UDC 004.934.2

USING THE CTC-BASED APPROACH OF THE END-TO-END MODEL IN SPEECH RECOGNITION

Ochilov M.M. [1]
[1] Tashkent University of information technologies named after Muhammad al-Khwarizmi, Tashkent, Uzbekistan
ochilov.mannon@mail.ru

Abstract. *This paper has been discussed the CTC-based approach of the E2E model is used in speech recognition. The article discusses the steps of speech recognition based on the CTC approach. It also reviews the types of problems encountered in speech recognition and possible solutions when usinga CTC-based approach.*

Keywords: *End-to-End , CTC-based , CNN, RNN, BRNN, CTC-Decoding.*

I. INTRODUCTION

However, the creation of a speech recognition system is a complex process that requires a lot of professional knowledge. In recent years, various attempts have been made to reduce the complexity of ASR in the hope of directly displaying tagged speech. End-to-end speech recognition is proposed. The end-to-end technology has been applied in many ways and has achieved remarkable results.

End-to-end is a system that directly maps a sequence of input acoustic features into a sequence of phoneme or words. A system that is trained to optimize criteria that are related to the final evaluation metric that we are interested in (typically, word error rate).

CTC was proposed by Graves et al. [1] as a way to train an acoustic model without the need for frame-level alignment. CTC allows you to train an acoustic model without the need for frame-level alignment between acoustics and transcripts. CTC is superior to other types of approaches in decoding speed. But decoding accuracy remains directly dependent on the language and pronunciation model.

II. MAIN PART

When using the CTC-based approach of the End-To-End model in speech recognition, it is necessary to use linear, convolution, and recurrent neural networks from direct deep learning algorithms. Below are the steps to perform speech recognition based on this approach.

Step 1. Creating a property map from the unwanted property vector. Convolution neural network uses, in order to solve this problem. This architecture serves to highlight the most useful of the unwanted features. The working principle of the CNN network architecture is fully described in [4,6] literatures. There are several layers in architecture, each of them consists of its own filter core. The core of the convolution processes the previous layers along the fragment. The weight coefficients of the filter core are not predetermined (unknown) before, they are set during the training process. The scaling process is calculated using the following formula (1):

$$a_j^l = f\left(\sum_{i \in M_j} a_i^{l-1} * k_{ij}^l + b_j^l\right) \quad (1)$$

Here a is the output value of the input layer, k is the filter core, l is the number of output layers determined by the number of filter core, i is the pitch of the core movement at each calculation stage, M_j is the characteristic of the j card created on the basis of different nuclei, b is bias and f is the activation function in the formula (1). Usually a sigmoid or ReLU uses as an activation function.

A simple convolution network consists of several residual CNN layers that process input spectrogram images and output feature maps. Usually, 2D type CNN uses. We can see the location of the CNN layers in the network, the visual appearance of the input and output data in Figure 1:

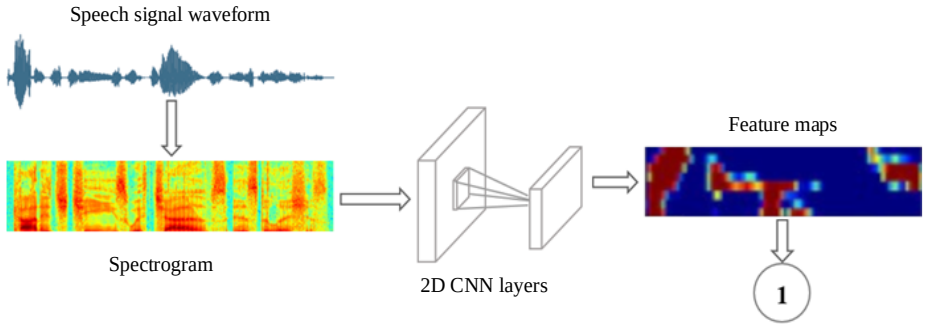

Fig. 1. Creating a feature map using CNN

Step 2. The problem of continuity (time dependence) of a point signal is solved by using recurrent neural networks that process feature maps that correspond to the desired sequence of output characters, a sequence of exact times, or as "frames".

