted strategies show that using only one accent for one mini-batch or selecting mini-batch samples randomly would not improve deep learning performance of speech recognition. However, representing each accent in a mini-batch proportional to their presence in the corpus provides better performance to the deep learning based speech recognition system. The reason for this is that, the deep learning systems could learn better and could achieve higher performance when all accents are present in one mini-batch. When all accents are present in one mini-batch, the deep learning system could learn all accents and could better model the speech signals. It could learn every possible accent and all varieties of the utterances of the characters within each mini-batch.

4.3.3. Results of Hybrid Strategies

In this experiment, we evaluated the effect of gender-accent hybrid strategies, i.e. homogeneous gender – homogeneous accent, homogeneous gender – heterogeneous accent, heterogeneous gender – homogeneous accent, and heterogeneous gender – heterogeneous accent, with respect to standard strategy. Batch size is selected as 64, and learning rate of optimizer is selected as 1e-3. We ran deep learning systems for 1000 iterations and the results are presented. Figures 4.9 and 4.10 presents train cost and train LER of gender-accent hybrid strategies and standard strategy and Table 4.3 presents train and test costs and LERs after 1000 iterations.

As can be seen in Figures 4.9 and 4.10, train costs and LERs follow the same trend for each hybrid strategy. The best strategy among all four candidate strategies

is homogeneous gender – heterogeneous accent strategy after 1000 iterations. Train cost is observed as 10.92 and train LER is 9.64% for homogeneous gender – heterogeneous accent hybrid strategy. Heterogeneous gender – heterogeneous accent strategy is closer in terms of train cost and LER. Worst performance is observed with heterogeneous gender – homogeneous accent strategy. These results show that heterogeneous accent is an important factor for mini-batch sample selection strategies. Also, homogeneous gender further improves the deep learning performance.

