Avrupa Bilim ve Teknoloji Dergisi
Özel Sayı, S. 161-168, Nisan 2020

Araştırma Makalesi

www.ejosat.com ISSN:2148-2683

European Journal of Science and Technology
Special Issue, pp. 161-168, April 2020

Research Article

LSTM Hiperparametrelerinin Ses Tanıma Performansına olan Etkilerinin Araştırılması*

Yeşim Dokuz [1 †], Zekeriya Tüfekçi [2]

[1] Computer Engineering Department, Nigde Omer Halisdemir University, Nigde, Turkey (ORCID: 0000-0001-7202-2899)
[2] Computer Engineering Department, Cukurova University, Adana, Turkey (ORCID: 0000-0001-7835-2741)

(Conference Date: 5-7 March 2020)

(**DOI:** 10.31590/ejosat.araconf21)

ATIF/REFERENCE: Dokuz, Y., & Tüfekçi, Z. (2020). Investigation of the Effect of LSTM Hyperparameters on Speech Recognition Performance. *European Journal of Science and Technology*, (Özel Sayı), 161-168.

Öz

Bilgisayara dayalı hesaplamalı metotlar ve donanım teknolojilerindeki gelişmelerle birlikte, bilgisayarlar ses tanıma ve görüntü işleme gibi zor görevlerin üstesinden gelme konusunda daha güçlü hale gelmiştir. Ses tanıma, hesaplamalı veya analitik yöntemler kullanarak ses sinyallerinin metinsel karşılığını çıkarma görevidir. Ses tanıma aksanlar ve diller arasındaki değişkenlikler, güçlü donanım gereksinimleri, doğru modellerin üretilebilmesi için büyük veri setlerine olan ihtiyaç ve ses kalitesini etkileyen çevresel faktörlerden dolayı zor bir problemdir. Son yıllarda, Grafiksel İşleme Birimleri gibi donanım cihazlarının yükselen veri işleme yetenekleri yardımıyla derin öğrenme metotları, özellikle Özyinelemeli Sinir Ağları (ÖSA – Recurrent Neural Networks, RNN) ve RNN'in bir varyantı olan LSTM (Long Short Term Memory – Uzun Kısa Dönem Hafıza), ses tanıma alanında çok yaygın ve kabul gören metotlar haline gelmişlerdir. Literatürde, RNN ve LSTM ses tanıma ve ses tanımanın uygulamaları için katman sayısı, gizli katman sayısı ve yığın boyutu gibi çeşitli parametrelerle kullanılmaktadır. Kullanılan bu parametre değerlerin hangi kriterlere göre seçildiği ve bu parametre değerlerinin daha sonraki çalışmalarda da kullanılabilirliği ise incelenmemiştir. Bu çalışmada, LSTM hiperparametrelerinin ses tanıma performansına olan etkileri hata oranları ve derin mimari maliyeti dikkate alınarak incelenmiştir. Her bir parametre ayrı olarak değerlendirilmiş ve bu esnada diğer parametreler sabit tutulmuş ve parametrelerin ses verisi üzerindeki etkisi gözlemlenmiştir. Deneysel sonuçlarda, daha düşük hata oranları ve daha iyi ses tanıma performansı elde edebilmek için her parametrenin seçilen eğitim seti için farklı değerlere sahip olduğu görülmüştür. Bu çalışmanın sonuçlarına göre, LSTM için en uygun parametrelerin seçilmesinden önce ses veri kümesi üzerinde farklı deneyler yapılarak her bir parametre için en uygun değerin bulunması gerektiği gözlemlenmiştir.

Anahtar Kelimeler: Ses tanıma, Derin Öğrenme, RNN, LSTM, LSTM hiperparametreleri

Investigation of the Effect of LSTM Hyperparameters on Speech Recognition Performance

Abstract

With the recent advances in hardware technologies and computational methods, computers became more powerful for analyzing difficult tasks, such as speech recognition and image processing. Speech recognition is the task of extraction of text representation of a speech signal using computational or analytical methods. Speech recognition is a challenging problem due to variations in accents

* This paper was presented at the ***International Conference on Access to Recent Advances in Engineering and Digitalization (ARACONF 2020)***.

