Avrupa Bilim ve Teknoloji Dergisi
Özel Sayı 36, S. 87-90, Mayıs 2022

Araştırma Makalesi

www.ejosat.com ISSN:2148-2683

European Journal of Science and Technology
Special Issue 36, pp. 87-90, May 2022

Research Article

Parallel Gated Recurrent Unit Networks as an Encoder for Speech Recognition

Zekeriya Tüfekci[1], Gökay Dişken[2*]

[1*] Cukurova University, Faculty of Engineering, Departmant of Computer Engineering, Adana, Turkey, (ORCID: 0000-0001-7835-2741), ztufekci@cu.edu.tr
[2*] Adana Alparslan Turkes Science and Technology University, Faculty of Engineering, Departmant of Electrical-Electronics Engineering, Adana, Turkey, (ORCID: 0000-0002-8680-0636), gdisken@atu.edu.tr

(1st International Conference on Engineering and Applied Natural Sciences ICEANS 2022, May 10-13, 2022)

(**DOI:** 10.31590/ejosat.1103714)

ATIF/REFERENCE: Tüfekci, Z., & Dişken, G. (2022). Parallel Gated Recurrent Unit Networks as an Encoder for Speech Recognition. *European Journal of Science and Technology*, (36), 87-90.

Abstract

Listen, Attend and Spell (LAS) network is one of the end-to-end approaches for speech recognition, which does not require an explicit language model. It consists of two parts; the encoder part which receives acoustic features as inputs, and the decoder network which produces one character at a time step, based on the encoder output and an attention mechanism. Multi-layer recurrent neural networks (RNN) are used in both decoder and encoder parts. Hence, the LAS architecture can be simplified as one RNN for the decoder, and another RNN for the encoder. Their shapes and layer sizes can be different. In this work, we examined the performance of using multi RNNs for the encoder part. Our baseline LAS network uses an RNN with a hidden size of 256. We used 2 and 4 RNNs with hidden sizes of 128 and 64 for each case. The main idea behind the proposed approach is to focus the RNNs to different patterns (phonemes in this case) in the data. At the output of the encoder, their outputs are concatenated and fed to the decoder. TIMIT database is used to compare the performance of the mentioned networks, using phoneme error rate as the performance metric. The experimental results showed that proposed approach can achieve a better performance than the baseline network. However, increasing the number of RNNs does not guarantee further improvements.

Keywords: Attention networks, Recurrent neural networks, Speech recognition, Timit.

Konuşma Tanıma için Kodlayıcı Olarak Paralel Kapılı Tekrarlayan Birim Ağları

Öz

Listen, Attend and Spell (LAS) ağı konuşma tanıma için belli bir dil modeline gereksinim duymayan uçtan-uca yaklaşımlardan biridir. İki kısımdan oluşur; akustik öznitelikleri girdi olarak alan kodlayıcı kısmı, kodlayıcı çıkışı ve dikkat mekanizmasına bağlı olarak bir zaman adımında tek bir karakter üreten kod çözümleyici kısmı. Hem kod çözümleyici hem de kodlayıcı kısımlarında çok katmanlı tekrarlayan sinir ağları (RNN) kullanılır. Bu nedenle LAS mimarisi kod çözümleyici için bir RNN ve kodlayıcı için bir başka RNN olarak basitleştirilebilir. Şekilleri ve katman boyutları farklı olabilir. Bu çalışmada, kodlayıcı kısmı için çoklu RNN kullanımının performasını inceledik. Temel alınan LAS ağı 256 gizli boyutu olan bir RNN kullanmaktadır. 128 ve 64 gizli boyutları için 2 ve 4 RNN kullandık. Önerilen yaklaşımın ardındaki ana fikir, RNN'leri verilerdeki farklı örüntülere (bu çalışma için fonemler) odaklamaktır. Kodlayıcının çıkışında bunların çıkışları birleştirilir ve kod çözümleyiciye iletilir. TIMIT veritabanı, performans metriği olarak fonem hata oranı seçilerek bahsedilen ağların performansını karşılaştırmak için kullanılmıştır. Deneysel sonuçlar, önerilen yaklaşımın temek alınan ağdan daha iyi bir performans elde edebileceğini göstermiştir. Ancak RNN'lerin sayısını artırmak daha fazla iyileşmeyi garanti etmemektedir.