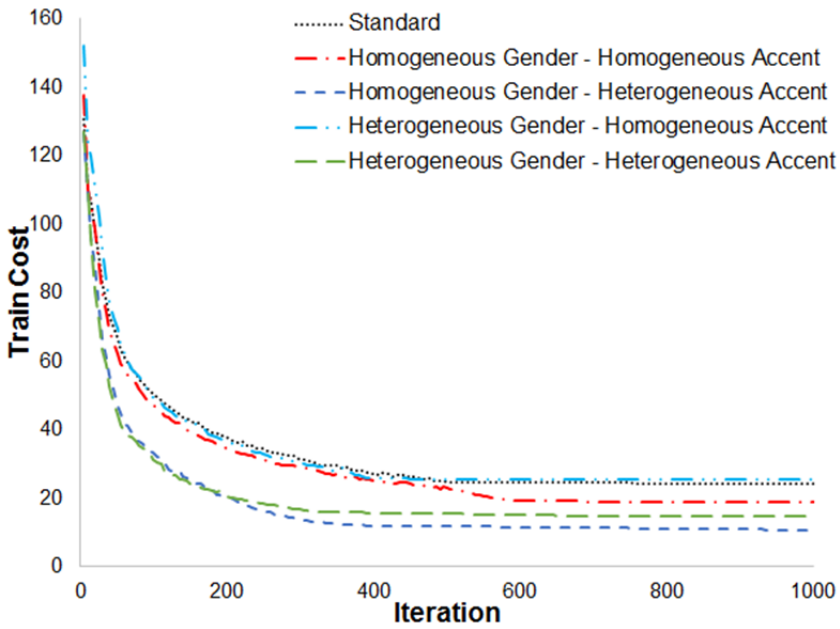

Figure 4.9. Train cost of gender-accent adjusted hybrid strategies

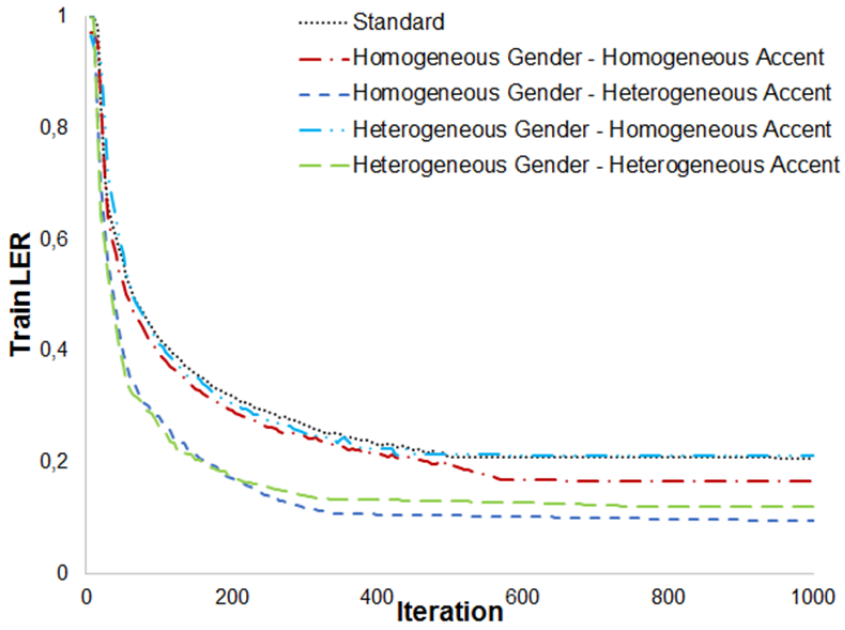

Figure 4.10. Train LER of gender-accent adjusted hybrid strategies

As can be seen in Table 4.3, homogeneous gender – heterogeneous accent hybrid strategy is the best strategy among all hybrid strategies and standard strategy in terms of train and test costs. Homogeneous gender – heterogeneous accent strategy has 10.92 and 20.52 train and test costs which is the least cost among all of the hybrid strategies. Also, homogeneous gender – heterogeneous accent strategy has the best train and test LER. The test LER is reduced from 25.70% to 15.21% with homogeneous gender – heterogeneous accent strategy and better speech recognition performance is achieved. Heterogeneous gender – heterogeneous accent hybrid strategy has close performance with homogeneous gender – heterogeneous accent strategy which has 19.05% test LER. The worst performance is observed for heterogeneous gender – homogeneous accent which is worse than standard strategy.

Table 4.3. Train and test cost and LERs of gender-accent adjusted hybrid strategies

	Train Cost	Train LER	Test Cost	Test LER
Standard	24.31	20.77%(±0.1)	32.42	25.70%(±0.12)
Homogeneous Gender – Homogeneous Accent	18.47	16.13%(±0.09)	29.76	22.81%(±0.11)
Homogeneous Gender – Heterogeneous Accent	**10.92**	**9.64%(±0.05)**	**20.52**	**15.21%(±0.08)**
Heterogeneous Gender - Homogeneous Accent	25.19	21.27%(±0.11)	32.83	26.46%(±0.13)
Heterogeneous Gender - Heterogeneous Accent	14.83	12.71%(±0.06)	24.02	19.05%(±0.1)

(The difference between LERs is significant at 95% confidence interval (alpha=0.05))

The results of hybrid strategies show that some of the proposed hybrid strategies achieve good performance, even better than gender and accent adjusted strategies, while some hybrid strategies show worse performance than standard random mini-batch sample selection strategy. When the results of hybrid strategies are evaluated, heterogeneous accent strategy has an importance on improvement of speech recognition performance. When heterogeneous accent is used in hybrid strategies, satisfying performances could be achieved. Another important result is that homogeneous gender further improves the performance of deep learning based speech recognition with combination of heterogeneous accent strategy. When homogeneous accent is selected for hybrid strategies, the performance of deep learning based speech recognition gets lower.

4.4. Discussions

In this section, the performances of all proposed mini-batch sample selection strategies are evaluated in comparison with each other. Figures 4.11 and 4.12 presents train and test costs and LERs of all proposed strategies. As can be seen in the figures, train costs and LERs are smaller than test costs and LERs because the deep learning systems are trained with train set and the systems does not meet test instances. For this reason, test scores of deep learning systems are higher than train scores. Costs and LERs of the strategies follow the same trend for each strategy which shows the stability of deep learning systems. When the cost of a strategy is higher, then LER is expected to be higher, and vice versa.

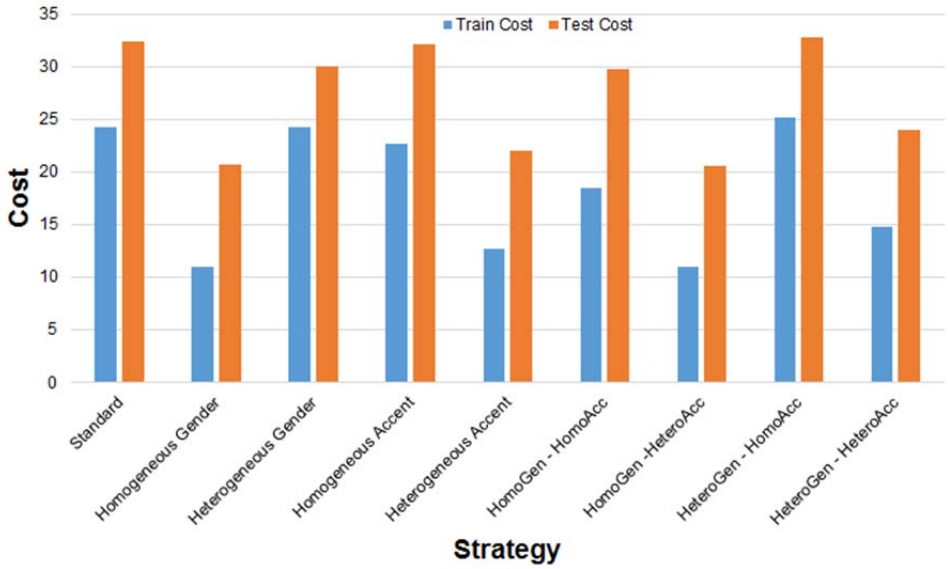

Figure 4.11. Train and test costs of all strategies

Figure 4.11 presents the train and test costs of all proposed mini-batch sample selection strategies and standard strategy. As can be seen in the figure, homogeneous gender – heterogeneous accent hybrid strategy and homogeneous gender strategy are the best strategies among all of the strategies in terms of train and test costs. Also, the worst strategies are standard strategy and heterogeneous