The recursive neural network takes feature maps that are continuous images of speech and converts them into discrete values. If we describe this process visually, it is as follows.

Bidirectional recurent neural networks (BRNN, BLSTM) connect two hidden layers in opposite directions to the same output. Using this form of in-depth study, the output layer can simultaneously receive information from past (backward) and future (forward) states. BRNN and BLSTM use the pre- and post-context data for each time step in the input sequence to compute the output sequence (Figure 2). This network will have two separate hidden layers in the forward and backward position.

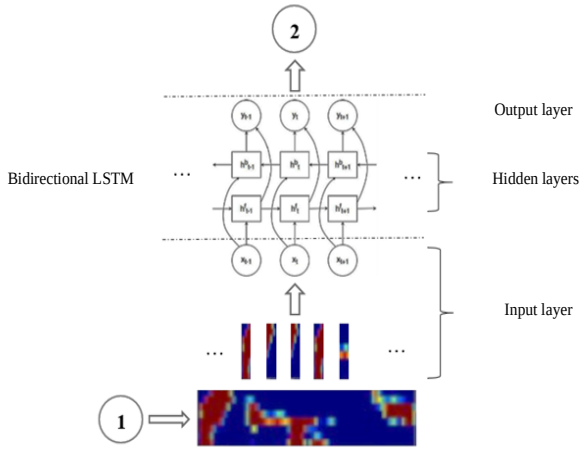

Fig. 2. Creating outputs over a recurrent network

They calculate the values of the forward layer util $t = 1...N$ the iteration and the hidden layer nodes $t = N...1$ the iteration. The actions performed in the reverse position are the same as in the forward position. In BRNN, forward and backward hidden cases are calculated and combined using the following equations:

$$h^{f_t} = \sigma\left(W_{\text{xh}^f} x_t + W_{h^f h^f} h_{t-1^f} + b_{h^f}\right)$$
$$h^{b_t} = \sigma\left(W_{\text{xh}^b} x_t + W_{h^b h^b} h_{t+1^b} + b_{h^b}\right)$$
$$y_t = \text{softmax}\left(h_t^f W_{h^f y} + h_t^b W_{h^b y} + b_y\right)$$

where h^{f_t} and h^{b_t} represent the position of the hidden forward and backward layers, respectively, while W_{h^f} and W_{h^b} represent the weights in the forward and reverse directions, respectively. The output is generated from the sum of the parameterized latent states at each time step.

Step 3. Fully connected neural networks use to produce the probabilities characteristic for each time of LSTM outputs. Neurons of optional size can be used in the hidden layers of a two-layer fully connected network.

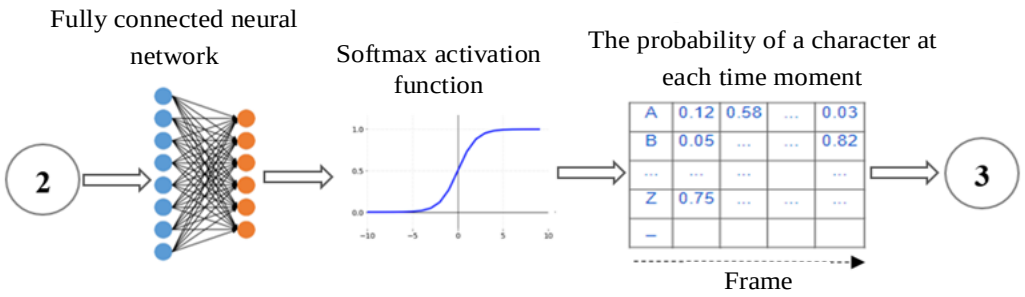

Fig. 3. Creating probability of characters

However, a neuron must be selected for the maximum number of characters for the existing language in the output layer (for example, in Uzbek language there are 29 outputs. 26 of them are alphabetical characters, three are special characters) [8].

