International Journal of Scientific & Engineering Research Volume 10, Issue 11, November-2019
ISSN 2229-5518

Leveraging End-to-End Speech Recognition with Neural Architecture Search

Ahmed Baruwa Mojeed Abisiga Ibrahim Gbadegesin Afeez Fakunle

Abstract—Deep neural networks (DNNs) have been demonstrated to outperform many traditional machine learning algorithms in Automatic Speech Recognition (ASR). In this paper, we show that a large improvement in the accuracy of deep speech models can be achieved with effective Neural Architecture Optimization at a very low computational cost. Phone recognition tests with the popular LibriSpeech and TIMIT benchmarks proved this fact by displaying the ability to discover and train novel candidate models within a few hours (less than a day) many times faster than the attention-based seq2seq models. Our method achieves test error of 7% Word Error Rate (WER) on the LibriSpeech corpus and 13% Phone Error Rate (PER) on the TIMIT corpus, on par with state-of-the-art results.

Index Terms—ASR, AutoML, CTC, DNNs, Neural Architecture Optimization, Reinforcement Learning, seq2seq.

1 Introduction

Many difficult learning tasks can be easily accomplished by stacking layers of neurons as shown by recent research results [1], [2], [3], [4], [5], [6] compared with traditional machine learning algorithms that have been in common use in the past. Deep neural networks improved a lot of research areas such as Computer Vision [7], [8], [9] and many Natural Language Processing tasks [10], [11], [12].
Their successes have been due to the availability of large computational capacity, improved optimization algorithms and the enormous amount of data that are being generated on a regular basis across the globe, but the most fascinating is their ability to self construct features from the input data, in contrast to other machine learning algorithms which require hand engineered features that takes a considerable amount of expertise and time.
Recurrent Neural Networks (RNN) are the natural choice when dealing with time series tasks like DNA sequencing, machine translation and text summarization.
Even though deep neural networks do not require hand-designed features, they have many hyper-parameters to be optimized. Getting optimal values of these hyper-parameter is quite difficult [13] and can often be found with random search [14]. Graphical techniques such as Bayesian Optimization techniques [15] are quite promising when exploring a wide search space but they require a lot of computation time to produce optimal results.
In this paper, we present neural architecture search approach to finding hyper-parameters for a neural architecture using reinforcement learning. We use a controller RNN to run experiments with child networks and feed it back with performance from such a child network, as a reward signal. Experiments on the LibriSpeech and TIMIT corpus show that this approach yields a very good results on the test corpus described above.

- *Ahmed Baruwa (Corresponding Author)*
  Data and Analytics, KPMG NG.
  E-mail: aobaruwa@gmail.com
- *Mojeed Abisiga*
  Data and Analytics, KPMG NG.
  E-mail: abisigadamilola@gmail.com
- *Ibrahim Gbadegesin*
  Electronic and Electrical Engineering, OAU.
  E-mail: ibrahimgbadegesin@gmail.com
- *Afeez Fakunle*
  Electronic and Electrical Engineering, OAU.
  E-mail: abiodunfakunle007@gmail.com

2 Related Work

There have been a ton of phenomenal research results from both academia and the industry towards solving Large Vocabulary Speech Recognition (LVSR). We highlight a few remarkable ones in this section.

2.1 Probabilistic Graphical Models

Acoustic models based on HMMs [16] are classical approaches that have exhibited impressive performance for decades. Gaussian mixture models (GMMs) work

well as frame-level classifiers for computing the probability of features vectors given each HMM state. These methods involve excruciating effort of forced alignment strategies between phones and the transcripts which involves a lot of expertise and time, in the training pipeline and yet prone to errors.
Decoding is usually based on the direct application of Bayes theorem on acoustic feature vectors.
Given a sequence of acoustic features (or observations), an acoustic model that defines a probability distribution $p(o|w)$ of an observation given a symbol, w and a language model that defines the distribution $p(w)$ which defines the the probability of every word in the vocabulary $\mathcal{L}$ existing in a surrounding context of neighbouring words. The best sequence of words is that which maximises the product of these two probabilities.