gender – homogeneous accent strategy. Other proposed strategies have test cost within the range of 20 and 30.

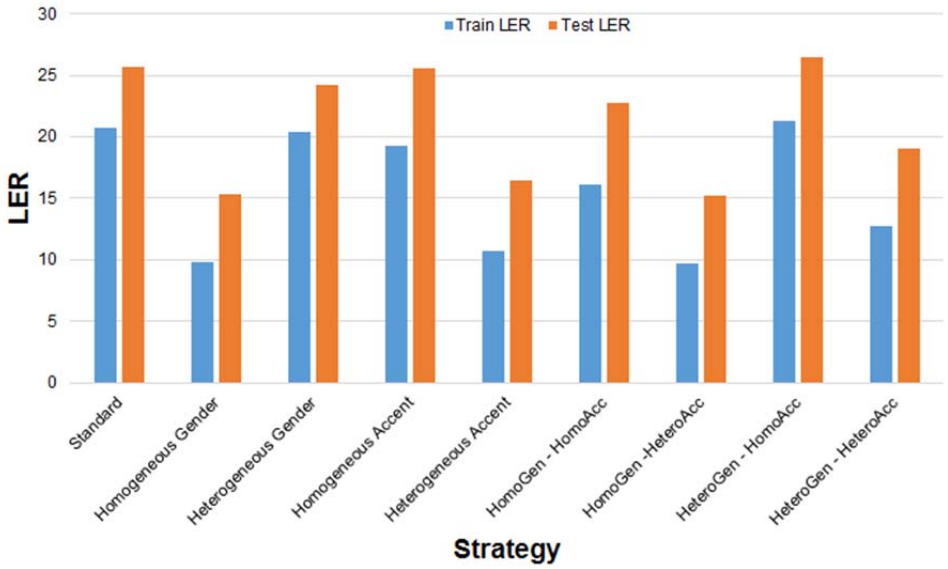

Figure 4.12. Train and test LERs of all strategies

Figure 4.12 presents the train and test LERs of all proposed mini-batch sample selection strategies and standard strategy. As can be seen in the figure, homogeneous gender – heterogeneous accent hybrid strategy and homogeneous gender strategy are the best strategies among all of the mini-batch sample selection strategies. The worst strategies are standard strategy and heterogeneous gender – homogeneous accent strategy which is compatible with train and test costs. Other strategies are within the range of 15%-25% test LER.

Table 4.4. Train and test cost and LERs and average runtime of one iteration of all proposed strategies

	Train Cost	Train LER	Test Cost	Test LER	Average Runtime (sec.)
Standard	24.31	20.77%	32.42	25.70%	484.12
Homogeneous Gender	11.04	9.77%	20.73	15.32%	481.27
Heterogeneous Gender	24.22	20.42%	30.05	24.20%	478.69
Homogeneous Accent	22.67	19.28%	32.15	25.56%	457.03
Heterogeneous Accent	12.68	10.67%	21.95	16.49%	471.83
Homogeneous Gender – Homogeneous Accent	18.47	16.13%	29.76	22.81%	475.25
Homogeneous Gender – Heterogeneous Accent	**10.92**	**9.64%**	**20.52**	**15.21%**	482.14
Heterogeneous Gender - Homogeneous Accent	25.19	21.27%	32.83	26.46%	469.54
Heterogeneous Gender - Heterogeneous Accent	14.83	12.71%	24.02	19.05%	477.85

Table 4.4 presents train and test costs and LERs and average runtime of one iteration of proposed mini-batch sample selection strategies and standard strategy. As can be seen in the table, best strategy is homogeneous gender – heterogeneous accent hybrid strategy in terms of train and test costs and LERs. Also, homogeneous gender strategy has train and test costs and LERs close to the best strategy. Third successful strategy is observed as heterogeneous accent strategy, which is the best accent adjusted strategy. These three strategies could be seen as

the best strategies and could be preferred for improving performance of deep learning based speech recognition.