The created model takes speech spectogram images and produces the characteristic probabilities for each time or “frame”.

Step 4. In each frame, the probability of all the possible output characters is quoted. The probability of meeting the characters at each moment of time can be seen in the following table (1).

In this step, CTC decoding operations are performed. That is, the most probable characters in each frame separates, one of the repeated characters will be left, and a special ‘-’ that separates the two characters will be removed. If the model defines the most probable sequence output at each time moment, for example, “- -ka-tt-t-o <b> <b> --uu-y”, where <b> is the space representation symbol.

In the decoding process, after the first operation, "-ka-t-t-o <b> -u-y", after the second operation, the output sequence "katto <b> uy" is formed. The correct label did not appear after decoding. This problem can be solved using the language model [5].

Table 1

The probability that characters will meet in each frame

a	0.0	0.0	0.0	0.8	0.0	0.0	0.0	0.0	0.0	0.0	0.4	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
b	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
c	0.0	0.0	0.0	0.0	0.0	0.0	0.1	0.0	0.0	0.0	0.0	0.1	0.0	0.0	0.0	0.0	0.0	0.2	0.0	0.0
d	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
e	0.0	0.0	0.1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.1	0.0
f	0.0	0.1	0.0	0.2	0.0	0.2	0.0	0.0	0.0	0.0	0.0	0.0	0.1	0.0	0.2	0.0	0.1	0.0	0.0	0.0
g	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
h	0.1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
i	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
j	0.0	0.0	0.0	0.0	0.0	0.0	0.1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
k	0.0	0.0	0.9	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
l	0.0	0.0	0.0	0.0	0.1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
m	0.1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.1	0.0	0.0	0.0	0.0
n	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.1	0.0	0.0
o	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.6	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
p	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
q	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
r	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.1	0.0	0.0	0.0	0.0
s	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
t	0.0	0.0	0.0	0.0	0.0	0.8	0.7	0.0	0.6	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
u	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.1	0.0	0.0	0.0	0.0	0.0	0.7	0.9	0.0	0.0	0.0
v	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.1	0.1	0.0	0.0	0.0	0.0	0.1	0.0
x	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
y	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.8	0.0
w	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.2	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
z	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
‘	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
<b>	0.9	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.9	0.8	0.0	0.0	0.1	0.0	0.0	0.0	0.0
>	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.6	0.2	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0
-	0.8	0.9	0.0	0.0	0.9	0.0	0.1	0.9	0.0	0.7	0.0	0.0	0.0	0.9	0.8	0.0	0.0	0.7	0.0	0.0
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20

We can list the overall execution sequence of this step as follows:

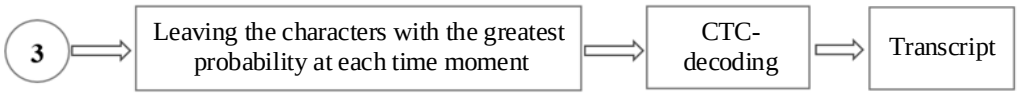

Fig. 4. Decoding sequence

The CTC layer is used in the training process as well as in the decoding function. This process is a bit difficult. We will try to understand it below. First, let’s look at how CTC can help solve problems.

The complexity of choosing a single unit of frame length(or the difficulty of setting frame boundaries) in order to equate the length of the incoming data to the length of the output values.

In speech recognition systems, incoming information is longer than output. The main task of the CTC is to maximize the input and output sequence. If there are no clear limits for comparing the input values with the values of the output sequence, the difficulty of calculating the error arises. This problem can be solved using the CTC Loss function [2,3,7].

The unique feature of CTC is that it performs this automatically. It calculates

the probability error of the network to predict the correct label sequence. To do this, the network looks at all the predictable sequences and separates them into subgroups that correspond to the target transcript. The implementation of this process can be illustrated in Figure 4. To make it easier to understand from the picture, we select the small-time moments (number of frames). For example, if the phrase “katta uy” is a target tag, we’ll look at how CTC chooses paths based on the number of frames as 11.