$$w^* = \underset{w \in \mathcal{L}}{\operatorname{argmax}} \{p\left(o|w\right) \cdot p\left(w\right)\}$$

2.2 Attention-Based Models

Attention-based speech models essentially are encoder-decoder configurations made up of Recurrent Neural Networks(RNNs) that generate an output sequence $(y_1, \ldots, y_T)$ given an acoustic input sequence x; with the encoder representing the knowledge from the inputs as a sequence of symbols $h = \{\, h1, \ldots, h_T\}$ suitable for the attention mechanism to work with [17]. Unlike the graphical models, they learn alignments between the input and output sequence during training by associating a weight with each frame in the entire input speech sequence by computing the decomposed joint probabilities

$$p\left(\mathbf{y}\right) = \prod_{t=1}^{T} p\left(y_t|\{y_1, \ldots, y_{t-1}\}, v\right)$$

where v is a context vector for a particular time step t and a weighted sum of the encoder outputs, h.

$$v_t = \sum_{t}^{T} \alpha_{tj} h_t$$

The weight α_{tj} of each annotation h_j is given by

$$\alpha_{tj} = \frac{\exp\left(e_{tj}\right)}{\sum_{k=1}^{T_x} \exp\left(e_{tk}\right)}$$

where

$$e_{tj} = \psi\left(s_{t-1}, h_j\right)$$

is a design dependent alignment model which scores how well a speech frame matches its own label. Even though these techniques perfectly handle alignment, they tend to perform based on the lengths of the training data. Our methods, described in a later section do not suffer from any of these impediments.

2.3 Bayesian Optimization

Bayesian Optimization(BayesOpt) is the go-to method for solving optimization problems of black-box models where the cost of evaluating these functions at certain points is very expensive. BayesOpt treats the hyper-parameter optimization problem by assuming an unknown function $\lambda(x)$ as samples from a Gaussian Process (GP), optimizing iteratively on a bounded set $x \in \mathcal{R}$ and maintains a posterior distribution for this function as observations are made.
Computing this posterior probability distribution is usually expensive and hence impractical for use-cases where the hyper-parameters are more than three orders of magnitude. However, BayesOpt has shown much success in in various computer vision tasks [18] and natural language processing tasks [19]. AutoML methods such as Neural Architecture Search(NAS), described in the following section however do not suffer from these inadequacies.

2.4 Neural Architecture Search (NAS)

Using Policy gradient technique in discovering DNN architectures by training a controller RNN to sample child networks and taking their accuracies as expected rewards towards improvement has been successfully implemented on Image classification and language modelling [20], [21]. The output of the controller RNN is usually a list of hyper-parameters specific to the learning task.
Vanilla NAS is deemed impractical due to the large amount of computational resources required in training a single child network. Parameter sharing among multiple child networks saves a lot of GPU hours (about 1000x) by concentrating only on a selected subset of the main search space as discussed in [22].
By directly optimizing the WER, a non differentiable metric with policy gradient, a better performance can be achieved compared to the conventional maximum likelihood method for setting up objective functions to train deep speech models. In this paper, we extend the use of AutoML with NAS to ASR by training an agent to minimize a maximum likelihood objective along with the WER and improve in the accuracy of speech recognition benchmarks; yeilding 7%, 13% WERs on both the TIMIT and LibriSpeech corpora respectively.

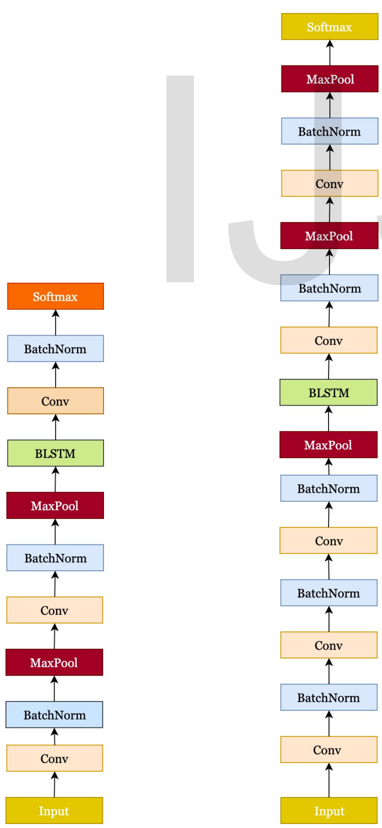
International Journal of Scientific & Engineering Research Volume 10, Issue 11, November-2019
ISSN 2229-5518