After the results in the Table 4.4 are evaluated, we can conclude that homogeneous gender – heterogeneous accent hybrid strategy, homogeneous gender strategy and heterogeneous accent strategy outperform other alternative strategies in terms of train and test metrics. The reason for this result could be that, selecting homogeneous gender as one mini-batch samples provide deep learning systems to better model speech inputs with output texts. Also, when all accents are present in every mini-batch, the deep learning system becomes more successful, because it could learn every possible accent within each mini-batch. When all of the results are evaluated, we can conclude that, when gender and accent meta information are used for mini-batch sample selection, deep learning based speech recognition models could achieve better performance. Especially when using hybrid of successful mini-batch sample selection strategies further improves the performance of the systems.

Additionally, Table 4.4 presents average runtime of one iteration of proposed mini-batch sample selection strategies. As can be seen in the table, all of the strategies are close to each other in terms of average runtime in seconds. The runtime variation occurs because of the length of selected samples. However, there is no major runtime difference. 1000 iterations of a strategy last approximately 120-140 hours.

4.5. Comparison of the Results with the Literature Studies

Table 4.5 presents the performances of our proposed mini-batch sample selection strategies with respect to the literature. Also, in the table, the utilized deep learning system is presented to provide a fair comparison of the results. We used LER of selected literature studies that has no language model and that can be compared with our study because in this thesis study we did not consider using a language model for speech recognition system.

Table 4.5. Comparison with the literature

	Deep Learning System	Test LER
Standard	LSTM with 5 layers and 600 units	25.70%
Homogeneous Gender	LSTM with 5 layers and 600 units	15.32%
Heterogeneous Gender	LSTM with 5 layers and 600 units	24.20%
Homogeneous Accent	LSTM with 5 layers and 600 units	25.56%
Heterogeneous Accent	LSTM with 5 layers and 600 units	16.49%
Homogeneous Gender – Homogeneous Accent	LSTM with 5 layers and 600 units	22.81%
Homogeneous Gender – Heterogeneous Accent	LSTM with 5 layers and 600 units	**15.21%**
Heterogeneous Gender - Homogeneous Accent	LSTM with 5 layers and 600 units	26.46%
Heterogeneous Gender - Heterogeneous Accent	LSTM with 5 layers and 600 units	19.05%
CTC (Graves et al., 2006)	BiLSTM with 2 layers and 100 units	31.47%
RNN-CTC without language model (Graves, Jaitly, 2014)	LSTM with 5 layers and 500 units	30.9%
ChiME-CTC result (Watanabe et al., 2017)	BiLSTM with 4 layers and 320 units	48.79%
CTC no LM (Maas et al., 2015)	BiRNN with 5 layers and 1824 units	27.7%

As can be seen in the table, some of our proposed models have better test LER with respect to the baseline models that are similar to our study. Especially, homogeneous gender – heterogeneous accent and heterogeneous gender – heterogeneous accent hybrid strategies, homogeneous gender and heterogeneous accent mini-batch sample selection strategies are better strategies with respect to the literature. However, some other proposed strategies fall behind the literature, i.e. standard strategy and heterogeneous gender – homogeneous accent hybrid strategy.

When the literature results are evaluated in terms of utilized deep learning system, if a deeper neural network is used, there is more chance to increase performance of the speech recognition systems and decrease LER.

5. CONCLUSION AND FUTURE WORK

Speech recognition is the process of converting speech signals into text transcription. Speech recognition is essential for converting voice records, controlling devices with speech inputs and for searching in speech records. With the increasing demand for speech recognition and with the help of computational advancements, speech recognition became more practical and accessible for every person that have commodity devices which do not require advanced processing ability. Speech recognition is widely used in many technologies, including mobile communication, search engines, and personal digital assistance systems.

On the other hand, deep learning is a sub-field of machine learning that uses a set of algorithms that attempt to model high-level abstractions in data by using a deep graph with multiple processing layers, composed of multiple linear and non-linear transformations (Goodfellow *et al.*, 2016; Deng, Yu, 2014). Deep learning provides automatic selection and ranking of features in the datasets using efficient algorithms. Recently, deep learning achieved great attention and tremendous success in application areas, such as image processing, natural language processing, sequence alignment, and speech recognition.

With the popularity and high performance of deep learning systems with respect to other candidates, these systems are also being utilized in speech recognition problems. Among the most popular deep learning architectures, Deep Neural Networks (DNN), Convolutional Neural Networks(CNN), and especially Recurrent Neural Networks(RNN) and Long Short Term Memory (LSTM) RNNs, which is a special kind of RNN, are preferred for speech recognition. CNN is mostly utilized for visual speech inputs, which is called as *spectrograms*, while RNN and LSTM are utilized for time series feature inputs of speech signals, such as MFCCs.

Deep learning based speech recognition systems are successful systems for recognizing a speech input with respect to the input features. The performance of

the deep learning based speech recognition systems is based on several factors, including a comprehensive speech corpus which contains the different characteristics of user types, such as gender, accent, speaker and linguistic contents. In order to increase the recognition performance of these systems is required to have a more reliable and accurate speech recognition systems. One important factor to increase the performances of deep learning based speech recognition systems is to train these systems with more data that have a wide range of accent, gender, and speakers.

