National Research University

Higher School of Economics

Nizhny Novgorod

Faculty of Informatics, Mathematics, and Computer Science

Project Proposal

**Research on Speech to Text Transformation Methods**

# Student: V.V. Kilyazov

Group: 15AMI

Scientific Supervisor: N.V. Karpov

# Nizhny Novgorod

# 2019

In this paper we focus on development of the Russian language speech recognition system based on DeepSpeech architecture. The system was trained on a custom speech dataset which was collected from YouTube. The language model was developed based on corpus of popular articles in Russian version of Wikipedia. The resulting system was tested on a dataset consisting of audio recordings of Russian literature recorded by more than 25 different speakers, which is known as voxforge.com dataset. The best WER demonstrated by our approach currently equals to 15.8% with language model and 27% without language model usage.

**Introduction**

*Background.* Over the last few years, computer science industry was developing rapidly. The problem of speech recognition is not an exception. Large and famous companies, for example, Facebook, Google and Yandex, have now achieved the level of 90-95% correct words recognized in speech, decreasing the gap with the human level to 2-5%. Nevertheless, none of these companies pays enough attention to the development of open-source technologies, keeping trained models under commercial secret, especially for non-popular languages.

With the development of technologies in the field of neural networks and the production of graphics accelerators, which have significantly increased the speed of parallel computing, projects have appeared using deep-learning neural networks (DNN). In such networks, in comparison with classical ones, the number of hidden layers is increased. The increase in the number of hidden layers made it possible to achieve significant progress in many tasks that were previously considered available only to humans.

Our speech recognition system is based on a project with open source code called Mozilla DeepSpeech. It was developed in order to recognize English speech using DNN. This project was launched in May 2016 and in November 2017 reached the lowest level of the Word Error Rate (WER) metric at 6.5% on the LibriSpeech-clean test dataset. The project is based on the work of a group of researchers from Baidu Research AI lab placed in Silicon Valley, USA.

*Problem Statement.*This paper describes the process of creating a Russian speech recognition system based on the DeepSpeech architecture using the recurrent DNN using training examples (audio and transcript) using the CTC method (Connectionist Temporal Classification). The principle of using CTC method for training models is also described. Methods for obtaining training datasets from various sources were also investigated and implemented, the neural network was trained on the obtained data with a variation of hyperparameters, and the accuracy of the resulting model was evaluated.

The speech recognition system should recognize speech fragments in Russian language in the clean speech dataset with an average length of 5-15 seconds with an average WER error of less than 30%.

*Professional Significance.*Despite significant progress and adaptation of the above methods in commercial products, in the field of speech recognition today there is a shortage of open source systems capable of achieving level close to human. Some companies, such as Yandex and Google, offer to use their good quality models for a fee. Our model will be the first open-source model for speech recognition in Russian language with high accuracy.

**Literature Review**

Let us consider the most important papers on the development of methods for the recognition of temporal sequences by using recurrent neural networks, leading to modern End-to-End speech recognition methods.

Neural network solutions for speech recognition are replacing proven implementations using hidden Markov models (Hidden Markov Models, HMM) originally proposed by Rabiner (1989). In 2006 Graves showed that solutions using HMM require more human intervention as opposed to more modern methods using neural networks. In addition, recently, solutions based on DNN began to exceed the HMM in recognition accuracy. Recently, implementations using neural networks from the beginning to the end (End-to-End) are especially popular. Such solutions minimize the participation of a developer in model training, requiring him only to determine the neural network architecture and to prepare data for training.

Recurrent neural networks are used in data sequence recognition problems. This is described in research by Graves et al (2009). According to Salehinejad et al. (2017), RNN allows you to use information about previously predicted sequence elements to predict the next. For example, in a speech recognition task, the use of symbols predicted several steps backwards helps the neural network to determine a more suitable symbol for the current sound frame in the context of already predicted symbols.

