Long Short-Term Memory Description and its Application in Text Processing

Lenka Skovajsová
Institute of Informatics
Slovak Academy of Science
Bratislava, Slovakia
Lenka.Skovajsova@savba.sk

Abstract— **The paper describes the state-of-the-art of Long Short-Term Memory (LSTM) neural networks with fixation to context utilization in text documents. Context in text documents is utilized mainly in the natural language processing, machine reading, but also in other areas, for example in the information retrieval. In the areas where the context is taken into account, the LSTM finds wide appliance because it is able to remember preceding states and on the base of them to evaluate the required task. Preceding states is not needed to place on the input again.**

Keywords—Long short-term memory, Natural language processing, Information retrieval, Machine reading, Text processing

I. Introduction

The Long Short-Term Memory (LSTM) is the neural network with highly adaptable architecture, so its shape can be adjusted to the particular form, depending on application. Moreover, it can be combined with architectures of other recurrent neural networks such as Deep Neural Networks (DNN), Recurrent Neural Networks (RNN), Convolutional Neural Networks (CNN) and so on.

The LSTM has wide application in text processing. The paper is oriented on the state-of-the-art of LSTM neural network with fixation on the text document processing, specifically on context utilization from the text for the solving of given task. Under the term context it is thought the immediate neighborhood of the word in the given text. So the words are not treated as independent individuals, but as the units dependent on their immediate neighborhood in the text. From the given context of these units their meaning in the document or sentence is derived. For the processing the context of words in text the LSTM is extremely suitable from the reason that it is able to remember preceding states and on the base of them to solve the given task.

LSTM has also another advantage and it is that the output does not depend on the length of input because the input is entered sequentially, one input in the one time step. The form of the output depends on the application or task and may have different forms.

In the first part of the paper the topic is introduced. In the second part of the paper, the LSTM progress from its origin the year 1997 to nowadays is described. Third part describes different LSTM architectures, in the part 4 the basic learning algorithm of LSTM is mentioned, in the part 5, the application of LSTM in machine reading area is described, and part 6 concludes the problematic.

II. Long Short-Term Memory Progress

The LSTM was mentioned for the first time in the year 1997 described by Hochreiter and Schmidhuber [1] as a neural network that can predict more longer time sequences of input data with comparison to other recurrent neural networks. The name Long Short-Term Memory arise from its ability to remember the preceding states not only for the short time (short-term), but also preceding states for far much longer time (long term) as the commonly used recurrent neural networks at that time. Original Long Short-Term Memory was without forget gate, what could make the inner state to grow indefinitely. In [16] (from the year 2000) the extension of LSTM by a forget gate is described, which solves this problem.

Later, in the years 2000 to 2010 the papers from Schmidthuber are arising that solve the speech recognition problems by LSTM, for example [18]. After the year 2010, the number of papers describing utilization of LSTM grows much more rapidly. The LSTM are utilized mainly in the speech recognition and language modeling area [3,4,5,9,12,13,14, 17, 19,20], but also other areas of LSTM utilization are appearing [2,6,8,11,15].

For our purposes, important chapter makes the text processing, in which other different works appeared recently [19-27].

LSTM is used also in tasks that process context dependency of text, to what predetermines it its structure that makes possible to store the inner states and on the base of them to predict future states.

III. LSTM Architecture

The basic construction unit of LSTM recurrent neural network is the memory block, that consists of several memory cells with which it can be communicated by input gate, forget

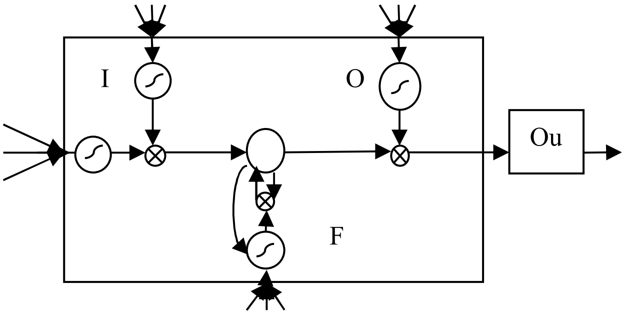

Fig. 1. The LSTM cell. I-Input gate, F-Forget gate, O-Output gate, OU-output from the cell

gate and output gate of that cell (Figure 1) [2]. The individual memory cells are organized to blocks that can be arbitrary combined to obtain the best results depending on application. In [6] the LSTM with tree structure is used, where the LSTM blocks create the nodes of the tree. In [13] the different LSTM architectures are used, where the hidden layers were replaced by recurrent neural network or by LSTM blocks.