† Corresponding Author: Nigde Omer Halisdemir University, Engineering Faculty, Computer Engineering Department, Nigde, Türkiye, ORCID: 0000-0001-7202-2899, ytorun@ohu.edu.tr

and languages, powerful hardware requirements, big dataset needs for generating accurate models, and environmental factors that affect signal quality. Recently, with the increasing processing ability of hardware devices, such as Graphical Processing Units, deep learning methods became more prevalent and state-of-the-art method for speech recognition, especially Recurrent Neural Networks (RNNs) and Long-Short Term Memory (LSTMs) networks which is a variant of RNNs. In the literature, RNNs and LSTMs are used for speech recognition and the applications of speech recognition with various parameters, i.e. number of layers, number of hidden units, and batch size. It is not investigated that how the parameter values of the literature are selected and whether these values could be used in future studies. In this study, we investigated the effect of LSTMs hyperparameters on speech recognition performance in terms of error rates and deep architecture cost. Each parameter is investigated separately while other parameters remain constant and the effect of each parameter is observed on a speech corpus. Experimental results show that each parameter has its specific values for the selected number of training instances to provide lower error rates and better speech recognition performance. It is shown in this study that before selecting appropriate values for each LSTM parameters, there should be several experiments performed on the speech corpus to find the most eligible value for each parameter.

Keywords: Speech recognition, Deep learning, RNNs, LSTMs, LSTMs hyperparameters.

1. Introduction

Speech recognition is the task of extraction of a textual transcription of an uttered speech by a computation process (Yu and Deng, 2016). Speech recognition is an important an interdisciplinary area that combines linguistics, natural language processing, computer science, signal processing, and electrical engineering. Speech recognition has various application areas, such as voice interface systems, speaker identification, speech-to-text processing, and text-to-speech conversion.

Speech recognition is a challenging task due to several reasons. First of all, developing a unified model for all speakers from all accents of a language is a difficult problem. Second, speech recognition systems require powerful computational infrastructures. Third, more data are needed for achieving a good recognition accuracy, however, generating new datasets for new languages is costly. Fourth, background noise, microphone quality, speaker variations and other factors affect speech recognition systems. Deep learning systems are gained much attention to answer many of these challenges.

Deep learning is a set of algorithms that model high-level abstractions in data by using deep graphs with multiple transformations (Yu and Deng, 2016). Deep learning gained attention in scientific domains when Graphical Processing Units (GPUs) became more powerful and have high performance on computation tasks. Deep learning systems with GPUs processing abilities are being standard processing systems for complex tasks, such as speech recognition, image processing, and natural language processing.

In speech recognition, deep learning systems are utilized in acoustic model generation, feature extraction phase, and language model generation parts. Recurrent Neural Networks (RNNs) and Long Short Term Memory (LSTMs) networks which is a variant of RNNs became a state-of-the-art method in deep learning based speech recognition due to their sequential processing ability.

RNNs is one of the deep learning architectures which is efficient in processing sequential data inputs, like time series, or speech signals (Graves *et al.*, 2013). RNNs processes one input at a time and generates results at every time step. Fig. 1 presents a sample structure of an RNNs architecture. However, training RNNs is problematic because of the exploding or vanishing gradient problem at back-propagation process. To overcome this limitation, Long Short Term Memory (LSTMs) structure is proposed.

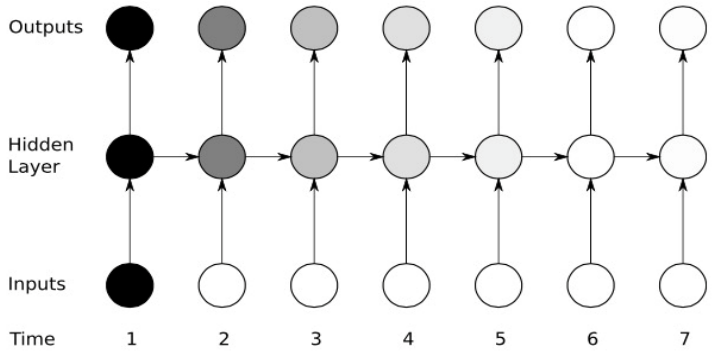

Fig. 1 Basic structure of RNNs

LSTMs is a special kind of RNN architecture which is capable of forgetting previous inputs that is not useful for current output (Graves *et al.*, 2013b). For defining usefulness of previous inputs, LSTMs proposes three gates to the network structure, input, forget and output gates. These gates carry several information between time steps and with the help of these gates exploding or vanishing gradient problem can be solved. LSTMs are also successfully applied in speech recognition tasks.

