Review Report on “Long Short-Term Memory”

GITHUB: https://github.com/zhang23ke/UTS_ML_2019

Content:

What is RNN?

Neural networks is kind of black boxes in some way and can fit arbitrary functions. Given enough training data and the networks can be trained. After training the neural network model, given an x at the input layer, a specific y can be obtained at the output layer through the network. They can only take and process one input by one. The former input has nothing to do with the latter. However, some tasks need to have the ability of processing sequence information. The previous input is related to the latter input.

The reason to use recurrent neural network (RNN) is that the current output of a sequence has relationship to the output of previous. The reason is that the network will remember the information from previous moment and after the calculation with previous information, output to the current output gate. The input of the hidden layer includes not only the original input of this layer but also the output of the same layer at the last moment (Qiu, 2018).

The problems of RNN

RNN is used widely recently, even at the publish time of the paper. RNN can be used in the field of speech processing, music composition and so on. The performance is not as expected since it takes long time to do the work. The problem is that error signals flowing backwards tend to either blow up or vanish. The distance between the relevant information and the information needed is very close. RNN has the ability to learn to use the previous information. However, in some cases, we need more context information. It is likely that the distance between the relevant information and the location where the information is needed is very far. Unfortunately, as the distance increases, RNN becomes unable to connect relevant information.

In the paper, CEC (constant error carrousel) is introduced and it's the self to self connection in RNN. Obviously because of the constraints above, this CEC module is linear and that's the motivation for the LSTM to deal with the problem of vanishing gradients. To avoid gradient explode or vanish, the naive solution in the paper is that making error flow as a constant number.

Innovation:

Why we use LSTM?

LSTM was designed to solve the problem of long-distance dependence. Remembering long-distance information is actually their most basic behavior. The key of LSTM is unit. The unit state is like a conveyor belt. It runs directly along the

whole chain, with only a few simple linear operations. Information can easily remain unchanged. LSTM can add or delete the information from unit state, and it can be processed by gates which have the ability to allow information pass or block.

In RNN, each memory unit h_t-1 is multiplied by W and derivative of activation function. This continuous multiplication makes memory decay very fast. LSTM "adds" memory and current input, so that the previous memory will continue to exist rather than be affected by multiplication and "disappears" partially, so it will not decay. But this naive approach is too straightforward. It's actually a linear model. So LSTM introduced three gates.

The paper introduces that all "gradient-based" methods will encounter two problems in weight updating: input weight conflict and output weight conflict. The main idea is that for the weight of neurons, different data will bring different updates, which may cause conflicts (for example, some inputs want to make the weight smaller, some want to make it larger). The network may need to selectively "forget" some inputs and “block” some outputs to avoid affecting the weight updates of the next layer. In order to solve these problems, the "door" is put forward. Since the operation of LSTM on memory is additive and linear, which makes the memory of different time series have the same effect on the current. In order to make the memory of different time series controllable, LSTM introduced input, output and forget gates.

Technical quality:

The technical quality of this paper is high because the authors proved the theory using experiment.

Firstly, authors provide 6 experiments to show LSTM’s performance in different case. These experiments explained that LSTM can perform well in many ways adequately.

Secondly, each experiment provided enough test data to prove the correctness of theory. The authors use lots of evidence to prove LSTM is possible to be applied in future and in different fields. For example, in experiment 2, it compares several models such as RTRL, ELM, RCC and LSTM and gets result. It compares property of each model such as hidden units, number of weights, learning rate and so on. This shows each model’s performance clearly. From these data, readers can easily figure out the introduced method’s performance.

Thirdly, the paper discussed limitations and advantages at last. LSTM cannot solve some problems like strongly delayed XOR problems. Compared with RNN, LSTM need additional gates. The third problem is that LSTM sees the entire input string at once. The advantages are also provided: LSTM can solve problems with long time lags and can handle noise, distributed representations, and continues values. As discussed in this paper, relevant inputs in the input sequence do not matter even the positions are widely separated. There is no need to tune parameters (Hochreter, 1997).

Application:

Applied fields

LSTM can be used in many fields, and it is widely used in natural language processing (NLP). This model can help computers to interpret humans' language and give response. NLP can be divided into several sub-fields such as:

Dialogue Systems: Famous examples include Siri, Alexa and Cortana.

Emotional Analysis: Emotional Recognition of a Text.

Mapping: Describe a picture in one sentence.

Machine Translation: Translating one language into another.

Speech Recognition: Let the computer recognize spoken language.

Improvement of LSTM

Understanding the main change is that cell status information is added to the input of the three control gates, which are called peephole connections. GRU is a much changed version, which was introduced in 2014. The main change is that the forgetting gate and input gate are combined into an update gate, and then the cell state information flow and the hidden layer state information flow are combined into one information flow. Its structure is simpler than standard LSTM, and it is also a variant of LSTM that is widely used nowadays (J Chung, 2015).

Another improvement of LSTM is bidirectional LSTM. Firstly, bidirectional RNN (BRNN) is introduced in paper '*HYBRID SPEECH RECOGNITION WITH DEEP BIDIRECTIONAL LSTM*'. BRRN can process data in both directions with two separate hidden layers, which are then fed forward to the same out layer. The authors combined BRNNs with LSTM to get the new that access long-range context in both input directions (Alex, 2013).

Presentation:

The quality of this paper's presentation is high. The paper firstly introduced the background of recurrent networks including the current problems and remedy. Then it introduces previous work and analysis the previous methods. After that, it introduces that new solution should use constant error backprop, and the method of long short-term memory. Followed by 6 experiments, the paper gets the limitations and advantages. This paper guides readers to get familiar with this topic and then understand it deeply. The structure and sub topics are organized well and the description quite appropriate.

Reference:

Passricha, V. and Aggarwal, R. (2019). A Hybrid of Deep CNN and Bidirectional LSTM for Automatic Speech Recognition. *Journal of Intelligent Systems*, 0(0).

Cs.toronto.edu. (2019). [online] Available at: http://www.cs.toronto.edu/~graves/asru_2013.pdf [Accessed 28 Aug. 2019].

Anon, (2019). [online] Available at: https://www.researchgate.net/publication/272195367_Gated_Feedback_Recurrent_Neural_Networks [Accessed 28 Aug. 2019].

Qiu, J., Tian, J., Chen, H. and Lu, X. (2019). *Prediction Method of Parking Space Based on Genetic Algorithm and RNN*.