The LSTM neural network is trained in two phases, forward pass, and backward pass [18]. In forward pass, the input to the net cell is computed by the formula

$$z_{c_j^v}(t) = \sum_m w_{c_j^v m} y_m(t-1) \tag{1}$$

where j denotes the j-th cell y_m denotes input from other cells or blocks, and $w_{c_j^v m}$ are the weights of the input of cell j. The result is multiplied with the result of the input gate activation, $y_{in_j}(t)$, which is evaluated by the formulas:

$$y_{in_j}(t) = f_{in_j}\left(z_{in_j}(t)\right) \tag{2}$$

$$z_{in_j}(t) = \sum_m w_{in_j m} y_m(t-1) \tag{3}$$

where f_{in_j} is the activation function of the input gate, $z_{in_j}(t)$ is the input from the input gate, $w_{in_j m}$ are the weights of the input gate, and $y_m(t-1)$ denotes input from other cells or blocks. Activation y_{in} multiplies input with all other cells of the memory block, and specifies in this way the patterns that are stored in it. During training the input gate is learned to open ($y_{in} \approx 1$), when the inputs are stored to the memory block or to close ($y_{in} \approx 0$), when the contents of the memory cell is protected from the irrelevant input. In the time $t = 0$ the state of the cell, $s_{c_j^v}$, is set to zero. Then, the activation of the forget gate is computed:

$$y_{\varphi_j}(t) = f_{\varphi_j}\left(z_{\varphi_j}(t)\right) \tag{4}$$

$$z_{\varphi_j}(t) = \sum_m w_{\varphi_j m} y_m(t-1) \tag{5}$$

where f_{φ_j} is activation function of the forget gate, $z_{\varphi_j}(t)$ is the input from the forget gate, $w_{\varphi_j m}$ are the weights of the forget gate, and $y_m(t-1)$ denotes input from other cells or blocks. In the next step the new cell state is acquired by the formula:

$$s_{c_j^v}(t) = y_{\varphi j}(t) s_{c_j^v}(t-1) + y_{in_j}(t) g\left(z_{c_j^v}(t)\right) \tag{6}$$

where $s_{c_j^v}(0) = 0$.

The activity if the cell circulates further, unless the forget gate remains open ($y_\varphi \approx 1$). The cell output is computed by the multiplication of the cell state s_c with output gate activation of the memory block, y_{out}:

$$y_{c_j^v}(t) = y_{out_j}(t) s_{c_j^v}(t) \tag{7}$$

where

$$y_{out_j}(t) = f_{out_j}\left(z_{out_j}(t)\right) \tag{8}$$

$$z_{out_j}(t) = \sum_m w_{out_j m} y_m(t-1) \tag{9}$$

where f_{out_j} is the activation function of the output gate, $z_{out_j}(t)$ is the input from the output gate, and $w_{out_j m}$ are the weights of the output gate. The main advantage of LSTM is that it has adaptable architecture that after making it suitable gives better results than other types of neural networks.

IV. LSTM Learning Algorithm

The described learning algorithm of LSTM [18], that we want to use in our next work, consists of combination of back propagation through time for the output units and the output gates, and from the learning algorithm truncated real time recurrent learning for the input units, input gates, and forget gates.

Firstly, the objective function E is minimized with the gradient descent change of weights Δw_{ml} from the neuron m to neuron l.

$$\Delta w_{lm} = \alpha \delta_k(t) y_m(t-1) \tag{10}$$

$$\delta_k(t) = f_k'\left(z_k(t)\right) e_k(t) \tag{11}$$

$$e_k(t) := t_k(t) - y_k(t) \tag{12}$$

where $e_k(t)$ is externally injected error, $t_k(t)$ is the target output of the memory cell, and $y_k(t)$ is the computed output of the memory cell.