The application of RNNs and LSTMs in speech recognition has a wide range of variety. Many previous studies used RNNs and LSTMs for calculation of posterior probabilities which is performed by Gaussian mixture model (GMM) in classical speech recognition systems and have complexity in model generation. Recently, end to end systems gained attention in deep learning based speech recognition systems which use deep learning architectures at each part of speech recognition task.

In the literature, many studies are present for speech recognition using deep learning architectures. The most used architectures are RNNs and Convolutional Neural Networks (CNNs), but other architectures, such as Deep Neural Networks (DNNs), Deep Belief Networks (DBNs), and Deep Auto Encoder (DAE), are also used. In speech recognition using RNNs, Graves et al. (2013) proposed a deep recurrent neural network that uses LSTMs architecture that has 1 to 5 hidden layers. Miao et al. (2015) proposed an end-to-end speech recognition system that uses weighted finite-state transducers (WFSTs) and bidirectional LSTMs deep architecture. Hori et al. (2017) investigated the impact of RNNs language models on the performance of end-to-end speech recognition with character-based and word-based language models. Wang et al. (2019) proposed an RNN-T (RNN Transducer) method for Chinese Large Vocabulary Continuous Speech Recognition (LVCSR) to simplify training process and achieve good performance. Liu et al. (2018) proposed a limited-memory Broyden Fletcher Goldfarb Shannon (L-BFGS) optimization technique for RNNs language models to handle slow convergence of stochastic gradient descend optimization. He et al. (2019) investigated the use of end-to-end speech recognition with RNNs transducer for on-device streaming speech recognition tasks. Sainath et al. (2019) improved the performance of He et al. (2019) by using a two pass deep architectures. Gao et al. (2019) proposed and described a speech recognition hardware system that uses a delta RNNs accelerator (DeltaRNN) that is implemented on a Xilinx Zynq-7100 FPGA device to enable low latency RNNs computation. Toshniwal et al. (2018) proposed a single end-to-end speech recognition system that works on 9 different Indian languages. Lee et al. (2018) presented methods to accelerate RNNs language models for online speech recognition systems.

When the literature studies are analyzed, all of them have different configurations, optimization algorithms, and architectures for deep learning systems. Besides, all of these studies select hyperparameters for deep learning architecture based on their computation power. Thus, real effect of deep learning architectures on speech recognition performance is not easily observed from the literature. In this study, we investigated the effect of RNNs LSTMs hyperparameters on speech recognition performance. In particular, batch size, number of layers, number of hidden units, and number of epochs are evaluated on accuracy of speech recognition system.

The rest of this study is organized as follows. Section 2 presents the speech recognition problem, RNNs and LSTMs deep learning architectures, LSTMs hyperparameters, and the speech corpus. Section 3 presents the experimental results and discussion. Section 4 presents the conclusions.

2. Materials and Methods

In this section, first speech recognition problem is presented. Then, RNNs and LSTMs are introduced and LSTMs hyperparameters are presented. Finally, the speech corpus that is used in this study is presented.

2.1. Speech Recognition Problem

Speech recognition is the task of generating a textual transcription of an audio signal recorded from a speaker (Yu and Deng, 2016). Speech recognition systems became more prevalent in daily life of people with the increase of success rates of these systems. Speech recognition has many application areas, including voice interface systems, keyword search systems, data entry tasks, speaker identification, speech-to-text processing, and text-to-speech conversion, mobile applications, personal assistants, and digital annotation systems.

Speech recognition is challenging due to several reasons. First, speech recognition systems require big speech corpuses which include many accents of the languages. However, current speech corpuses are for only limited number of languages and accents. Second, speech recognition systems require powerful computational infrastructures to be able to analyze huge amount of speech data. Third, more data are needed for achieving a good recognition accuracy, however, generating new datasets for new languages is costly. Fourth, speech signals are complex and have many distorting factors, such as background noise, microphone quality, speaker based variations.