The vanishing gradient problem is complex and especially critical for RNN, since the number of layers depends on the length of the sequence and the memory effect of past predictions in basic RNN disappears after a few time steps, which prevents the use of a more general context for prediction. Several solutions have been proposed for the vanishing gradient problem: use of the activation function ReLU, GRU (Gated Recurrent Unit), LSTM (Long Short-Term Memory), firstly discovered by Hochreiter and Schmidhuber in 1997. Method of using LSTM memory cells is the most successful today and is most common for using in sequence to sequence tasks.

The LSTM approach replaces simple activation functions in a neuron (such as a logistic function or a hyperbolic tangent) with memory cells that can store an analog value. Each such memory cell has gates on the input and output, which control whether the input signal can change the internal memory and whether the internal memory can affect the output result, respectively.

There is also a forget gate, which controls the decreasing of the value stored in the cell's memory. Thus, while the input and forgetting shutter are closed, the value of the gradient stored in memory remains unchanged in time.

According to Graves (2009), LSTM was actively used after success in recognizing handwriting text sequences. After this, this approach was successfully applied in machine translation of the text, handwriting generation, described by Graves (2013), text description of images, predicting the output of simple computer programs, speech recognition (Graves and Jaitly, 2014) and other areas related to processing sequences.

The Connectionist Temporal Classification (CTC) approach proposed by Alex Graves in 2006 was a breakthrough approach based on which many modern End-to-End speech recognition systems are built. His paper describes the type of output layer of the neural network, as well as the loss function which is calculated based on the values ​​of this layer. CTC does not limit the choice of network architecture. The approach developed by Alex Graves in 2012 solves the problem of the need for accurate data labeling for sequence recognition problems. In the field of speech recognition, this problem consists in comparing the sequence of audio frames and the sequence of letters — a set of audio frames can correspond to one letter. The CTC method also offers a differentiated loss function called CTC Loss, applicable for learning by the method of stochastic gradient descent.

The DeepSpeech architecture is an example of an End-to-End DNN system, which implies a simpler process for preparing data for training, as well as minimal adjustment of parameters. This architecture includes several hidden layers, one of which is recurrent with LSTM cells, others use the activation function ReLU. CTC Loss is used as a loss function. The architecture is optimized for learning on multiple graphics accelerators. In addition, a language model capable of evaluating sequences of words is used to correct predictions. The paper claims an achievement of 16% WER in a full test subset of the Switchboard Hub 5’00 dataset, consisting of 40 telephone conversations in English.

Further development of the DeepSpeech architecture is DeepSpeech 2. This work optimizes the architecture for use in industrial speech recognition. It also describes minor changes that allowed to transfer the system from learning English to learning North Chinese (Mandarin) speech.

Article by Sriram et al. (2017) describes the Cold Fusion architecture based on the Sequence to Sequence approach with attention layer. In this papeFr, it was proposed to use the language model not only for decoding the result of a neural network, but also in the learning process. The paper claims the achievement of 27.5% WER on its own dataset.

Based on the reviewed papers, it can be concluded that the most actively used and promising solutions for End-to-End speech recognition at the moment are solutions using CTC approach.

**Methods**

Since the implementation of the task involves the use of deep neural networks for speech recognition, the hardware requirements for the implementation are quite high. Graphic accelerators are used to train neural networks with several internal layers with thousands of neurons. It is also a necessary condition to have enough disk space on the computer used for storing gigabytes of data to train the neural network.

Based on the above requirements, for the implementation of the task we used computing cluster of Intel company, which was provided to us for research.

To implement the speech recognition system in this paper, the existing deep neural network architecture for speech recognition Mozilla DeepSpeech was chosen. The implementation of this architecture is available as an open-source code on GitHub and was developed based on an article (Hannun A. et al, 2014). To describe the structure of the neural network and optimize computations of the Mozilla DeepSpeech project, we used TensorFlow machine learning framework. This framework allows you to «freeze» a trained graph of a neural network and use the model on other devices (including mobile). Nowadays, the Mozilla DeepSpeech project is the most popular and active among open source speech recognition projects according to GitHub. This project aims to recognize English speech and has achieved at the moment the result below 10% WER. The Mozilla DeepSpeech project is also not tied to a specific language and has the ability to adapt to the Russian language.