In this thesis study, we aim to improve performance of deep learning based speech recognition systems by proposing new methods for mini-batch sample selection which is an important part in training deep learning systems. We proposed several strategies for selecting mini-batch samples for speech recognition tasks to increase model performance of deep learning based speech recognition. In particular, we proposed three types of mini-batch sample selection strategies, i.e. gender adjusted strategies, accent adjusted strategies, and hybrid strategies which selects mini-batch samples using combination of gender and accent adjusted proposed strategies.

Proposed gender and accent adjusted strategies are divided into two further categories, i.e. homogeneous and heterogeneous. The homogeneous and heterogeneous categories show the selection of each meta feature with respect to the speech corpus. In homogeneous strategies, only one meta feature, i.e. only female speaker or only English accent, is used for one mini-batch. In heterogeneous strategies, proportional meta features, i.e. half female and half male speakers, are used for one mini-batch. Thus, each proposed gender and accent adjusted strategies contain two separate strategies, i.e. homogeneous gender, heterogeneous gender, homogeneous accent, and heterogeneous accent strategies.

Proposed hybrid strategies are combined usage of gender and accent adjusted strategies. Four strategies are proposed in this context, i.e. homogeneous gender – homogeneous accent, homogeneous gender – heterogeneous accent,

heterogeneous gender – homogeneous accent, and heterogeneous gender – heterogeneous accent. Hybrid mini-batch sample selection strategies use the concept of gender and accent adjusted strategies in compatible with their names in hybridized manner. For example, homogeneous gender – homogeneous accent hybrid strategy selects only one gender and only one accent for one mini-batch.

The proposed three types of mini-batch sample selection strategies are developed to investigate the performance of deep learning based speech recognition systems and to monitor the effect of gender and accent information on training of deep learning systems. All of the proposed strategies are evaluated on the same deep learning system to compare with each other and to determine the best strategy among all proposed strategies which use gender and accent information for mini-batch sample selection.

The experimental results of the proposed mini-batch sample selection strategies show that, homogeneous gender – heterogeneous accent hybrid strategy, homogeneous gender strategy and heterogeneous accent strategy outperform other alternative strategies in terms of train and test metrics. The reason for this result could be that, selecting homogeneous gender as one mini-batch samples provide deep learning systems to better model speech inputs with output texts. Also, when all accents are present in every mini-batch, the deep learning system becomes more successful, because it could learn every possible accent within each mini-batch. When all of the results are evaluated, we can conclude that, when gender and accent meta information are used for mini-batch sample selection, deep learning based speech recognition models could achieve better performance. Especially when using hybrid of successful mini-batch sample selection strategies further improves the performance of the systems.

As a result, we can conclude that, when gender and accent meta information are used for mini-batch sample selection, deep learning based speech recognition models could achieve better performance. Especially when using hybrid of

successful mini-batch sample selection strategies further improves the performance of the systems.

For the future, a deep learning system which has more layers and more units could be used to evaluate performances of mini-batch sample selection strategies in comparison with each other. New features could be evaluated for mini-batch sample selection strategies, such as age, and speaker. Also, other than VCTK speech corpus, new speech corpuses could be evaluated to further analyze the performance of proposed mini-batch sample selection strategies. Proposed mini-batch sample selection strategies could be adapted to other deep learning application areas, such as image recognition and machine vision based on features of these domains.

REFERENCES

Abdel-Hamid, O., Mohamed, A. R., Jiang, H., Deng, L., Penn, G., and Yu, D., 2014. Convolutional neural networks for speech recognition. *IEEE/ACM Transactions on audio, speech, and language processing*, *22*(10), 1533-1545.

Aggarwal, R. K., and Dave, M., 2011. Acoustic modeling problem for automatic speech recognition system: conventional methods (Part I). *International Journal of Speech Technology*, *14*(4), 297–308.

Aubert, X. L., 2002. An overview of decoding techniques for large vocabulary continuous speech recognition. *Computer Speech & Language*, *16*(1), 89-114.

Badshah, A. M., Ahmad, J., Rahim, N., and Baik, S. W., 2017. Speech emotion recognition from spectrograms with deep convolutional neural network. In *2017 international conference on platform technology and service (PlatCon)* (pp. 1-5). IEEE.

Benba, A., Jilbab, A., and Hammouch, A., 2015. Detecting patients with Parkinson's disease using Mel frequency cepstral coefficients and support vector machines. *International Journal on Electrical Engineering and Informatics*, *7*(2), 313-323.

Chan, W., Ke, N. R., and Lane, I., 2015. Transferring knowledge from a RNN to a DNN. *arXiv preprint arXiv:1504.01483*.

Chen, M., He, X., Yang, J., & Zhang, H., 2018. 3-D convolutional recurrent neural networks with attention model for speech emotion recognition. *IEEE Signal Processing Letters*, *25*(10), 1440-1444.

