Using Google TensorFlow Machine Learning Library for speech recognition

M S Medvedev and Y V Okuntsev
Institute of Space and Information Technology, Siberian Federal University, 26 Kirenskogo street, Krasnoyarsk, 660074, Russia

E-mail: 2149972@mail.ru

Abstract. This work is devoted to the development and study of the procedural processing of speech signals using recurrent neural networks. Our method for speech recognition was a connectionist temporal classification based on networks of long short-term memory. The main goal of this work was to study the specifics of the Russian language, to develop methods and algorithms for converting oral speech into text based on artificial neural networks. The final recognition coefficient is about 62%.

1. Introduction
Controlling the device through voice control greatly expands the possibilities for work and life. To teach a machine to recognize and understand human language is the task of speech recognition. Machine speech recognition systems are widely used in many areas of human life. User-friendliness can be singled out as the main advantage of such systems, because voice commands allow to control the device without any special tools. Thanks to the widespread dissemination and development of technologies related to speech recognition, features such as voice control, voice text input and voice search have become possible. In addition, using speech recognition technology, special services are created that allow people with disabilities to use a machine: use voice input to control household appliances and electronic robots, a navigation system in the car, etc.

2. Connectionist temporal classification
Neural networks in solving the problem of speech recognition are usually trained using separate fragments of sound speech recordings. For this, it is necessary to allocate separate labels for each frame, which entails the need to align the sound track and the corresponding transcription [1]. However, such alignment will be reliable only when the neural network is already trained, which leads to a cyclical relationship between segmentation and recognition, also known as the Cyrus paradox. Moreover, in speech recognition tasks based only on transcription of words, alignment does not bring any benefit at all [2].

The most common solution in this situation is to use the Connectionist Temporal Classification (CTC). Connectivity time classification is a function that allows recurrent neural networks to learn to recognize a sequence of words without initially aligning the input and output sequences. [3]

At the stage of training a neural network, in which the CTC function is used as a loss function, the output layer of the neural network contains one block for each symbol of the output sequence (in our case, letters of the Russian language) and one more for the additional symbol “skip” (“space” for our
specific task), corresponding to silence on the sound track. The output vector is normalized using the softmax function.

3. Speech database
The training data set is a collection of a manifest.csv text file containing the address of the audio file, the corresponding transcription, as well as information about the file size, which is then used to sort the examples when training a neural network. Audio files in wav format contain fragments of speech; and for each file there is a text document with the transcription in it. Audio files in wav format have a single sound channel (mono), a sampling frequency of 16000 Hz; and they are encoded with a width of 2 bytes for each value.

Also, the decoding algorithm for CTC matrix implies a restriction on the ratio of the length of the audio and the number of characters in the transcript. A number of steps in CTC matrix must be greater than a number of characters in the transcript.

Table 1. Example contents of manifest.csv file.

| Wav_filename         | Transcript_filename | filesize |
|----------------------|---------------------|----------|
| ./data/train/ru_01.wav| ./data/train/ru_01.txt | 220044   |
| ./data/train/ru_02.wav| ./data/train/ru_02.txt | 252044   |

4. The principle of work
At the input of the speech recognition module a speech signal from a microphone or a previously recorded audio file of wav format is received. It should be noted that the sound recorded from the microphone does not immediately enter the recognition module, but is previously stored in a separate audio file in the appropriate format for further recognition.

The acoustic component converts the signal to a digital form with a sampling frequency parameter of 16000 Hz. The features extraction of a speech signal is carried out in several stages:

- the first step is to split the entire signal into short frames of 10 ms length and apply the Hann (Henning) window function to them; using the fast Fourier transform, we obtain the signal spectrum for each frame [4];
- the next step will be the calculation of the cepstrum and the imposition of chalk-frequency windows on it; energy is calculated for each frame;
- the selection of the features of the speech signal by the discrete cosine transform function is completed, resulting in the values of the cepstral coefficients of small frequencies [5].

This set of coefficients is the input to a neural network, the subsequent operation of which depends on the choice of the operating mode. When training a neural network, the input data is transmitted to the training unit of the neural network, where direct training takes place. After completing the training, the neural network retains its progress, and the module ends the work. When the neural network is in recognition mode, the data goes to the connection time classification block, where the speech signal is decrypted and the result is displayed to the user in text form.

5. Neural network design
It is known that training of neural networks is focused on the following objects [6]:

- layers that are combined into a common network;
- data for training a neural network;
- loss function, which determines the feedback signal used for training;
- an optimizer that determines how the training goes.
The relationship of these objects can be represented, as it is shown in figure 1. The network consists of two layers of a recurrent network with long short-term memory, which are combined in a chain and map the data into predictions. Then the loss function, which in our case is played by the loss function for the connection time classification, compares the predictions with the goals and returns the value of the loss: a measure of the correspondence of the prediction made by the network to the expected result. The optimizer uses this loss value to change the network weights. The Adaptive Moment Estimation is used as an optimization function for a speech recognition module. This algorithm for gradient optimization of first-order stochastic objective functions is based on adaptive low-order estimates. The method is easy to implement, computationally efficient, has a small memory requirement, is invariant to diagonal scaling of gradients, and is well suited for tasks that are large in terms of data. Empirical results show that Adaptive Moment Estimation works well in practice and wins in comparison with other stochastic optimization methods. [7]

6. Conclusion
A database of recorded speech with a total size of about 3 gigabytes was loaded into the system. The duration of one audio file in the “wav” format did not exceed 20 seconds, while the total duration of all files was about 17 hours of recording. For an average of 7.5 seconds of recording, there were 77 text characters. As a criterion for the quality of speech recognition, the values of the loss function were used. The loss function is a function that, in the theory of statistical decisions, characterizes losses due to incorrect decision-making based on observed data. As a loss function, the objective function was used to train the networks of connection time classification with gradient descent. The objective function in this case is derived from the principle of maximum probability, in other words, to minimize errors, the correction of the logarithmic probability is used to meet a certain letter. The final recognition coefficient is about 62%.
Figure 2. Loss function graph.

The graph visualizing the loss function was done using TensorBoard, which is TensorFlow's built-in machine learning library tool.

References
[1] Lawrence R R 1990 A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition (San Francisco: Morgan Kaufmann Publishers Inc)
[2] Vyugin V 2013 Mathematical foundations of the theory of machine learning and forecasting (ICMNO)
[3] Alex G et al. 2006 Connectionist Temporal Classification: Labelling Unsegmented Sequence Data with Recurrent Neural Networks (ICML)
[4] Afonsky A A and Dyakonov V P 2009 Digital spectrum analyzers, signals and logic (Moscow: SOLON-Press)
[5] Zakovryashin A S, Malinin P V and Lependin A A 2019 The use of distributions of small-frequency cepstral coefficients for voice identification of a person Bulletin of Altai State University 3(107) 156-8
[6] Geron O 2018 Applied Machine Learning with Scikit-Learn and Tensorflow: Concepts, Tools, and Techniques for Creating Intelligent Systems (St. Petersburg: LLC Alpha-book)
[7] Diederik K and Jimmy B 2015 Adam: A Method for Stochastic Optimization / K. Diederik, B. Jimmy 3rd International Conference for Learning Representations, San Diego, 2015