ID2223 - Scalable Machine Learning and Deep Learning

- Review Questions 4 -

PAOLO TETA RALFS ZANGIS

teta | zangis @kth.se

December 4, 2021

1 Question 1

Considering RNNs, going deeper in the network by adding more and more layers can lead to the vanishing or exploding gradient problem. If we focus on a specific layer, the gradient is calculated as the product of all the previous gradients from the previous layers in the network. This means that step-by-step the gradient starts to decrease if the previous one is less than 1. So, this implies the vanishing gradient problem, thus the weights will be no more updated and the training stops. In order to solve this problem, LSTM (Long Short-Term Memory) have been introduced, where the network is able to learn what is necessary to store and what to throw away.

2 Question 2

Considering LSTM networks, we have three different gates that regulate information flow in an LSTM cell: forget gate, input gate and output gate. These gates are different neural networks that decide which information can pass on the cell state. Moreover, they can learn which information is relevant to keep or forget during the training process.

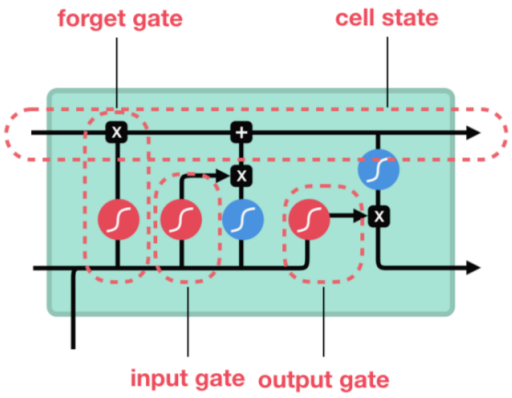

Figure 1: LSTM layout

FORGET GATE $\Longrightarrow$ It decides which information should be kept or thrown away. This gate accepts as input both information from the previous hidden state and the one from the current input, which are evaluated by the sigmoid function. As output we obtain values between 0 and 1. The closer to 0 means to forget, while the closer to 1 means to keep.
INPUT GATE $\Longrightarrow$ It's used to update the cell state. Both the information from the previous hidden state and the current input are passed into the sigmoid and the tanh activation functions. The first one decides whether the values will be updated or not, so if the information is important or not, as in the forget gate. The second one squeeze the values between -1 and 1, thus regulating the whole network. Then, these two outputs are multiplied by each other and the result is passed to the cell state. So, this means that the sigmoid function will decide if the output of the tanh function is important or not.
OUTPUT GATE $\Longrightarrow$ It decides what the next hidden state should be. Firstly, both the previous hidden state and the current input are passed into the sigmoid function. Then, the result is multiplied by the updated cell state passed through the tanh function. So, again we multiply the sigmoid and tanh outputs to decide which information should be kept for the hidden state. After this, the updated cell state and the new hidden state are passed to the next time step.

3 Question 3

Given the network and the error $E = E^{(1)} + E^{(2)}$, we can obtain $\frac{\partial E}{\partial u}$ as follows:

$$\frac{\partial E}{\partial u} = \sum_t \frac{\partial J^{(t)}}{\partial u} = \frac{\partial J^{(1)}}{\partial u} + \frac{\partial J^{(2)}}{\partial u}$$

$$\frac{\partial J^{(1)}}{\partial u} = \frac{\partial J^{(1)}}{\partial \hat{y}^{(1)}} \frac{\partial \hat{y}^{(1)}}{\partial z^{(1)}} \frac{\partial z^{(1)}}{\partial h^{(1)}} \frac{\partial h^{(1)}}{\partial s^{(1)}} \frac{\partial s^{(1)}}{\partial u}$$

$$\frac{\partial J^{(2)}}{\partial u} = \frac{\partial J^{(2)}}{\partial \hat{y}^{(2)}} \frac{\partial \hat{y}^{(2)}}{\partial z^{(2)}} \frac{\partial z^{(2)}}{\partial h^{(2)}} \frac{\partial h^{(2)}}{\partial s^{(2)}} \frac{\partial s^{(2)}}{\partial u} + \frac{\partial J^{(2)}}{\partial \hat{y}^{(2)}} \frac{\partial \hat{y}^{(2)}}{\partial z^{(2)}} \frac{\partial z^{(2)}}{\partial h^{(2)}} \frac{\partial h^{(2)}}{\partial s^{(2)}} \frac{\partial s^{(2)}}{\partial h^{(1)}} \frac{\partial h^{(1)}}{\partial s^{(1)}} \frac{\partial s^{(1)}}{\partial u}$$

4 Question 4

Considering RNNs, a *sequence-to-sequence* network takes as inputs a sequence and produce another sequence as outputs, as shown in Figure 2. Both input and output sequences have the same length.

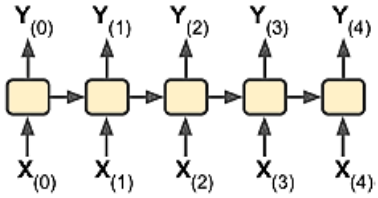

Figure 2: *Sequence-to-sequence* model

If we want to address a language translation task, we should give as input sequence a sentence in one language and then the network will produce as output another sequence that is the sentence in another language. But there is a problem, the two sentences must have the same number of words, that is not always true and satisfied for each pair of languages. So, to overcome this limitation we can use the *encoder-decoder* network: a *sequence-to-vector* network (encoder) followed by a *vector-to-sequence* network (decoder).

5 Question 5

ATTENTION $\Longrightarrow$ It's used in combination with RNNs to improve their performance for NLP.
SELF-ATTENTION $\Longrightarrow$ It avoids using RNNs by relying on the concept of attention to encode sequences. This allows for the solution to offer better and faster results.