3 Methodology

We follow closely policy gradient method as discussed in [20], [22], [23] for our Neural Architecture Search. With the enormous size of our search space, it was expedient that a distributed computing technique be used in training the child sample models, we speed up the training by HPC techniques [24] on 8 GPUs.

3.1 Model Architecture

In this paper, we limit the search space to involve just the convolution operation, maxpool operation, Batch Normalization and the Recurrent blocks. The controller RNN learns to improve on its choice of parameters for each child network sample. The best architectures our controller RNN found for both the libriSpeech and TIMIT training corpora are shown below figure 3.2.



(a) (b)

Figure 3.1 a) Neural architecture found by NAS for the LibriSpeech corpus. b) Neural architecture found by NAS for the TIMIT corpus.

3.2 Beam Search Decoding

We incorporate beam search during inference with a size of 128, obtained by a hyper-parameter sweep; we got about a 5% improvement in the performance of the child networks at the cost of speed of traversing the beam width. It is essentially a character-level beam search decoding which zips through candidate transcript sequence.

We train a 3-gram, 5-gram and a 7-gram language model on common crawl [1]. The relative performances are summarised in tables 1 and 2.

Decoding is done by beam-searching for the output y that maximizes $\phi(c)$ given by

$$\phi(c) = \log(\mathbb{P}(c|x)) + \alpha \log(\mathbb{P}_{lm}(c)) + \beta \text{ count(c)}$$

The weights α and β reduce bias caused by the language model and encourages the effect caused by the frequency of words in the transcripts respectively.

3.3 Bi-directional LSTM(BLSTM)

In this work, the RNN flavour we choose in running our experiments is the Long Short-Term Memory network (LSTM) [25]. Even though it has a more complex architecture, it performed a lot better than both conventional RNNs and Gated Recurrent Units (GRU). Figure 3.2 illustrates the deep bidirectional LSTM network.We used gradient clipping [12] to solve the exploding gradients problem. Another benefit of the LSTM is that it readily solves the vanishing gradient problem in deep RNNs. Our method performs well without the use of an external language model (LM). The inherent capability of LSTMs to remember long-term information makes it easy for them to learn an implicit language model during training. Conventional RNNs fail to capture previous contexts; hence our use of bidirectional LSTMs which learn information in both forward and backward directions. The LSTM architecture used in this work is described by the following equations:

$$\begin{aligned}
i_t &= \sigma\left(W_{xi}x_t + W_{hi}h_{t-1} + b_i\right) \\
f_t &= \sigma\left(W_{xf}x_t + W_{hf}h_{t-1} + b_f\right) \\
o_t &= \sigma\left(W_{xo}x_t + W_{ho}h_{t-1} + b_o\right) \\
c_t &= f_t * c_{t-1} + i_t * \tanh\left(W_{xc}x_t + W_{hc}h_{t-1} + b_c\right) \\
h_t &= o_t * \tanh\left(c_t\right)
\end{aligned}$$

1. https://commoncrawl.org

International Journal of Scientific & Engineering Research Volume 10, Issue 11, November-2019
ISSN 2229-5518

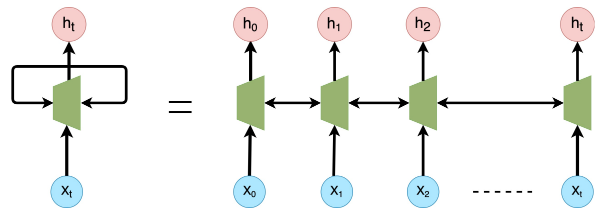


Figure 3.2. Structure of an unrolled BLSTM.