Speech recognition systems have three main steps (Yu and Deng, 2016). In first step, feature extraction process is performed on raw audio signals. Noise removal, signal conversion to feature domain, and feature extraction are performed in feature extraction. In second step, acoustic model and language model are processed. Acoustic model takes extracted features as inputs and generates an acoustic model score for variable-length feature space. Language model estimates language model score for the words in training corpus. In third step, hypothesis search combines acoustic model score and language model score to generate final score and text transcription of the audio signal. The basic speech recognition steps are presented in Figure 2.

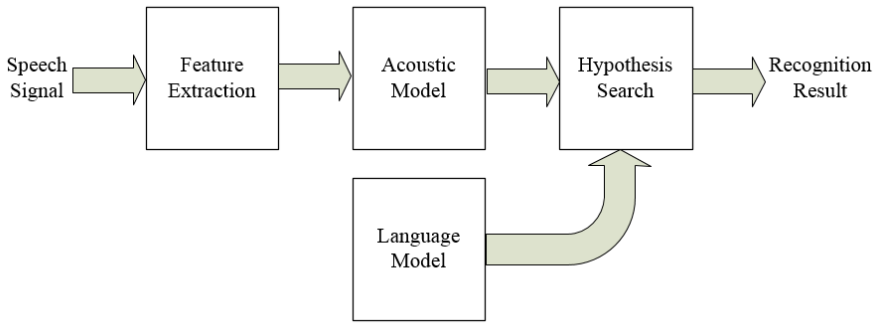

Fig. 2 Steps of speech recognition

Traditional speech recognizers are based on hidden Markov models (HMMs) with Gaussian mixture model (GMM) emission distributions, n-gram language models, and use beam search for decoding. HMM acoustic models assume that all audio frames are independent given the hidden sequence. Also, GMM could have a very high number of Gaussians and this impacts the recognition model to be very complex. For these reasons and recent success of deep learning architectures, speech recognition systems started using deep learning architectures, especially Recurrent Neural Networks, rather than GMM HMM systems.

2.2. Recurrent Neural Networks (RNNs)

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.

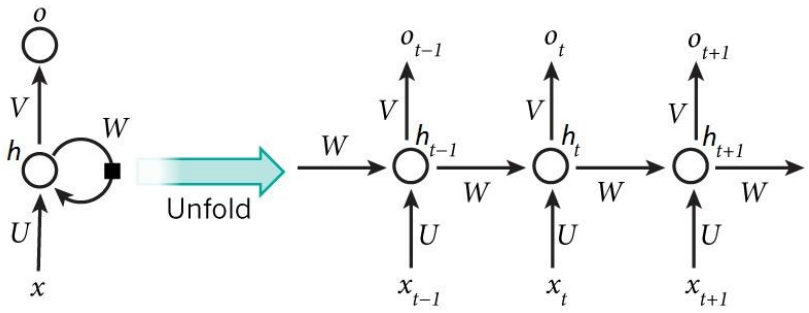

Fig. 3 An RNN (left) and unfolded over time (right)

Fig. 3 presents an RNN architecture which is unfolded over three time instances. x, h and o are input, hidden state and output vectors and U, W and V are weights of input, previous hidden states and current hidden state values, respectively. h_t and o_t are calculated based on (1) and (2). σ_h and σ_o are activation functions of hidden state and output vectors which regulate effect of input and hidden state instances. b_h and b_o are biases of hidden state and output vectors.

$$h_t = \sigma_h(U_{xt} + W_{ht-1} + b_h) \quad (1)$$

$$o_t = \sigma_o(V_{ht} + b_o) \quad (2)$$

There are several things to note in RNN architecture. First is, hidden state of the nodes, h_t in this case, is the memory of the network which passes information through time steps. Second, U, W and V are same for all time steps of the network. Third, there is no need to provide output for every time steps.