Anahtar Kelimeler: Dikkat ağları, Tekrarlayan sinir ağları, Konuşma tanıma, Timit

[*] Corresponding Author: gdisken@atu.edu.tr

1. Introduction

Traditional speech recognition systems consists of an acoustic model, a language model, a pronunciation model, etc. [1]. Hidden Markov Model (HMM) method was the dominating approach for speech recognition [2], however, end-to-end systems, where different components trained jointly, gained popularity recently. Deep neural networks (DNNs) based systems achieved a higher performance on several speech recognition benchmarks [3]. End-to-end training was achieved with the aid of Connectionist Temporal Classification (CTC) and Recurrent Neural Network (RNN) [4], [5].

Another popular architecture, namely Listen, Attend and Spell (LAS) is an alternative end-to-end system which can emit one character at a time at the output, without CTC or language model [6]. Further, the LAS network does not make any independence assumption on the output probabilities, contrary to the CTC. Two main parts of the LAS network are encoder and decoder, where both are realized with RNNs, and this type of networks called as sequence-to-sequence networks [7]–[9].

In this work, we use a similar network to the LAS architecture and its modifications for speech recognition on TIMIT database, which is a classical database for phoneme/speech recognition or related studies [10]–[12]. The inputs of the networks are filter bank energies, which usually work well with neural networks compared to the conventional mel-frequency cepstral coefficients [13], [14]. For the modified encoders, we increased the number of RNNs to 2 and 4, instead of the single RNN of the baseline. Also, the baseline system has the hidden size of 256, where we tried 128 and 64 in the modified networks. Using 2 (or 4) parallel RNNs, we aim to achieve finer models for different patterns, as different RNNs may focus on different sections of the data during training. The experiments showed that the proposed modifications increase the performance of the LAS network, and also, they can have a reduced number of trainable parameters, compared to the baseline network.

2. Modified LAS Network

We used the modified versions of [6]. The original LAS network used pyramidal long short-term memory (LSTM) layers in the encoder. In this work, gated recurrent unit (GRU) [9] layers were preferred as they include one less gate than the LSTM, and can achieve a higher performance for small databases [15]. GRU layers were also used in the decoder, and a fully-connected network with two layers was used as the attention layer [16]. The baseline LAS network is given in Figure 1. The encoder part is represented with green, and decoder part is represented by blue. The inputs of the decoder are ground-truth phoneme labels in the training. In the evaluation mode, the inputs are the predictions of the previous time step. Fully-connected layers serve as embedding layers for both encoder and decoder. After the embedding layer of the encoder, sigmoid activation is applied.