Cho, K., van Merriënboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H., and Bengio, Y., 2014. Learning Phrase Representations using RNN Encoder–Decoder for Statistical Machine Translation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)* (pp. 1724-1734).

Dahl, G. E., Yu, D., Deng, L., and Acero, A., 2011. Context-dependent pre-trained deep neural networks for large-vocabulary speech recognition. *IEEE Transactions on audio, speech, and language processing*, *20*(1), 30-42.

Dahl, G. E., Sainath, T. N., and Hinton, G. E., 2013. Improving deep neural networks for LVCSR using rectified linear units and dropout. In *2013 IEEE international conference on acoustics, speech and signal processing* (pp. 8609-8613). IEEE.

Deng, L., and Yu, D., 2014. Deep learning: methods and applications. *Foundations and trends in signal processing*, *7*(3–4), 197-387.

Doetsch P., Golik P., and Ney H., 2017. A comprehensive study of batch construction strategies for recurrent neural networks in mxnet. *arXiv preprint arXiv:1705.02414*.

Fu, S. W., Tsao, Y., and Lu, X., 2016. SNR-Aware Convolutional Neural Network Modeling for Speech Enhancement. In *INTERSPEECH 2016*, San Francisco, USA (pp. 3768-3772).

Gales, M., and Young, S., 2008. The application of hidden Markov models in speech recognition. *Now Publishers Inc*.

Gavat, I., and Militaru, D., 2015. Deep learning in acoustic modeling for automatic speech recognition and understanding-an overview. In *2015 International Conference on Speech Technology and Human-Computer Dialogue (SpeD)* (pp. 1-8). IEEE.

Gers, F. A., Schraudolph, N. N., and Schmidhuber, J., 2002. Learning precise timing with LSTM recurrent networks. *Journal of machine learning research*, *3*, 115-143.

Graves, A., Fernández, S., Gomez, F., & Schmidhuber, J., 2006. Connectionist temporal classification: labelling unsegmented sequence data with recurrent neural networks. In *Proceedings of the 23rd international conference on Machine learning* (pp. 369-376).

Graves, A., 2012. Connectionist temporal classification. In *Supervised Sequence Labelling with Recurrent Neural Networks* (pp. 61-93). Springer, Berlin, Heidelberg.

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

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

Graves, A., and Jaitly, N., 2014. Towards end-to-end speech recognition with recurrent neural networks. In *International conference on machine learning*, Beijing, China (pp. 1764-1772).

Han, K., Yu, D., and Tashev, I., 2014. Speech emotion recognition using deep neural network and extreme learning machine. In *Fifteenth annual conference of the international speech communication association INTERSPEECH 2014*, (pp. 223-227), Singapore.

Hori, T., Watanabe, S., Zhang, Y., and Chan, W., 2017. Advances in joint CTC-attention based end-to-end speech recognition with a deep CNN encoder and RNN-LM. *arXiv preprint arXiv:1706.02737*.

Hughes, T., and Mierle, K., 2013. Recurrent neural networks for voice activity detection. In *2013 IEEE International Conference on Acoustics, Speech and Signal Processing* (pp. 7378-7382). IEEE.

Jaitly, N., Nguyen, P., Senior, A., and Vanhoucke, V., 2012. Application of Pretrained Deep Neural Networks to Large Vocabulary Speech Recognition. In *Thirteenth Annual Conference of the International Speech Communication Association INTERSPEECH 2012*, (pp. 1-4), Portland, OR, USA.

Krause, B., Murray, I., Renals, S., and Lu, L., 2016. Multiplicative LSTM for sequence modelling. In *5th International Conference on Learning Representations*, (p. 1-9).

Lalitha, S., Tripathi, S., and Gupta, D., 2019. Enhanced speech emotion detection using deep neural networks. *International Journal of Speech Technology*, *22*(3), 497-510.

Lei, Y., Scheffer, N., Ferrer, L., and McLaren, M., 2014. A novel scheme for speaker recognition using a phonetically-aware deep neural network. In *2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)* (pp. 1695-1699). IEEE.

Li, X., and Wu, X., 2015. Constructing long short-term memory based deep recurrent neural networks for large vocabulary speech recognition. In *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)* (pp. 4520-4524). IEEE.

Lim, W., Jang, D., and Lee, T., 2016. Speech emotion recognition using convolutional and recurrent neural networks. In *2016 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA)* (pp. 1-4). IEEE.

Loshchilov I., and Hutter F., 2015, Online batch selection for faster training of neural networks, *arXiv preprint arXiv:1511.06343.*

Lu, L., Zhang, X., and Renais, S., 2016. On training the recurrent neural network encoder-decoder for large vocabulary end-to-end speech recognition. In *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)* (pp. 5060-5064). IEEE.