The basic RNNs structure is a useful model but has several limitations for many applications. Simple RNNs may have gradients which either increase or decrease exponentially over time. Thus the basic RNNs are difficult to train, and in practice can only model short-range effects. Long-term dependency and back propagation through time training are the most important weaknesses of RNNs. To overcome these limitations LSTMs network is proposed.

2.3. Long-Short Term Memory RNNs (LSTM RNNs)

LSTMs is a special kind of RNNs model which is proposed to overcome long-term dependency problem of classical RNNs (Graves *et al.*, 2013b). In a standard LSTMs network, there exists three layers, input, LSTMs and output layers. Input layers carries information of each time step to LSTMs layer. LSTMs layer produces outputs for every input instance and provides to the output layer. For this purpose, LSTMs proposes three gates; input, forget and output gates. These gates provide which proportion of the data will be allowed to pass on them.

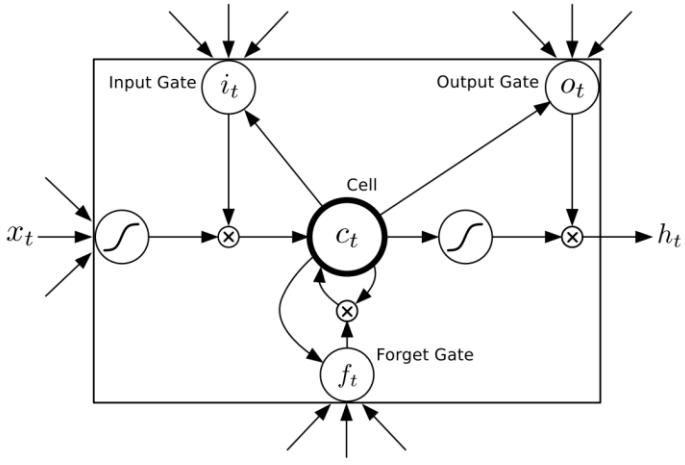

Fig. 4 LSTMs structure with gates

Fig. 4 presents a basic LSTMs network with gates. *x*, *c*, *h* and *o* are input, cell, hidden state and output vectors. *i*, *f* and *o* present input, forget and output gates. Equations (3) to (7) show how to calculate each vector. The weight matrices present the weights of denoted two parts connections, i.e. W_{xi} denotes the weight of input to input gate connection.

$$i_t = \sigma(W_{xi}x_t + W_{hi}h_{t-1} + W_{ci}c_{t-1} + b_i) \quad (3)$$

$$f_t = \sigma(W_{xf}x_t + W_{hf}h_{t-1} + W_{cf}c_{t-1} + b_f) \quad (4)$$

$$c_t = f_t c_{t-1} + i_t \sigma(W_{xc}x_t + W_{hc}h_{t-1} + b_c) \quad (5)$$

$$o_t = \sigma(W_{xo}x_t + W_{ho}h_{t-1} + W_{co}c_t + b_o) \quad (6)$$

$$h_t = o_t \sigma(c_t) \quad (7)$$

The proposed gates provide LSTMs to decide which information to write, store, or read from and to cells by using activation functions and weights.

2.3.1. LSTM Hyperparameters

LSTMs has several hyperparameters that affect performance of deep learning systems regardless of the application area. These hyperparameters are the values that modify LSTMs structure and can help achieve better performance with the same dataset. The main and the most important hyperparameters of LSTMs are listed and explained below:

- **Number of Layers:** This hyperparameter controls the number of layers of which deep learning systems will be built. When the number of layers are increased, the deep learning could better handle variations in the feature space but also complexity of the structure increases.
- **Number of Hidden Units:** This hyperparameter controls the number of hidden units that will be constructed on LSTMs. When the number of hidden units are increased, the more backward dependencies could be handled by the deep learning system, but also it increases complexity.
- **Batch Size:** This hyperparameter controls the number of samples that will be evaluated together before updating weights. When the batch size increases more complex systems and overfitting problems could occur. Contrarily when the batch size is decreased, the performance of deep learning systems would be lower.

2.4. The Speech Corpus

In this study, the CSTR VCTK (Centre for Speech Technology Voice Cloning Toolkit) corpus is used (Veaux *et al.*, 2017). The corpus includes 109 native English speakers with different accents. Each speaker has around 400 recording of sentences. We used a part of the corpus due to the limitation of our computational background. Random 10000 records are selected from all of the speakers for training dataset. Also, for test purposes random 1024 recordings are selected from the corpus that is not in training dataset.