3.4 Data Augmentation

By adding gaussian noise to the speech frame [26], we artificial synthesised data. This created about 2000 additional hours of audio data and improved the performance of the model with about 10% reduction in the WER. As proposed by Park et al., [27] we also perform Augmentation on the mel-frequency filter banks by adding noise in the time and frequency dimensions of the features.
This approach is computationally expensive and for our work we limit the augmentation process to the time dimension (time warping) alone.

3.5 CTC Loss

We use the Connectionist Temporal Classification (CTC) loss [28] for our Maximum Likelihood training. Each child network in the search space is trained to minimize the CTC objective function described below; given a target transcript $\boldsymbol{y^*}$ and an input $\boldsymbol{x}$ without any prior alignment.

$$CTC(\boldsymbol{x}) = -\log \Pr\left(\boldsymbol{y^*}|\boldsymbol{x}\right)$$

This likelihood of the label sequence is the sum of probabilities of all CTC paths, $\boldsymbol{q}$

$$\Pr\left(\boldsymbol{y^*}|\boldsymbol{x}\right) = \sum_{\boldsymbol{q}\in\mathcal{B}^{-1}(\boldsymbol{y})} \Pr\left(\boldsymbol{q}|\boldsymbol{x}\right)$$

where $\mathcal{B}$ is an operator that removes occurrences of repeated labels and blanks during alignment.
Monte-Carlo sampling is often used to estimate the expected loss function $\mathcal{L}$ and its gradient;

$$\mathcal{L}\left(\boldsymbol{x}\right) \approx \frac{1}{N}\sum_{i=1}^{N} \mathcal{L}\left(\boldsymbol{x}, \mathcal{B}\left(\boldsymbol{q}^i\right)\right), \boldsymbol{q}^i \sim \Pr\left(\boldsymbol{q}|\boldsymbol{x}\right)$$

$$\frac{\partial \mathcal{L}\left(\boldsymbol{x}\right)}{\partial \Pr\left(k, t|\boldsymbol{x}\right)} \approx \frac{1}{N}\sum_{i=1}^{N} \mathcal{L}\left(\boldsymbol{x}, \mathcal{B}\left(\boldsymbol{q}^{i,t,k}\right)\right)$$

where k, t and i are constants.

3.6 Model Objective

For our experiments, we train our controller RNN to optimize the joint objective function comprising of the CTC component and the expected reward over every iterative step as stated below,

$$\mathcal{J}\left(\boldsymbol{x}, \theta_c\right) = -\log \Pr\left(\boldsymbol{y^*}|\boldsymbol{x}\right) + \gamma \mathcal{Q}\left(\mathcal{R}|_{\theta_c, \pi_c}\right)$$

where $\mathcal{Q}\left(\mathcal{R}|_{\theta_c, \pi_c}\right)$ represents the expected reward received by the controller for every episodic execution of the policy π_c; θ_c represents the parameters of the RNN controller; γ is a weighting factor that regularizes the reward function. The optimization is carried out with a REINFORCE [29] algorithm described below:

Algorithm 1: REINFORCE on CTC loss

Input: initial controller parameters θ_0, initial (randomized) policy parameters, π_0
Output: optimal value of controller's policy weight θ_c

```
for each child network do
    for each episode τ <= S do
        compute J(x, θ_c)
        compute ∇J(x, θ_c)
    end
    θ_c ← θ_c + α∇J(x, θ_c)
end
```


4 Experiments

Here, we discuss in detail, how we set up our experiments and juxtapose results obtained by our methods with other top-notch methods declared in other publications.

4.1 Dataset

For our experiments, we use the popular LibriSpeech 960h [30] and TIMIT corpus [31] to train and evaluate our models.
LibriSpeech corpus defines a training set which is split into three (100h, 360h, 500h), two dev sets and two test sets derived from audiobooks in the LibriVox project [2].
TIMIT corpus is phonetic speech corpus collected by the Texas Instrument and Massachusetts Institute of Technology as part of the DARPA Speech Recognition research project in 1984, with the goal of providing succinct amount of phonetic speech data for evaluating automatic speech recognition systems. It consists of 6300 sentences from 630 speakers, about

2. https://librivox.org/

5 hours of audio data. It comes with a total of 60 acoustic phonetic labels used in transcribing the audio data.
The audio data was further processed in frames 10ms each with 20ms window size, using 16 Mel-Frequency Cepstrum Coefficients (MFCCs) from 32 filter-bank channels using discrete Fourier transform (DFT) filtering.
The large amount of publicly available transcribed corpus of speech is a good indication of how much more progress can be made in end-to-end methods for automatic speech recognition.

