ON THE LONG SHORT-TERM MEMORY IN THE RECURRENT NEURAL NETWORKS

O. Namicheishvili, Zh. Gogiashvili, M. Kiknadze, G. Acharadze

Georgian Technical University, Faculty of Informatics
and Control Systems
M. Kostava str. 77, Tbilisi 0171, Georgia
O.namicheishvili@gtu.ge, J.Gogiasvili @gtu.ge,
m.kiknadze@gtu.ge, guramiacharadze@gmail.com

(Received 20.05.2021; accepted 05.11.2021)

Abstract

Up to now, the recurrent neural networks have been trained by means of their unfolding in time and using the modified backpropagation through time method. It is turned out that in the teaching process with the availability of sufficiently long input sequences a network forgets information on remoted objects. In some cases, it becomes necessary for the network to "remember" information on the objects being in the beginning of the sequence.

The article deals with the examples of problems, which confirm the necessity of memory presence for the network.

Keywords and phrases: Recurrent neural network, long short-term memory network, forget gate, gate, forget layer.

Unfolding in time tool and the modified backpropagation through time method were used for training the recurrent neural networks. But it was turned out that input sequences are sufficiently long in the training process, and the network forgets information on remoted objects. In some cases, it becomes necessary for the network to "remember" information on the objects being in the beginning of the sequence.

Let's consider the examples, which confirm the necessity of memory presence for the network.

Human action recognition [**1**]. The problem is to determine based on the sequence of video frames (videos), what type of movement the human performs (sits, stands, marshes, runs, jumps etc.). It is evident that initial actions may be identical for different types of movements, that is why the information on the entire sequence of actions is necessary for making a decision.

Semantic segmentation of the video sequence [**2**]. The goal of semantic segmentation is to determine the class of that object, to which

every pixel of this state belongs. Video is a set of video frames associated to each other.

At the same time, regardless of shooting conditions the scene on the neighbor videos changes slowly enough. That is why in the current frame semantic segmentation process it is possible to use that information, which was obtained when segmenting the previous frame.

Image verbal description formation. The given problem is at the boundary of computer vision and natural language processing spheres.

The sense of the problem is that a harmonious (well-ordered) sentence has to be composed that describes the image content (Fig. 1). At each stage of description, takes place an attempt of next word restoration in the sentence based on image context. It is natural that there are the rules of competent construction of sentence, that's why appearance of a new word is largely determined not only by image context, but also by those words, which have been restored for a current moment.

General structure of the long short-term memory network. General structure of the cell with long short-term memory (LSTM) implies the existence of neuron, which is linked with its own.

Let's consider a general pattern of cell operation by the one neuron example. Data are delivered to neuron input and are taken off from the output. Recurrent connection has a weight equal to 1 at its input. Thus, if not any new data are delivered to input, then the neuron value is rewritten anew and remains unchanged.

1. A view from above of the night city street-lamps with a well-lit church square in the foreground;
2. Humans at the stairway in front of twin-tower temple illuminated in the night

Figure 1: Examples of image description formation [7]

Three gates (in other words, taps, windows) are used for controlling this structure and they determine signal passage: input gate, forget gate and output gate (Fig. 2).

The values of these gates are established by other network neurons,

when a corresponding need arises. Each gate can be in one out of two states: 1 – gate is open, 0 – gate is closed [3].

1. If the input gate is open (is set to **1**), then an input signal is recorded on hidden neuron, after which the value will be recorded and kept on neuron by means of recurrent feedback.

2. If the input gate is closed (is set to **0**) then the values delivered to neuron input have no impact on this neuron content.

3. If it is necessary to get the value kept in the cell, the output gate has to be opened (set to **1**).

4. If we need to "forget" the value placed in the cell, one the forget gate has to be opened. Hereupon this value will be deleted from neuron and neuron is ready to keep a new input value.

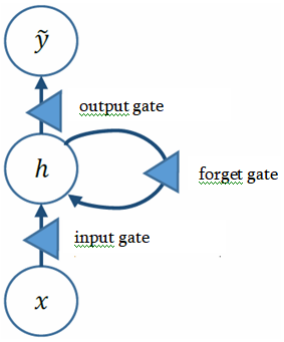

Figure 2: The principle of cell with long short-term memory

Let's consider the **long short-term memory network** implementation. Similar to common recurrent network, LSTM-networks have their own recurring block expanded in time [4]. Let's consider step-by-step below each element of the block (Fig. 3).

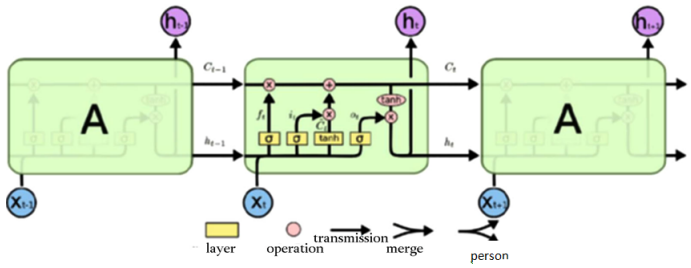

Figure 3: General structure of long short-term memory network [4]

The main component part of LSTM-cell is represented by its C_t **state**, which is transmitted through time. The cell can add information to the

state or to remove it from the state and thereby takes place a careful adjustment by the structures called the **gates**. Gate is the way of information transmission, it consists of sigmoid layer and element by element multiplication operation [4].