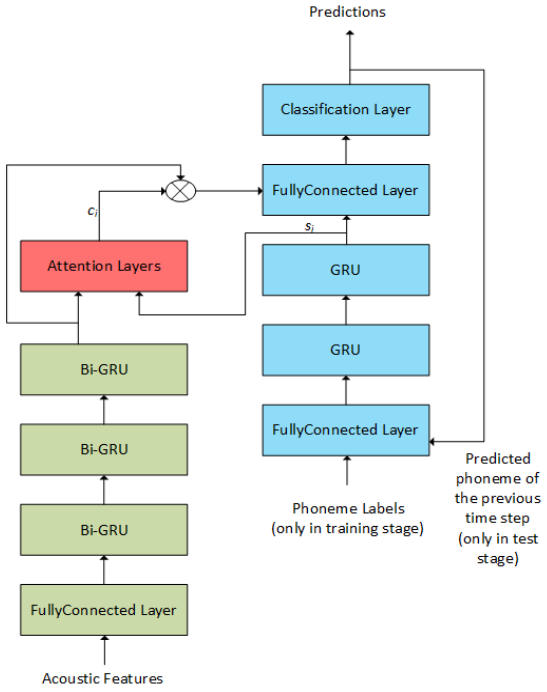

Fig. 1 Block diagram of LAS network

The GRU layers of the encoder are bidirectional, hence both the input sequence and its reversed form are learned. This is also true for the proposed multi RNN layers. The output of the decoder at time step i depends on current decoder state (si) and the context vector (ci, obtained via attention layer). When we use more than one GRU in parallel, their outputs are concatenated to create the decoder state. Three fully-connected layers are used for the attention. One of those layers takes its inputs from the decoder, and another layer takes them from the final GRU layer (or layers for the parallel GRUs) of the encoder. The outputs of these two fully-connected layers are added together, and sent to the third layer. It should be noted that the size of the inputs from the encoder will vary (depending on the number of GRUs and their hidden sizes). The dimensions were adjusted accordingly in the experiments. The output of the third layer is the attention vector. The purpose of the attention layer is to focus the decoder on a few frames of the encoder output [6], as not all of the frames affect the output equally. The output of the attention layer is therefore can be thought of a mask that weights features based on their relation with the target output. Therefore, encoder output states are multiplied with the attention mask.

After obtaining the masked output vector, it is concatenated with the last GRU layer of the decoder and served as input to the fully-connected layer before the classification layer. Softmax function is used in the classification layer where the phoneme with the maximum probability is chosen as the output of that time step. The classification stops when end-of-sentence token is produced.

Table 1 shows the total number of trainable parameters for the encoder parts of the baseline and modified LAS networks. Except the two GRUs with 128 hidden size, the other configurations have less trainable parameters compared to the baseline architecture. This will lead to faster training and execution times. Hence, even if the proposed architectures perform similar to the baseline, they will still possess an advantage due to the reduced number of parameters.

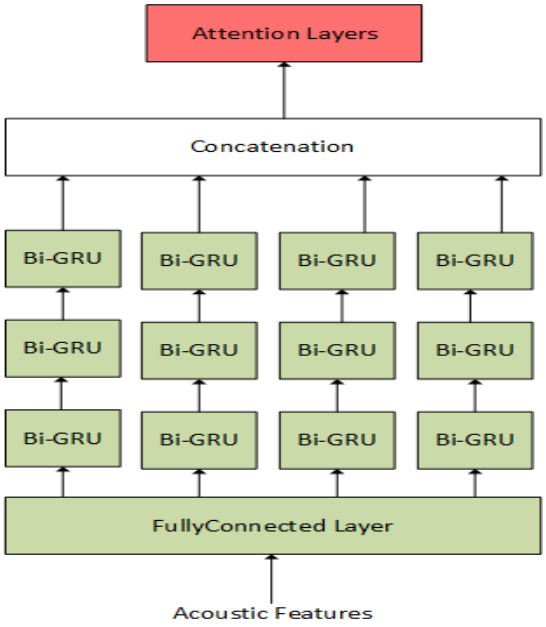

Fig. 2 Multi GRU approach for the encoder part

Table 1. Total number of trainable parameters for the encoder part

Hidden size * # GRUs	# Parameters	Absolute Change
Baseline / 256 * 1	3,179,776	-
128 * 2	1,787,392	-1,392,384
128 * 4	3,566,848	+387,072
64 * 2	553,984	-2,625,792
64 * 4	1,100,032	-2,079,744

3. Experiments

The modified LAS networks' performance for phoneme recognition was examined on TIMIT database, where 3696 sentences from 462 speakers were used for training (SA records were removed), 400 sentences were chosen as the validation data, and the test set contains 192 sentences from 24 speakers. Transcriptions of TIMIT are based on 61 phonemes. However, typical approach is to map those phonemes into 48 for modelling. Further, confusions among some of the phones are not considered as error, so only 39 phoneme categories are used in the evaluation [17].