The weight changes of the output gates of the j-th memory block are obtained from the usual backpropagation:

$$\Delta w_{out_j m}(t) = \alpha \delta_{out_j}(t) y_m(t) \tag{13}$$

During each step we need to change the ratio $\partial s_{c_j^v} / \partial w_{lm}$ for the weights to cell ($l = c_j^v$), to input gate ($l = in$) and to forget gate ($l = \varphi$).

$$\frac{\partial s_{c_j^v}(t)}{\partial w_{c_j^v m}} = \frac{\partial s_{c_j^v}(t-1)}{\partial w_{c_j^v m}} y_{\varphi_j}(t) + g'\left(z_{c_j^v}(t)\right) y_{in_j}(t) y_m(t-1) \tag{14}$$

$$\frac{\partial s_{c_j^v}(t)}{\partial w_{in_j m}} = \frac{\partial s_{c_j^v}(t-1)}{\partial w_{in_j m}} y_{\varphi_j}(t) + g\left(z_{c_j^v}(t)\right) f'_{in_j}\left(z_{in_j}(t)\right) y_m \tag{15}$$

$$\frac{\partial s_{c_j^v}(t)}{\partial w_{\varphi_j m}} = \frac{\partial s_{c_j^v}(t-1)}{\partial w_{\varphi_j m}} y_{\varphi_j}(t) + s_{c_j^v}(t-1) f'_{\varphi_j}\left(z_{\varphi_j}(t)\right) y_m(t-1) \tag{16}$$

In the beginning these ratios are set to zero. These ratios are then used for computation of the weigh changes to the cell, to the input gate and to the forget gate.

V. Experiments Made with LSTM

In [27] the utilization of Long Short-Term Memory in machine reading area is described. The models falling to the two main groups were described, namely, (i) symbolic matching models, where the Frame semantic parsing is mentioned that is used mainly in the question answering systems and word distance benchmark, that consists in measuring of the distance between two words, and (ii) neural network models, where the three basic approaches are used: Deep LSTM Reader, which obtains which token from the given document represents the query the most, the attentive reader, that uses bidirectional LSTM and makes a composition of the forward and backward inputs for each token, and impatient reader, that is equipped with ability to read the document again for each query token.

Experiments were made on the two data sets, CNN (www.cnn.com) and Daily Mail (www.dailymail.co.uk). The aim of the experiments is to evaluate the model ability to read and comprehend the document, that is, on the base of the document context to correctly answer on the given query.

From the results of experiments it follows that the most accurate answer on the query gave two neural network models: on the CNN collection the most accurate was the impatient reader with answer accuracy 63.8 % in the testing phase, and on the daily mail collection the most accurate approach was the attentive reader with the accuracy 69.0 % on the tested collection.

The given experiments have shown that the LSTM neural networks are suitable for machine reading and text comprehension on the base of its context.

VI. Conclusions

From the result shown higher it follows that LSTM is pouring its way and appears as promising neural network for applications in the natural language processing area that are more and more complicated for processing either in space and time. In our work we want to use Long Short-Term Memory in the natural language processing, particularly document classification based on context that, we think, is very suitable area mainly for this type of neural network.

VII. Acknowledgment

This work is supported by Slovak Science Agency VEGA No. 2/0167/16.

References

[1] HOCHREITER, Sepp; SCHMIDHUBER, Jürgen. Long short-term memory. *Neural computation*, 1997, 9.8: 1735-1780.

[2] MA, Xiaolei, et al. Long short-term memory neural network for traffic speed prediction using remote microwave sensor data. *Transportation Research Part C: Emerging Technologies*, 2015, 54: 187-197.

[3] RAO, Kanishka, et al. Grapheme-to-phoneme conversion using long short-term memory recurrent neural networks. In: *Acoustics, Speech and Signal Processing (ICASSP), 2015 IEEE International Conference on*. IEEE, 2015. p. 4225-4229.

[4] ZEN, Heiga; SAK, Haşim. Unidirectional long short-term memory recurrent neural network with recurrent output layer for low-latency speech synthesis. In: *Acoustics, Speech and Signal Processing (ICASSP), 2015 IEEE International Conference on*. IEEE, 2015. p. 4470-4474.

[5] SAK, Haşim; SENIOR, Andrew; BEAUFAYS, Françoise. Long short-term memory based recurrent neural network architectures for large vocabulary speech recognition. *arXiv preprint arXiv:1402.1128*, 2014.

[6] TAI, Kai Sheng; SOCHER, Richard; MANNING, Christopher D. Improved semantic representations from tree-structured long short-term memory networks. *arXiv preprint arXiv:1503.00075*, 2015.

[7] DUCHI, John; HAZAN, Elad; SINGER, Yoram. Adaptive subgradient methods for online learning and stochastic optimization. *Journal of Machine Learning Research*, 2011, 12.Jul: 2121-2159.