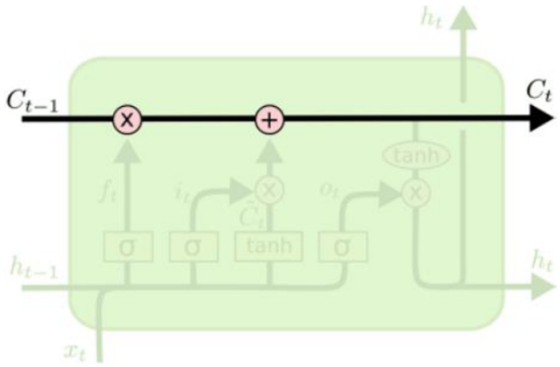

Figure 4: LSTM-cell state transition (transmission) [11]

At the first stage of cell operation it is necessary to take a decision, which state elements have to be "forgotten". For this purposes, we have to pass **"forget gate" layer**- sigmoid layer of the network (Fig. 5). Sigmoid layer has numbers from 0 to 1 at the output. In fact, it determines weights, by which the state elements' passing occurs. Value 0 means that there is no element passing, while **1** denotes complete passing of element.

At the ensuing step, one has to determine, what a new information has to be kept in the cell state. Implementation of this step includes two layers of the network. Sigmoid layer - **input skip layer** - takes decision regarding what values have to be renewed. The layer by means of activation function corresponding to hyperbolic tangent forms the vector of those new values - candidates, which are added to the current state (Fig. 6).

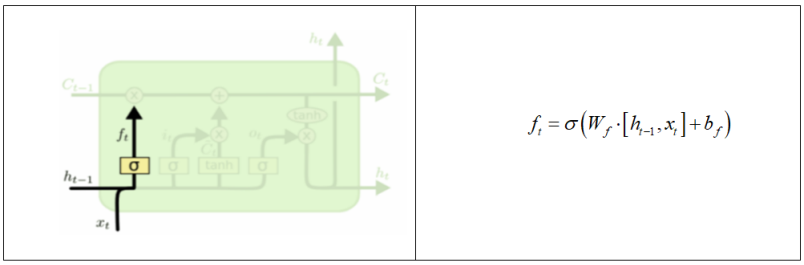

Figure 5: LSTM-cell "forget" layer [4]

Afterwards, a cell state updating remains. In fact, it is necessary to remove that information from the state vector, for which the decision is

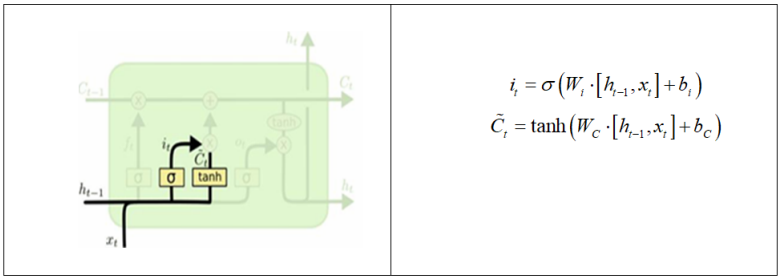

Figure 6: The layer for new information storage in LSTM-cell state [11]

taken that its "forgetting" and new information addition is possible.

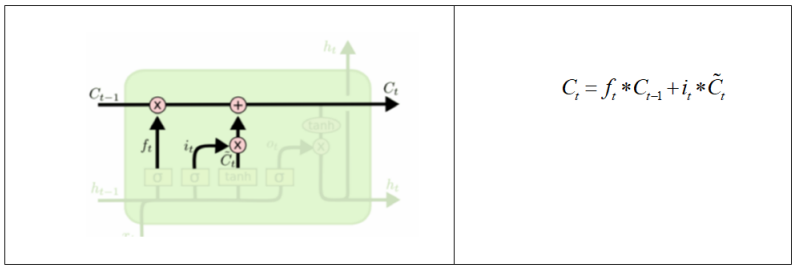

Figure 7: LSTM-cell state updating [4]

There remains to make a decision, what has to be taken as the cell output. The output is constructed based on cell state and represents its filtered version. First the decision is taken, what parts of the state vector have to be taken out in case of sigmoid layer existence. Afterwards takes place cell state elements norming within [-1; 1] segment, using hyperbolic tangent function and multiplication of these elements at the sigmoid layer output.

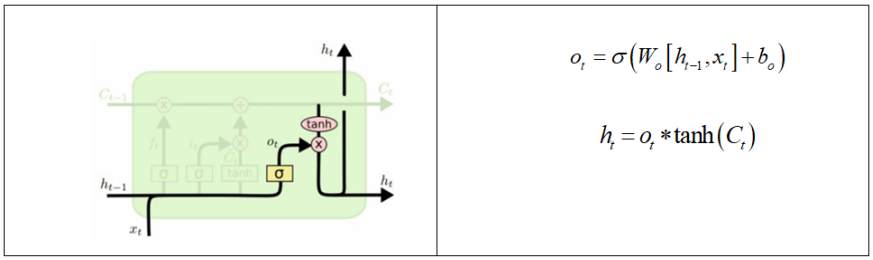

Figure 8: LSTM-cell output formation [4]

Implementation of LSTM-cell given here, is not the only possible option, there are many other modifications, as well. Description of these

modifications is given in more detail in the work [4].

References

1. Recognition of human actions. Action database: http://www.nada.kth.se/cvap/actions/

2. DAVIS: Densely Annotated Video Segmentation. In-depth analysis of the state-of-the-art in video object segmentation: http://davischallenge.org

3. Sozykin A.V. Online course "Programming Deep Neural Networks in Python": https://www.asozykin.ru/courses/nnpython

4. Understanding LSTM Networks:

5. https://web.stanford.edu/class/cs379c/archive/2018/class_messages_listing/content/Artificial_Neural_Network_Technology_Tutorials/OlahLSTM-NEURAL-NETWORK-TUTORIAL-15.pdf