4.2 Experimental Setup

We trained on a cluster of 8 NVIDIA GPUs on a time span of 11 hours; we used a bidirectional RNN with 512 hidden layers to generate a list of architectural hyper-parameters and trained it to minimize WER and the CTC loss with REINFORCE as shown in Algorithm 1. For every iteration we limit the components predicted by the controller RNN to number of filters, filter height, filter width, stride height, stride width, presence or absence of: a maxpool operator, a batchNorm block and the RNN block. We train using Adam [32] as our optimizer, with parameters $\alpha = 0.001$, $\beta_1 = 0.9$, β_2=0.999, $\epsilon = 10^-8$ and ReLU [33] as the default activation function. We set a limit on the training process to explore only 1024 child networks in the entire search space.

4.3 Results and Discussions

Tables 2 and 3 summarize results from our methods on the LibriSpeech and TIMIT corpora respectively and also how they compare with other state-of-the-art results. Training the controller RNN to discover neural architectures gave good results without a language model and we obtained better performance from the addition of a 7-gram language model on the two training sets. A number of transcripts generated by our methods are highlighted in appendix A. It is obvious that not only does the model easily predicts short utterances, it performs brilliantly well on long speech sequences as well.

5 Conclusion

We have shown that novel deep neural network architectures discovered with neural architecture search in an ASR pipeline can improve the performance of LVSR systems. With policy gradient augmented with CTC loss, our agent explored 1024 child networks in 11 hours of computation improved based on performance on the development set. We obtained results that are on par with the state-of-the-art when evaluated on the LibriSpeech and TIMIT test data.

Model	LibriSpeech Test Clean	LibriSpeech Test Other
PyTorch-Kaldi [34]	6.2	
Baidu's D.S2 [35]	5.33	13.25
Espnet [36] + LM	4.6	13.7
Our own + no LM	**10.14**	**12.61**
Our own + 3-gram LM	**8.23**	**9.45**
Our own + 5-gram LM	**8.49**	**11.17**
Our own + 7-gram LM	**7.11**	**10.22**

Table 1: Performance achieved by our method on the LibriSpeech Test set

Model	TIMIT Dev	TIMIT Test
LAS multitask [37]		20.4
QCNN-10L-256FM [38]		19.64
RNN Transducer [39]		17.7
Attention based with conv nets [17]	15.8	17.6
RNN + Dropout + BatchNorm [34]		15.9
Our own proposed + no LM	**14.9**	**18.4**
Our own proposed + 3-gram LM	**14.4**	**17.3**
Our own proposed + 5-gram LM	**18.7**	**15.2**
Our own proposed + 7-gram LM	**11.1**	**13.4**

Table 2: Performance achieved by our method on the TIMIT dev and Test set

References

[1] Y. LeCun, L. Bottou, Y. Bengio, P. Haffner, Gradient-based learning applied to document recognition. In *Proceedings of IEEE*, 1998.
[2] D. Bahdanau, J. Chorowski, D. Serdyuk, P. Brakel, and Y. Bengio, End-to-end attention-based large vocabulary speech recognition. In *https:arxiv.com/pdf/1508.04395*, 2015.
[3] O. Abdel-Hamid, A.-r. Mohamed, H. Jang, and G. Penn, Applying convolutional neural networks concepts to hybrid nn-hmm model for speech recognition. In *ICASSP*, 2012.
[4] Y. Bengio, J. Louradour, R. Collobert, and J. Weston, Curriculum learning. In *ICML* 2009.
[5] H. Bourlard and N. Morgan, Connectionist Speech Recognition: A Hybrid Approach. *Kluwer Academic Publishers*, Norwell, MA, 1993.
[6] V. Mnih, K. Kavukcuoglu, D. Silver, A. Graves, I. Antonoglou. D. Wierstra and M. Riedmiller, Playing Atari with Deep Reinforcement Learning. In *NIPS*, 2013.
[7] A. Krizhevsky, I. Sutskever, G. E. Hinton, ImageNet Classification with Deep Convolutional Neural Networks. In *NIPS*, 2012.
[8] K. Simonyan, A. Zisserman. Very Deep Convolutional Networks for Large-Scale Image Recognition. In *CoRR*, 2014.
[9] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, A. Rabinovich. Going Deeper with Convolutions. In *CVPR*, 2014.
[10] D. Chen, C. D. Manning. A Fast and Accurate Dependency Parser using Neural Networks. In *EMNLP*, 2014.