[8] DYER, Chris, et al. Transition-based dependency parsing with stack long short-term memory. *arXiv preprint arXiv:1505.08075*, 2015.

[9] SAINATH, Tara N., et al. Convolutional, long short-term memory, fully connected deep neural networks. In: *Acoustics, Speech and Signal Processing (ICASSP), 2015 IEEE International Conference on*. IEEE, 2015. p. 4580-4584.

[10] HEIGOLD, Georg, et al. Asynchronous stochastic optimization for sequence training of deep neural networks. In: *Acoustics, Speech and Signal Processing (ICASSP), 2014 IEEE International Conference on*. IEEE, 2014. p. 5587-5591.

[11] CHEN, Xinchi, et al. Long Short-Term Memory Neural Networks for Chinese Word Segmentation. In: *EMNLP*. 2015. p. 1197-1206.

[12] GEIGER, Jürgen T., et al. Robust speech recognition using long short-term memory recurrent neural networks for hybrid acoustic modelling. In: *Interspeech*. 2014. p. 631-635.

[13] SAK, Hasim; SENIOR, Andrew W.; BEAUFAYS, Françoise. Long short-term memory recurrent neural network architectures for large scale acoustic modeling. In: *Interspeech*. 2014. p. 338-342.

[14] SAK, Hasim, et al. Sequence discriminative distributed training of long short-term memory recurrent neural networks. *entropy*, 2014, 15.16: 17-18.

[15] YAO, Kaisheng, et al. Spoken language understanding using long short-term memory neural networks. In: *Spoken Language Technology Workshop (SLT), 2014 IEEE*. IEEE, 2014. p. 189-194.

[16] GERS, Felix A.; SCHMIDHUBER, Jürgen; CUMMINS, Fred. Learning to forget: Continual prediction with LSTM. *Neural computation*, 2000, 12.10: 2451-2471.

[17] GRAVES, Alex; JAITLY, Navdeep; MOHAMED, Abdel-rahman. Hybrid speech recognition with deep bidirectional LSTM. In: *Automatic Speech Recognition and Understanding (ASRU), 2013 IEEE Workshop on*. IEEE, 2013. p. 273-278.

[18] GERS, Felix A.; SCHRAUDOLPH, Nicol N.; SCHMIDHUBER, Jürgen. Learning precise timing with LSTM recurrent networks. *Journal of machine learning research*, 2002, 3.Aug: 115-143.

[19] VINYALS, Oriol, et al. Grammar as a foreign language. In: *Advances in Neural Information Processing Systems*. 2015. p. 2773-2781.

[20] WEN, Tsung-Hsien, et al. Semantically conditioned lstm-based natural language generation for spoken dialogue systems. *arXiv preprint arXiv:1508.01745*, 2015.

[21] MIAO, Yishu; YU, Lei; BLUNSOM, Phil. Neural variational inference for text processing. In: *International Conference on Machine Learning*. 2016. p. 1727-1736.

[22] MA, Xuezhe; HOVY, Eduard. End-to-end sequence labeling via bi-directional lstm-cnns-crf. *arXiv preprint arXiv:1603.01354*, 2016.

[23] CHENG, Jianpeng; DONG, Li; LAPATA, Mirella. Long short-term memory-networks for machine reading. *arXiv preprint arXiv:1601.06733*, 2016.

[24] ZHOU, Chunting, et al. A C-LSTM neural network for text classification. *arXiv preprint arXiv:1511.08630*, 2015.

[25] PALANGI, Hamid, et al. Deep sentence embedding using long short-term memory networks: Analysis and application to information retrieval. *IEEE/ACM Transactions on Audio, Speech and Language Processing (TASLP)*, 2016, 24.4: 694-707.

[26] JI, Yangfeng; HAFFARI, Gholamreza; EISENSTEIN, Jacob. A latent variable recurrent neural network for discourse relation language models. *arXiv preprint arXiv:1603.01913*, 2016.

[27] HERMANN, Karl Moritz, et al. Teaching machines to read and comprehend. In: *Advances in Neural Information Processing Systems*. 2015. p. 1693-170oble, and I.N. Sneddon, “On certain integrals of Lipschitz-Hankel type involving products of Bessel functions,” Phil. Trans. Roy. Soc. London, vol. A247, pp. 529-551, April 1955.