Maas, A., Le, Q. V., O'neil, T. M., Vinyals, O., Nguyen, P., and Ng, A. Y., 2012. Recurrent neural networks for noise reduction in robust ASR. *INTERSPEECH 2012*, (pp. 1-4), Portland, OR, USA.

Mao, Q., Dong, M., Huang, Z., and Zhan, Y., 2014. Learning salient features for speech emotion recognition using convolutional neural networks. *IEEE transactions on multimedia*, *16*(8), 2203-2213.

Miao, Y., Gowayyed, M., and Metze, F., 2015. EESEN: End-to-end speech recognition using deep RNN models and WFST-based decoding. In *2015 IEEE Workshop on Automatic Speech Recognition and Understanding (ASRU)* (pp. 167-174). IEEE.

Mirsamadi, S., Barsoum, E., and Zhang, C., 2017. Automatic speech emotion recognition using recurrent neural networks with local attention. In *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)* (pp. 2227-2231). IEEE.

Narang, S., and Gupta, M. D., 2015. Speech feature extraction techniques: a review. *International Journal of Computer Science and Mobile Computing*, *4*(3), 107-114.

Ou, C., Yang, J., Du, Z., Zhang, X., and Zhu, D., 2019. Integrating Cellular Automata with Unsupervised Deep-Learning Algorithms: A Case Study of Urban-Sprawl Simulation in the Jingjintang Urban Agglomeration, China. *Sustainability*, *11*(9), 2464.

Palaz, D., Doss, M. M., and Collobert, R., 2015. Convolutional neural networks-based continuous speech recognition using raw speech signal. In *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)* (pp. 4295-4299). IEEE.

Park J.S., Kim H.G., Kim D.G., Yu I.J., and Lee H.K., 2018. Paired mini-batch training: A new deep network training for image forensics and steganalysis, *Signal Processing: Image Communication*, *67*, 132-139.

Peng X., Li L., and Wang F.Y., 2019. Accelerating minibatch stochastic gradient descent using typicality sampling. *IEEE Transactions On Neural Networks and Learning Systems* (Early Access).

Permanasari, Y., Harahap, E. H., and Ali, E. P., 2019. Speech recognition using Dynamic Time Warping (DTW). In *Journal of Physics: Conference Series* (Vol. 1366, No. 1, p. 012091). IOP Publishing.

Qian, Y., Bi, M., Tan, T., and Yu, K., 2016. Very deep convolutional neural networks for noise robust speech recognition. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, *24*(12), 2263-2276.

Sainath, T. N., Mohamed, A. R., Kingsbury, B., and Ramabhadran, B., 2013. Deep convolutional neural networks for LVCSR. In *2013 IEEE international conference on acoustics, speech and signal processing* (pp. 8614-8618). IEEE.

Sainath, T. N., Vinyals, O., Senior, A., and Sak, H., 2015. Convolutional, long short-term memory, fully connected deep neural networks. In *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)* (pp. 4580-4584). IEEE.

Sak, H., Senior, A., Rao, K., and Beaufays, F., 2015. Fast and accurate recurrent neural network acoustic models for speech recognition. *arXiv preprint arXiv:1507.06947*.

Sak, H., Senior, A., and Beaufays, F., 2014. Long short-term memory based recurrent neural network architectures for large vocabulary speech recognition. INTERSPEECH 2014, (pp. 1-5). Singapore.

Salomons, E. L., and Havinga, P. J., 2015. A survey on the feasibility of sound classification on wireless sensor nodes. *Sensors*, *15*(4), 7462-7498.

Seltzer, M. L., Yu, D., and Wang, Y., 2013. An investigation of deep neural networks for noise robust speech recognition. In *2013 IEEE international conference on acoustics, speech and signal processing* (pp. 7398-7402). IEEE.

Sercu, T., Puhrsch, C., Kingsbury, B., and LeCun, Y., 2016. Very deep multilingual convolutional neural networks for LVCSR. In *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)* (pp. 4955-4959). IEEE.

Snyder, D., Ghahremani, P., Povey, D., Garcia-Romero, D., Carmiel, Y., and Khudanpur, S., 2016. Deep neural network-based speaker embeddings for end-to-end speaker verification. In *2016 IEEE Spoken Language Technology Workshop (SLT)* (pp. 165-170). IEEE.

Sun, L., Kang, S., Li, K., and Meng, H., 2015. Voice conversion using deep bidirectional long short-term memory based recurrent neural networks. In *2015 IEEE international conference on acoustics, speech and signal processing (ICASSP)* (pp. 4869-4873). IEEE.

Swietojanski, P., Ghoshal, A., and Renals, S., 2014. Convolutional neural networks for distant speech recognition. *IEEE Signal Processing Letters*, *21*(9), 1120-1124.

Testolin, A., Piccolini, M., and Suweis, S., 2020. Deep learning systems as complex networks. *Journal of Complex Networks*, *8*(1), cnz018.