For the baseline system, all of the GRU layers have 256 hidden states. The networks were trained for 200 epochs, using a batch size of 16, and ADAM optimizer. Learning rate was 3e-4 initially, and reduced by half if there is no improvement after ten epochs. Also, dropout was used after the GRU layers with a 0.5 probability. Pytorch was used to build, test, and train the network.

Torchaudio was used for feature extraction, mel-scaled log filter bank energies were used as the features. Frame length was 25 ms, and frame shift was 10 ms. The number of triangular filters was 30. All other parameters were used as their default values.

The results for the baseline system and the modified versions are given in Table 2. Using four GRUs with the size of 128 yielded the best performance, 16.71%. Only the case where two GRUs with the size of 64 performed worse than the baseline. The results indicate that using a single GRU may not always be the best option. Although the main advantage of the GRUs is the ability to model long relations of the input, using several GRUs to specialize in the different sections of input can provide extra information to network, which may vanish otherwise. On the other hand, increasing the number of parallel GRUs will increase the training time as the required mathematical operations will increase. Reducing the hidden size may counter this problem, but the observed results showed that it may deteriorate the performance as using 64 hidden nodes performed poor compared to the others.

Table 2. Phoneme error rates (PER, %) for different encoder configurations

Hidden size * # GRUs	PER	Relative Reduction
Baseline / 256 * 1	17.67	-
128 * 2	17.11	3.17
128 * 4	**16.71**	5.43
64 * 2	17.84	-0.96
64 * 4	17.21	2.60

It should be noted that at the first glance, the performance improvements may be assigned to the increased number of hidden nodes (hence number of trainable parameters). As the GRUs used in the encoder are bi-directional, the output size of the baseline encoder is 512. The best performing system (4 GRUs with 128 nodes) has the output size of 1024. However, if the 2 GRU of 128 nodes is considered, it has the same output size of the baseline, which is 512. Its performance is still 3.17% better than the baseline, relatively. Also, the number of trainable parameters in that case is almost 1,4 million less than the baseline.

Nevertheless, when we reduced the number of trainable parameters of the encoder over 2,6 million, the performance loss was less than 1%. Even a 2.6% PER improvement was achieved with 2 million less parameters for the four GRUs with 64 hidden sizes. Therefore, besides the number of trainable parameters, the architecture itself plays an important role on the final recognition performance. Still, there is a trade-off between the number of parameters and the performance of the network. The results tell us that after reaching a limit, the recognition performance will drop if we further decrease the trainable parameters. This is mainly due to the fact that the network will not have the flexibility to model different patterns and tend to overfit to the training data. So, in the test stage where the network can encounter with some unseen data, it will likely to be misclassified.

4. Conclusions

Phoneme recognition performance of the modified LAS network was examined in this work, using the TIMIT database. Instead of the single GRU of the encoder part, we used two and four GRUs in parallel. Hidden node sizes of 128 and 64 were considered, compared to the 256 nodes of the baseline network. Two GRUs with 64 hidden nodes performed worse than the baseline, however the performance loss was under 1% and about 3-fold less learnable parameters were used. On the other hand, the other modifications achieved better performances. The best performance was achieved with four GRUs with 128 hidden nodes, with 16.71% PER. The relative error reduction for the best case was 5.43%.

The results indicate that the encoder-decoder type end-to-end networks' performances could be increased with this slight modification. By using parallel GRUs, the number of trainable parameters were also reduced, without compromising the performance. However, as the network gets smaller, its performance will decrease. So, a sweet point should be aimed with comparing different networks experimentally.

For further improvements, using different activation functions, different dropout rates, etc. will be considered in the future works. The network can focus on different patterns found in the data with this approach. Also, using different number of hidden nodes in each network may affect the performance. A more detailed investigation is necessary to analyze performances of different architectures. For these mentioned modifications, behavior of the attention may also play an important role in the recognition performance, hence it should also be considered.