Deep neural network learning process requires 500-2000 hours of speech with corresponding transcripts. Several sources of data from the Russian language were considered:

1. Voxforge.org dataset

There is a ready-made set of transcribed speech data on voxforge.org, created by a community of users who recorded fragments from Russian literary, read by their voice. The total amount of this data is about 26 hours of speech. This source, unlike the second one, provides a ready-made set of data.

1. Own-collected dataset from YouTube

Another source of transcribed speech was YouTube's video hosting site, where the video has subtitles. Two types of subtitles are provided: human-created subtitles and subtitles automatically recognized by Google’s speech recognition algorithms. Both types have their pros and cons. Manual subtitles have greater accuracy in the sense of the transcript, but less accuracy in the correlation of text and audio in time. Also, the advantage of automatic subtitles is a greater number of videos that have these subtitles. Automatic subtitles are available in almost any video where speech can be recognized. YouTube has hosted millions of videos with Russian speech recorded by various voices, using a variety of recording devices and in a variety of noise conditions. Training on such a variety of data with sufficient quality of transcripts should have a positive effect on the adaptation of the neural network to different recording conditions and accents. Due to the amount of information available and the diversity of speech, YouTube’s video hosting will be used as the main source of speech data.

**Achieved Results**

The experiments were conducted using a voxforge dataset. This test dataset consists of 11.5 hours of clean literary works recordings.

Training of the neural network was carried out up to the moment when, over several epochs, the loss metric of CTC Loss ceases to decrease or begins to increase.

As shown by tests, the minimum error values of 27% WER and 10.6% CER were achieved. A ten percent improvement in WER on the voxforge dataset using the language model was obtained. A slight increase in the CER is observed when using the language model. Even though, based on the CTC Loss error rate, overfitting began in the second epoch, the WER index continued to decline to the 14th epoch, even with an increase in the CTC Loss value, which can be explained by the difference in the CTC Loss and calculation of WER method. In addition, the optimal number of neurons in the hidden layers of the neural network was determined: 2048 per layer.

We also managed to confirm that the DeepSpeech architecture also has a minimum number of bindings to a specific language (only the alphabet setting), which allows the basic learning process to be used when creating speech recognition systems for other languages.

**Conclusion**

We considered the principle of speech recognition with the use of deep neural networks based on the DeepSpeech architecture in that paper. The methods of automatic data collection for creating transcribed speech corpuses were investigated. Test results showed that accuracy of our speech recognition system is quite good: it is showing a result of 15.8% WER on a dataset with a clean speech (voxforge) using language model. We also showed that DeepSpeech system could work quite accurate for other languages that differs from English and the size of the language alphabet in this case doesn’t matter.

As prospects for further work, we can consider the following options: Increase recognition speed by parallelizing the beam search algorithm or transferring calculations to a graphics accelerator, studying of the possibility of using DeepSpeech architecture for recognition on less powerful computers (such as mobile or IoT devices).

**References**

Graves, A., Liwicki, M. (2006). *Connectionist temporal classification: labelling unsegmented sequence data with recurrent neural networks*. Proceedings of the 23rd international conference on Machine learning.

Graves, A., Fernandez, S. (2009). *A novel connectionist system for unconstrained handwriting recognition*. IEEE transactions on pattern analysis and machine Intelligence.

Graves, A. (2012). *Supervised sequence labelling with recurrent neural networks.* Springer.

Graves, A. (2013). *Generating sequences with recurrent neural networks*. arXiv.

Graves, A., Jaitly, N. (2014). *Towards end-to-end speech recognition with recurrent neural networks*. International Conference on Machine Learning.

Hannun, A., Case, C. (2014). *Deep speech: Scaling up end-to-end speech recognition*. arXiv.

Hochreiter, S., Schmidhuber, J. (1997). *Long short-term memory*. Neural Computation.

Rabiner, L. (1989). *A tutorial on hidden Markov models and selected applications in speech recognition*. Proceedings of the IEEE.

Salehinejad, H., Sankar, S. (2017). *Recent Advances in Recurrent Neural Networks.* arXiv.

Sriram, A., Satheesh, S. (2017). *Cold fusion: Training seq2seq models together with language models*. arXiv.