Thomas, S., Ganapathy, S., Saon, G., and Soltau, H., 2014. Analyzing convolutional neural networks for speech activity detection in mismatched acoustic conditions. In *2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)* (pp. 2519-2523). IEEE.

Torfi, A., Dawson, J., and Nasrabadi, N. M., 2018. Text-independent speaker verification using 3d convolutional neural networks. In *2018 IEEE International Conference on Multimedia and Expo (ICME)* (pp. 1-6). IEEE.

Trigeorgis, G., Ringeval, F., Brueckner, R., Marchi, E., Nicolaou, M. A., Schuller, B., and Zafeiriou, S., 2016. Adieu features? end-to-end speech emotion recognition using a deep convolutional recurrent network. In *2016 IEEE international conference on acoustics, speech and signal processing (ICASSP)* (pp. 5200-5204). IEEE.

Variani, E., Lei, X., McDermott, E., Moreno, I. L., and Gonzalez-Dominguez, J., 2014. Deep neural networks for small footprint text-dependent speaker verification. In *2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)* (pp. 4052-4056). IEEE.

Wang, D., Wang, X., and Lv, S., 2019. End-to-End Mandarin Speech Recognition Combining CNN and BLSTM. *Symmetry*, *11*(5), 644.

Watanabe, S., Hori, T., Kim, S., Hershey, J. R., and Hayashi, T., 2017. Hybrid CTC/attention architecture for end-to-end speech recognition. *IEEE Journal of Selected Topics in Signal Processing*, *11*(8), 1240-1253.

Weninger, F., Erdogan, H., Watanabe, S., Vincent, E., Le Roux, J., Hershey, J. R., and Schuller, B., 2015. Speech enhancement with LSTM recurrent neural networks and its application to noise-robust ASR. In *International Conference on Latent Variable Analysis and Signal Separation* (pp. 91-99). Springer, Cham.

Weninger, F., Hershey, J. R., Le Roux, J., and Schuller, B., 2014. Discriminatively trained recurrent neural networks for single-channel speech separation. In *2014 IEEE Global Conference on Signal and Information Processing (GlobalSIP)* (pp. 577-581). IEEE.

Wu, Z., Sivadas, S., Tan, Y. K., Bin, M., and Goh, R. S. M., 2016. Multi-modal hybrid deep neural network for speech enhancement. *arXiv preprint arXiv:1606.04750*.

Yu, D., Yao, K., Su, H., Li, G., and Seide, F., 2013. KL-divergence regularized deep neural network adaptation for improved large vocabulary speech recognition. In *2013 IEEE International Conference on Acoustics, Speech and Signal Processing* (pp. 7893-7897). IEEE.

Yu, D., and Deng, L., 2016. Automatic Speech Recognition A Deep Learning Approach. Springer, 347 p.

Yu, D., and Li, J., 2017. Recent progresses in deep learning based acoustic models. *IEEE/CAA Journal of automatica sinica*, *4*(3), 396-409.

Zhang, Y., Pezeshki, M., Brakel, P., Zhang, S., Bengio, C. L. Y., and Courville, A., 2017. Towards end-to-end speech recognition with deep convolutional neural networks. *arXiv preprint arXiv:1701.02720*.

Zhang C., Kjellström H., and Mandt S., 2017. Determinantal point processes for mini-batch diversification, *arXiv preprint arXiv:1705.00607*.

Zhang C., Öztireli C., Mandt S., and Salvi G., 2018. Active Mini-Batch Sampling using Repulsive Point Processes. In *Proceedings of the AAAI Conference on Artificial Intelligence* (Vol. 33, pp. 5741-5748).

Zhao, H., Zarar, S., Tashev, I., and Lee, C. H., 2018. Convolutional-recurrent neural networks for speech enhancement. In *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)* (pp. 2401-2405). IEEE.

Zhu, J., Gong, X., and Chen, G., 2016. Deep Learning Based Language Modeling for Domain-Specific Speech Recognition. In *International Forum of Digital TV and Wireless Multimedia Communication* (pp. 242-251). Springer, Singapore.

Zen, H., Senior, A., and Schuster, M., 2013. Statistical parametric speech synthesis using deep neural networks. In 2013 *IEEE international conference on acoustics, speech and signal processing* (pp. 7962-7966). IEEE.

Zen, H., and Sak, H., 2015. Unidirectional long short-term memory recurrent neural network with recurrent output layer for low-latency speech synthesis. In *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)* (pp. 4470-4474). IEEE.

CURRICULUM VITAE

Yeşim DOKUZ was born in Niğde in 1987. She received her BSc. from Department of Computer Engineering in Harran University in 2011, and MSc. From Department of Computer Engineering in Erciyes University in 2014, respectively. She has been working as a research assistant in Department of Computer Engineering in Niğde Ömer Halisdemir University since 2012. Her research areas are speech recognition, deep learning and machine learning applications.