References

[1] C. Kim et al., "A Review of On-Device Fully Neural End-to-End Automatic Speech Recognition Algorithms," in *2020 54th Asilomar Conference on Signals, Systems, and Computers*, 2020, pp. 277–283.

[2] A. P. Varga and R. K. Moore, "Hidden Markov model decomposition of speech and noise," in *International Conference on Acoustics, Speech, and Signal Processing*, 1990, pp. 845–848.

[3] G. Hinton et al., "Deep Neural Networks for Acoustic Modeling in Speech Recognition: The Shared Views of Four Research Groups," *IEEE Signal Process. Mag.*, vol. 29, no. 6, pp. 82–97, Nov. 2012.

[4] Yiğit, E., Özkaya, U., Öztürk, Ş., Singh, D. and Gritli, H. "Automatic detection of power quality disturbance using convolutional neural network structure with gated recurrent unit", *Mobile Information Systems*, 2021.

[5] A. Graves, A. Mohamed, and G. Hinton, "Speech recognition with deep recurrent neural networks," in *2013 IEEE International Conference on Acoustics, Speech and Signal Processing*, 2013, pp. 6645–6649.

[6] W. Chan, N. Jaitly, Q. Le, and O. Vinyals, "Listen, attend and spell: A neural network for large vocabulary conversational speech recognition," in *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2016, pp. 4960–4964.

[7] I. Sutskever, O. Vinyals, and Q. V. Le, "Sequence to sequence learning with neural networks," in *27th International Neural Information Processing Systems*, 2014, pp. 3104–3112.

[8] D. Bahdanau, K. Cho, and Y. Bengio, "Neural Machine Translation by Jointly Learning to Align and Translate," in *3rd International Conference on Learning Representations*, 2015, pp. 1–15.

[9] K. Cho et al., "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)*, 2014, pp. 1724–1734.

[10] M. T. S. Al-Kaltakchi, W. L. Woo, S. S. Dlay, and J. A. Chambers, "Comparison of I-vector and GMM-UBM approaches to speaker identification with TIMIT and NIST 2008 databases in challenging environments," in *2017 25th European Signal Processing Conference (EUSIPCO)*, 2017, pp. 533–537.

[11] T. Drugman, Y. Stylianou, Y. Kida, and M. Akamine, "Voice Activity Detection: Merging Source and Filter-based Information," *IEEE Signal Process. Lett.*, vol. 23, no. 2, pp. 252–256, Feb. 2016.

[12] I. Cohen and B. Berdugo, "Speech enhancement for non-stationary noise environments," *Signal Processing*, vol. 81, no. 11, pp. 2403–2418, Nov. 2001.

[13] S. Davis and P. Mermelstein, "Comparison of parametric representations for monosyllabic word recognition in continuously spoken sentences," *IEEE Trans. Acoust.*, vol. 28, no. 4, pp. 357–366, Aug. 1980.

[14] F. Seide, G. Li, X. Chen, and D. Yu, "Feature engineering in Context-Dependent Deep Neural Networks for conversational speech transcription," in *2011 IEEE Workshop on Automatic Speech Recognition & Understanding*, 2011, pp. 24–29.

[15] S. Yang, X. Yu, and Y. Zhou, "LSTM and GRU Neural Network Performance Comparison Study: Taking Yelp Review Dataset as an Example," in *2020 International Workshop on Electronic Communication and Artificial Intelligence (IWECAI)*, 2020, pp. 98–101.

[16] T. Luong, H. Pham, and C. D. Manning, "Effective Approaches to Attention-based Neural Machine Translation," in *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, 2015, pp. 1412–1421.

[17] K.-F. Lee and H.-W. Hon, "Speaker-independent phone recognition using hidden Markov models," *IEEE Trans. Acoust.*, vol. 37, no. 11, pp. 1641–1648, 1989.