We can see this from the table above. It is possible to re-adjust the network parameters during training on the basis of approaching 1 the probability that the signs laying on the possible roads will meet at the same time moment. It should be noted that in the teaching of the network are used sentences of different lengths, consisting of different characters. This makes it difficult for the CTC to select possible mitigation pathways during training.

The estimated paths should be the same as the target label after the CTC decoding operations. The CTC knows exactly the lower and upper bounds of the paths that can be pushed to the target label when selecting paths.

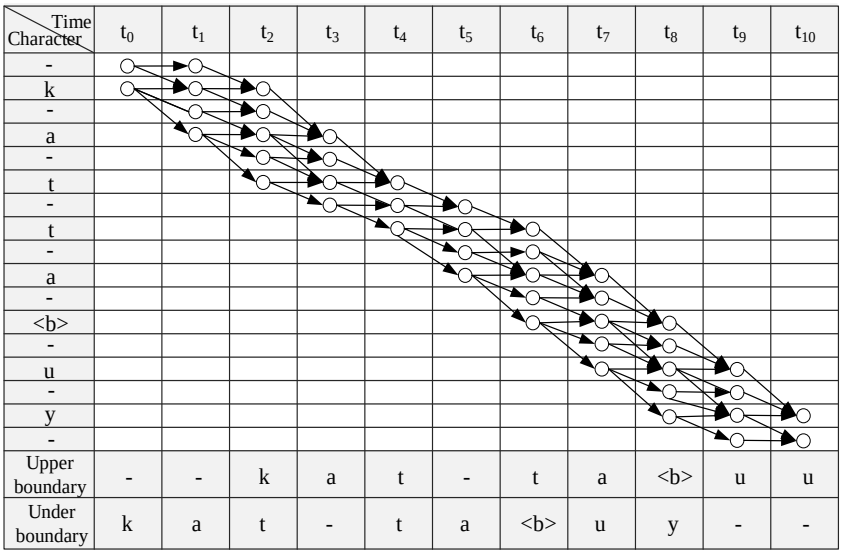

Fig. 5. Shorten the paths in CTC decoding

The CTC therefore selects paths that can be customized to the target label for each unwanted information. To do this, only the sequence of characters available on the target label is added. Characters that do not occur on the label are not considered.

CTC Loss (error) is calculated during network training. Proper operation of the network is ensured by minimizing the error. The calculation of the CTC error is given in formula 2.

The CTC error for the unwanted $\bar{x} = x_1, \dots, x_T$ acoustic properties and the target word transcript $\bar{\omega} = \omega_1, \dots, \omega_M$ is described as follows.

$$L_{CTC}(\theta, \bar{\omega}, \bar{x}) := -\log\Big(\sum_{\bar{s} \epsilon S_{\bar{\omega}}} p_\theta(\bar{s}|\bar{x})\Big) = -\log\Big(\sum_{\bar{s} \epsilon S_{\bar{\omega}}} \prod_{t=1}^{T} p_\theta(\bar{s}_t|\bar{x}_t)\Big), \quad (2)$$

Here are the θ model parameters. Paths that can be bridged with S^w–W. It includes all scaling paths allowed by CTC, such as $\overline{s} = s_1, \ldots, s_T$. Each step here can be a st class symbol or a blank character for t. The correct path in the CTC topology for a given configuration is the path that will be the same as the actual shortcut after all blank and sequential characters have been deleted.

III. CONCLUSION

In conclusion, CTC-based approach to speech recognition is more efficient than other types of E2E models in terms of simplicity and decoding speed. CTC is used as a basic acoustic model for modern hybrid speech recognition approaches.

Currently, speech recognition technology based on end-to-end has achieved remarkable results, but CTC-based end-to-end speech recognition still needs a language model to get better results.

REFERENCES

[1] *Alex Graves, Santiago Fernández, Faustino Gomez, and Jürgen Schmidhuber*. 2006. “Connectionist temporal classification: labelling unsegmented sequence data with recurrent neural networks”. In Proceedings of the 23rd international conference on Machine learning (ICML '06). Association for Computing Machinery, New York, NY, USA, 369–376.