International Journal of Scientific & Engineering Research Volume 10, Issue 11, November-2019
ISSN 2229-5518

[11] C. A. Huang, A. Vaswani, J. Uszkoreit, N. Shazeer, I. Simon, C. Hawthorne, A. M. Dai, M. D. Hoffman, M. Dinculescu, D. Eck. Music Transformer: Generating Music with Long-Term Structure. In *ICLR*, 2018.

[12] D. Bahadanu, K. Cho, Y. Bengio. Neural machine translation by jointly learning to align and translate. In *ICLR*, 2015.

[13] R. Pascanu, T. Mikolov, Y. Bengio. On the difficulty of training recurrent neural networks. In Proceedings of *MLR* press, 2013.

[14] J. Bergstra, Y. Bengio. Random Search for Hyper-Parameter Optimization. In *Journal of Machine Learning Research* 13, 2012.

[15] J. Bergstra, D. Yamis, D. D. Cox. Making a science of model search: Hyperparameter optimization in hundreds of dimensions for vision architectures, *ICML*, 2013

[16] L. Rabiner. A tutorial on Hidden Markov Models and Selected Applications in Speech Recognition. In *Proceedings of IEEE*, 1989.

[17] J. Chorowski, D. Bahdanau, D. Serdyuk, K. Cho, P. Brakel, and Y. Bengio. Attention-Based Models for Speech Recognition. In *http://arxiv.org/abs/1506.07503*, 2015.

[18] Y. Zhang, K. Sohn, R. Villegas, G. Pan, H. Lee, Improving Object Detection with Deep Convolutional Networks via Bayesian Optimization and Structured Prediction. In *ICVPR*, 2016.

[19] D. Yogatama L. Kong, N.A Smith, Bayesian Optimization of Text Representations. In *CEMNLP*, 2015.

[20] B. Zoph, Q.V. Le. Neural Architecture Search with Reinforcement Learning. In *ICLR*, 2017.

[21] B. Zoph, V. Vasudevan, J. Shlens, Q.V. Le. Learning Transferable Architectures for Scalable Image Recognition, arxiv preprint *arxiv:1707.07012*, 2017.

[22] H. Pham, M. Y. Guan, B. Zoph, Q.V. Le, J. Dean. Efficient Neural Architecture Search via parameter sharing. In *CoRR* 2018.

[23] R. Luo, F. Tian, T. Qin, E. Chen, T. Liu, Neural Architecture Optimization. In *NIPS*, 2018.

[24] A. Coates, B. Huval, T. Wang, D. J. Wu, A. Y. Ng, Deep learning with COTS HPC systems. In *ICML*, 2013.

[25] S. Hochreiter and J. Shmidhuber. Long-short-term memory. *Neural Computation*, 1997.

[26] D. Liang, Z. Huang, Z. Lipton. Learning Noise-Invariant Representations for Robust Speech Recognition. *IEEE SLT* 2018.

[27] D. S. Park, W. Chan, Y. Zhang, C.Chiu, B. Zoph, E. D. Cubuk, Q. V. Le. SpecAugment: A Simple Data Augmentation Method for Automatic Speech Recognition. In *https://arxiv.org/pdf/1904.08779*, 2019.

[28] A. Graves, S. Fernandez, F. Gomez, J. Schmidhuber. Connectionist Temporal Classification: Labelling Unsegmented Sequence Data with Recurrent Neural Networks. In *ICML*, 2006.