Before the speech recognition is performed, several pre-processing steps are applied to prepare the recordings to deep learning system. First of all, Mel Frequency Cepstral Coefficients (MFCC) features of the audio files are extracted using sample rate of 8000. In MFCC feature extraction step, we used 25 milliseconds of window length and 10 milliseconds of window step. We removed all punctuation characters from the text files of audio recordings to better model MFCC features with the text transcriptions.

3. Results and Discussion

In this section experimental evaluation of LSTMs hyperparameters on VCTK corpus is presented. The hyperparameters of number of layers, number of hidden units, batch size, and the number of samples are used for evaluation. Each hyperparameter is evaluated while other hyperparameters remain constant.

3.1. Effect of Number of Layers

In this experiment, the effect of number of layers on speech recognition performance is evaluated. Number of hidden units, batch size, and number of samples are set to 200, 32, and 4096, respectively. Number of layers is selected as 1, 2 and 3, and label error rate and train cost are calculated. The effect of number of layers is presented in Figure 5 and Table 1.

Table 1. Effect of number of layers on test datasets

Number of Layers	Test Cost	Test LER
1	55.62	0.46
2	42.44	0.35
3	**36.22**	**0.29**
4	37.65	0.29
5	48.44	0.39

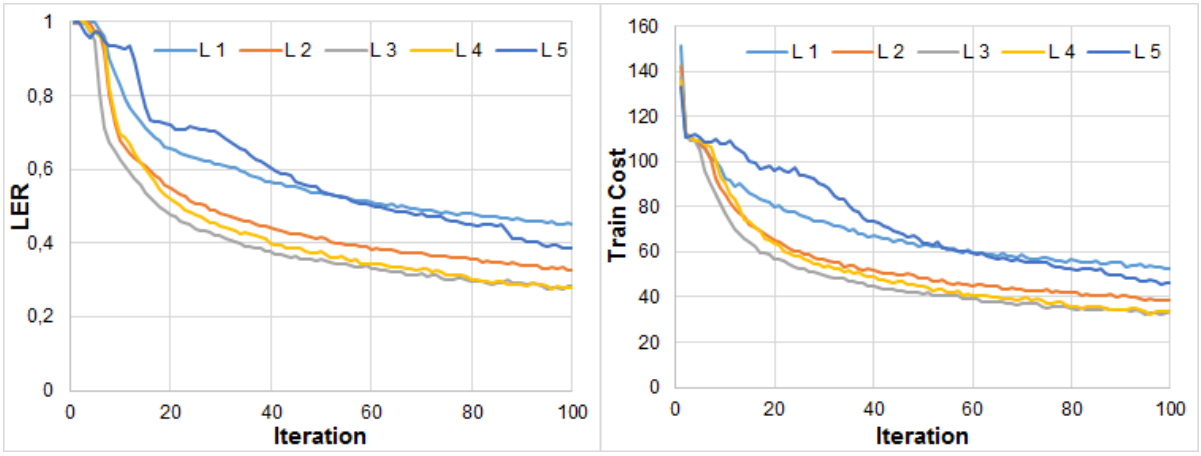

Fig. 5 Effect of number of layers on a) LER, b) train cost

As can be seen in Figure 5, as the number of layers increase, the deep networks become more accurate for both LER and train cost. However, after 4 layers, the increase of number of layers decrease the performance of the deep network. The best performance is observed at 4 layers followed with 3 layers, and 1 layer and 5 layer deep networks perform worst for speech recognition task.

3.2. Effect of Number of Hidden Units

In this experiment, the effect of number of hidden units on speech recognition performance is evaluated. Number of layers, batch size, and number of samples are set to 3, 32, and 4096, respectively. Number of hidden units is selected as 50, 100, 150, 200 and 250, and label error rate and train cost are calculated. The effect of number of hidden units is presented in Figure 6 and Table 2.