[2] *Alex Graves, Navdeep Jaitly*. “Towards End-to-End Speech Recognition with Recurrent Neural Networks.”(ICML 2014).

[3] *Graves, A.* (2012). “Supervised Sequence Labelling with Recurrent Neural Networks”. Studies in Computational Intelligence.

[4] *Musaev M., Mussakhojayeva S., Khujayorov I., Khassanov Y., Ochilov M., Atakan Varol H.* (2021) USC: An Open-Source Uzbek Speech Corpus and Initial Speech Recognition Experiments. In: Karpov A., Potapova R. (eds) Speech and Computer. SPECOM 2021. Lecture Notes in Computer Science, vol 12997. Springer, Cham. https://doi.org/10.1007/978-3-030-87802-3_40.

[5] *Musaev, M., Khujayorov, I., Ochilov, M.:* Image approach to speech recognitionon CNN. In: Proc. of the International Symposium on Computer Science andIntelligent Control (ISCSIC). pp. 57:1–57:6. ACM (2019)21.

[6] *Musaev, M., Khujayorov, I., Ochilov, M.:* Development of integral model of speech-recognition system for Uzbek language. In: Proc. of the IEEE International Con-ference on Application of Information and Communication Technologies (AICT). pp. 1–6. IEEE (2020)22.

[7] *Musaev, M., Khujayorov, I., Ochilov, M.*: The use of neural networks to improvethe recognition accuracy of explosive and unvoiced phonemes in Uzbek language.In: Proc. of the Information Communication Technologies Conference (ICTC). pp. 231–234. IEEE (2020)23.

[8] *Musaev, M., Khujayorov, I., Ochilov, M.*: Automatic recognition of Uzbek speechbased on integrated neural networks. In: Proc. of the World Conference “Intelligent System for Industrial Automation”(WCIS-2020). pp. 215–223. Springer Interna-tional Publishing (2021).

Поступила в редакцию 23.12.2022

Citation: *Ochilov M.M.* 2023. Using the CTC-based approach of the end-to-end model in speech recognition. *International Journal of Theoretical and Applied Issues of Digital Technologies*. 1(3): 135-141.

ИСПОЛЬЗОВАНИЕ ПОДХОДА НА ОСНОВЕ CTC МОДЕЛИ END-TO-END ДЛЯ РАСПОЗНАВАНИЯ РЕЧИ

Очилов М.М.[1]
[1] Ташкентский университет информационных технологий имени Мухаммада ал-Хорезми, Ташкент, Узбекистан
ochilov.mannon@mail.ru

Аннотация. *В этой статье обсуждается подход основанный на CTC модели E2E, используемый в распознавании речи. В работе рассмотрены этапы распознавания речи на основе CTC-подхода. Также рассматриваются типы проблем, возникающих при распознавании речи, и возможные решения с использованием подхода на основе CTC.*

Ключевые слова: *End-to-End, подход CTC, CNN, RNN, BRNN, CTC – декодирование.*

NUTQNI ANIQLASHDA END-TO-END MODELINING CTC ASOSIDAGI YONDASHUVIDAN FOYDALANISH

Ochilov M.M.[1],
[1] Muhammad al Xorazmiy nomidagi Toshkent axborot texnologiyalari universiteti, Toshkent, O‘zbekiston
ochilov.mannon@mail.ru

Annotatsiya. *Ushbu maqolada nutqni aniqlashda ishlatiladigan E2E modelining CTC-ga asoslangan yondashuvi muhokama qilingan. Ishda CTC yondashuviga asoslangan nutqni aniqlash bosqichlari ko‘rib o‘tilgani. Shuningdek, nutqni aniqlashda duch keladigan muammolar turlarini va CTCga asoslangan yondashuvdan foydalanishda mumkin bo‘lgan echimlarni ko'rib chiqalgan.*

Kalit so‘zlar: *End-to-End, CTC-asosida, CNN, RNN, BRNN, CTC-dekodlash.*