[29] Ronald J. Williams. Simple statistical gradient-following algorithms for connectionist reinforcement learning. In *Machine Learning*, 1992.

[30] V. Panayotov, G. Chen, D. Povey, and S. Khudanpur, “Librispeech: An ASR corpus based on public domain audio books”. In *Proceedings of ICASSP*, 2015.

[31] J. Garfolo, L. Lamel, and W. Fisher, “TIMIT acoustic-phonetic continuous speech corpus LDC93S1”. *Philadelphia: Linguistic Data Consortium*, 1993.

[32] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[33] V. Nair and G.E Hinton, Rectified Linear Units Improve Restricted Boltzmann Machines. In *ICML*, 2010.

[34] M. Ravanelli, T. Parcollet, Y. Bengio. The PyTorch-Kaldi Speech Recognition Toolkit, 2019.

[35] D. Amodei, R. Anubhai, E. Battenberg, C. Case, J. Casper, B. Catanzaro, J. Chen, M. Chrzanowski, A. Coates, G. Diamos, et al., “Deep Speech 2: End-to- end speech recognition in English and Mandarin”, *arXiv preprint arXiv:1512.02595*, 2015.

[36] S. Watanabe, T. Hori, S. Karita, T. Hayashi, J. Nishitoba, Y. Unno et al., ESPnet: End-to-End Speech Processing Toolkit, *arxiv Prepint, arxiv:1804.0015*, 2018.

[37] I. Karaulov, D. Tkanov, Attention model for articulatory features detection. In *INTERSPEECH*, 2019.

[38] T. Parcollet, Y. Zhang, M. Morchid et al., Quaternion Convolutional Neural Networks for End-to-End Automatic Speech Recognition. In *INTERSPEECH*, 2018.

[39] A. Graves, Sequence Transduction with Recurrent Neural Networks. In *ICML*, 2012.

Appendix A. Comparison of Model Generated and Ground Truth Transcripts

OUTPUT	i happen to have m cal's box for tonight or there'd be no chance of our getting places
TARGET	i happen to have mac connell's box for tonight or there'd be no chanc e of our getting places
OUTPUT	he were words than before but with the same mysterious music in them
TARGET	fewer words than before but with the same mysterious music in them
OUTPUT	then she turned towards the quarter indicated and disappeared round _ laurel bushes
TARGET	then she turned towards the quarter indicated and disappeared round the laurel bushes
OUTPUT	she bit her lip and looked down at her hands which were clospd tightly in front of her
TARGET	she bit her lip and looked down at her hands which were clasped tightly in front of her
OUTPUT	it came from the wife of one of his father's former work men and was concerning her son whom she begged justling to recommend as candidate for some post in town that she wished him to fill
TARGET	it came from the wife of one of his father's former workmen and was concerning her son whom she begged jocelyn to recommend as candidate for some post in town that she wished him to fill
OUTPUT	at these blasphemous sounds the pillars of the sanctuary were shaken
TARGET	at these blasphemous sounds the pillars of the sanctuary were shaken
OUTPUT	he looked intently and inquiringly into his friend's eyes evidently trying in vain to find the answer to some question
TARGET	he looked intently and inquiringly into his friend's eyes evidently trying in vain to find the answer to some question
OUTPUT	if we are to have any mythology at all he seems to argue why object to adding to it the mythus of jesus
TARGET	if we are to have any mythology at all he seems to argue why object to adding to it the mythus of jesus
OUTPUT	only seven of the attendants remained in the emperor's chamber and there the two sovereigns conversed for an hour after which they moved to the hall where a splendid supper awaited them
TARGET	only seven of the attendants remained in the emperor's chamber and there the two sovereigns conversed for an hour after which they moved to the hall where a splendid supper awaited them
OUTPUT	he strongly opposed this as depreciating the shares but i had no intention of going alone into what was then considered a wild and dangerous country finally we compromised
TARGET	he strongly opposed this as depreciating the shares but i had no intention of going alone into what was then considered a wild and dangerous country finally we compromised

Table 3: selected transcripts of audio data from the test sets in the TIMIT and LibrSpeech corpora.