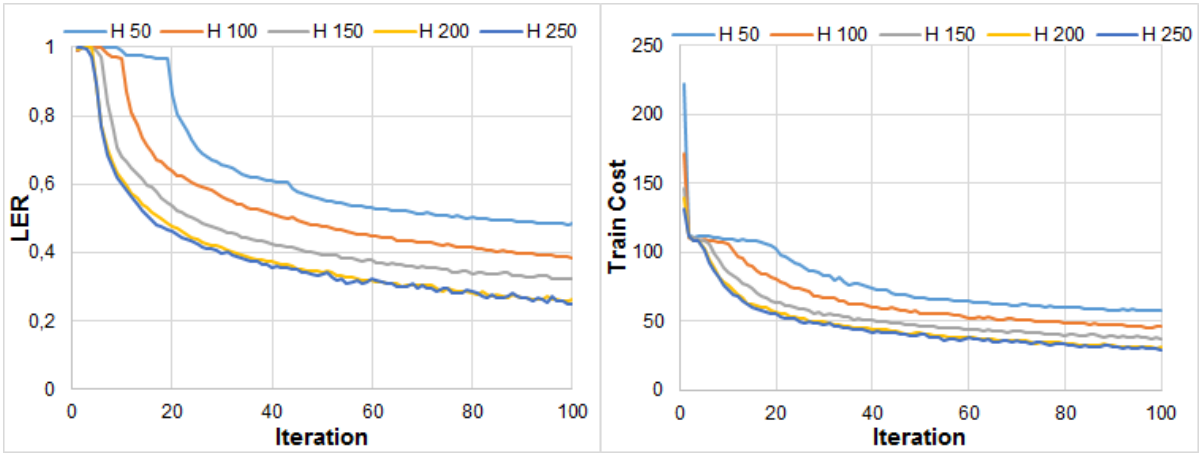

Fig. 6 Effect of number of hidden units on a) LER, b) train cost

Table 2. Effect of number of hidden units on test datasets

Number of Hidden Units	Test Cost	Test LER
50	58.54	0.48
100	46.31	0.39
150	41.69	0.34
200	34.37	0.27
250	**31.91**	**0.27**

As can be seen in Figure 6, as the number of hidden units increase, the accuracy of the deep networks increases too for both LER and train cost. However, after 200 hidden units, the accuracy keeps constant and do not increase. The best performances are observed for the number of hidden units of 200 and 250.

3.3. Effect of Batch Size

In this experiment, the effect of batch size on speech recognition performance is evaluated. Number of layers, number of hidden units, and number of samples are set to 3, 200, and 4096, respectively. Batch size is selected as 8, 16, 32 and 64, and label error rate and train cost are calculated. The effect of batch size is presented in Figure 7 and Table 3.

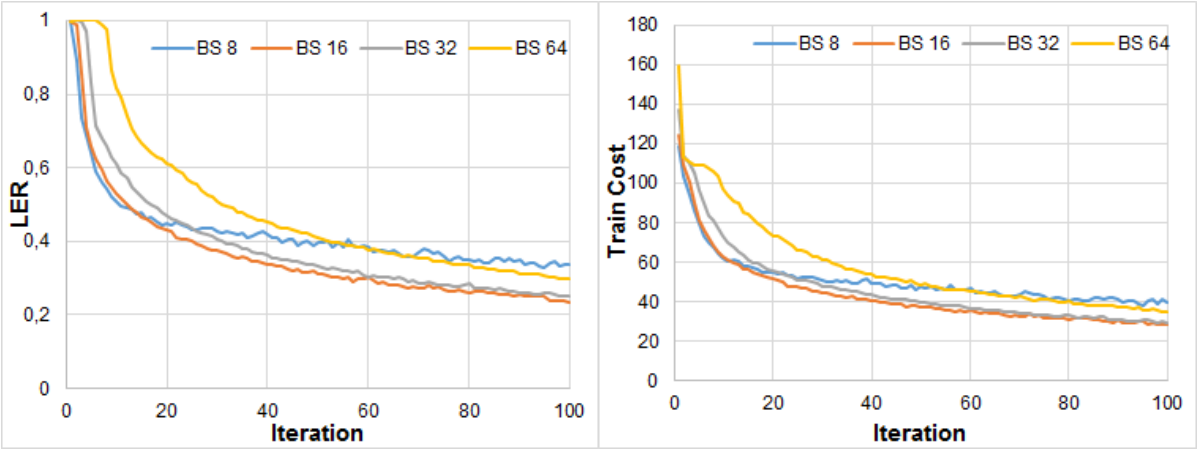

Fig. 7 Effect of batch size on a) LER, b) train cost

Table 3. Effect of batch size on test datasets

Batch Size	Test Cost	Test LER
8	40.71	0.35
16	34.68	0.28
32	**33.19**	**0.26**
64	40.73	0.31

As can be seen in Figure 7, the batch sizes of 16 and 32 provide better results than other batch sizes. Lower and higher batch sizes cause the deep networks perform worse and result higher LER and train costs.

4. Conclusion

In this study, we investigated the effect of RNNs LSTMs hyperparameters on speech recognition performance. In particular, number of layers, number of hidden units, and batch size are evaluated on performance of deep learning based speech recognition system. When the results are evaluated, all of the hyperparameters have impact on performance of speech recognition. When number of layers are increased, the performance increases until 3 and 4 layers. After 3 and 4 layers, the performance gets worse in our setup. When number of hidden units are increased, the performance of the system gets better. However, the improvement gets smaller with the increase of number of hidden units. When batch size is increased, the performance of the system gets better until 32 batch size. After batch size of 32, the performance of the system gets worse.

When the evaluation results of this study is investigated, all hyperparameters have effect on the performance of LSTMs for speech recognition. For our setup, best performance is observed for number of layers of 3, number of hidden units of 250, and for batch size of 32. The outcome of this study is that when using LSTMs for speech recognition, several experiments should be performed to find best LSTMs hyperparameters.

References

Gao, C., Braun, S., Kiselev, I., Anumula, J., Delbruck, T., & Liu, S. C. (2019, May). Real-time speech recognition for IoT purpose using a delta recurrent neural network accelerator. In 2019 IEEE International Symposium on Circuits and Systems (ISCAS) (pp. 1-5). IEEE.

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

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

He Y., Sainath T. N., Prabhavalkar R., McGraw I., Alvarez R., Zhao D., Rybach D., Kannan A., Wu Y., Pang R., Liang Q., Bhatia D., Shangguan Y., Li B., Pundak G., Sim K. C., Bagby T., Chang S., Rao K., and Gruenstein A. (2019, May). Streaming end-to-end speech recognition for mobile devices. In ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP) (pp. 6381-6385). IEEE.

Hori, T., Watanabe, S., Zhang, Y., & 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.

Lee, K., Park, C., Kim, N., & Lee, J. (2018, April). Accelerating recurrent neural network language model based online speech recognition system. In 2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP) (pp. 5904-5908). IEEE.

Liu, X., Liu, S., Sha, J., Yu, J., Xu, Z., Chen, X., & Meng, H. (2018, April). Limited-memory bfgs optimization of recurrent neural network language models for speech recognition. In 2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP) (pp. 6114-6118). IEEE.

Miao, Y., Gowayyed, M., & Metze, F. (2015, December). 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.

Sainath T. N., Pang R., Rybach D., He Y., Prabhavalkar R., Li W., Visontai M., Liang Q., Strohman T., Wu Y., McGraw I., and Chiu C.-C. (2019). Two-Pass End-to-End Speech Recognition, In INTERSPEECH 2019, Graz, Austria, 2019.

Toshniwal, S., Sainath, T. N., Weiss, R. J., Li, B., Moreno, P., Weinstein, E., & Rao, K. (2018, April). Multilingual speech recognition with a single end-to-end model. In 2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP) (pp. 4904-4908). IEEE.

Veaux C., Yamagishi J., and MacDonald K. (2017, 04/02/2020). Cstr vctk corpus: English multi-speaker corpus for cstr voice cloning toolkit. Available: https://datashare.is.ed.ac.uk/handle/10283/2651.

Wang, S., Zhou, P., Chen, W., Jia, J., & Xie, L. (2019, November). Exploring RNN-Transducer for Chinese speech recognition. In 2019 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC) (pp. 1364-1369). IEEE.

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