COMP9417 - Assignment 2

Avinash K. Gupta, Yuyang Shu, Maria Oei

June 4, 2017

1 Introduction

With over 100 millions of monthly visitor in Quora, it is inevitable that many people are asking similar questions. This has become an issue as user has to read through responses to many questions in order to find the best answer. The aim of this project is to implement algorithms to identify 2 similar questions which can help Quora in improving user experience by finding high quality answers to questions.

The 2 approaches used in this project were perceptron learning and LSTM. In the perceptron learning, we used 3 inputs which measured semantic similarity, word order and word overlaps between 2 sentences.

In LSTM, bla bla bla bla

We will see that for our case the performance of the 2 models were quite similar. However, there are other research done on LSTM where it has been shown that LSTM has the potential to performs well with enough training and with careful selection of initial weight.

2 Methodology

2.1 Perceptron Learning

- What do we implement (Theory) - How do we implement it

2.2 LSTM

2.2.1 Overview

In this approach, we present siamese adaption of Long Short-Term Memory(LSTM) network for labeled data contains pairs of variable-length sequences. This model is used to get the semantic similairty between the sentences using complex neural network. For this applications, we provide word-embedding vectors supplemented with synonymic information to the LSTMs, which use a fixed size vector to convert the syntactic meaning of the sentence.

2.2.2 Model

It's a supervised learning model, where each data consists of pair of sequences $(x_1^{(a)}, ..., x_{n_a}^{(a)}), (x_1^{(b)}, ..., x_{n_b}^{(b)})$ of fixed size vectors along with a single label y(human labeled) for the pair. Note that sequences may be differenent length. These data will be pass to the model with a purpose of learning the semantics. In the above diagram, there are two networks $LSTM_a$ and $LSTM_b$,

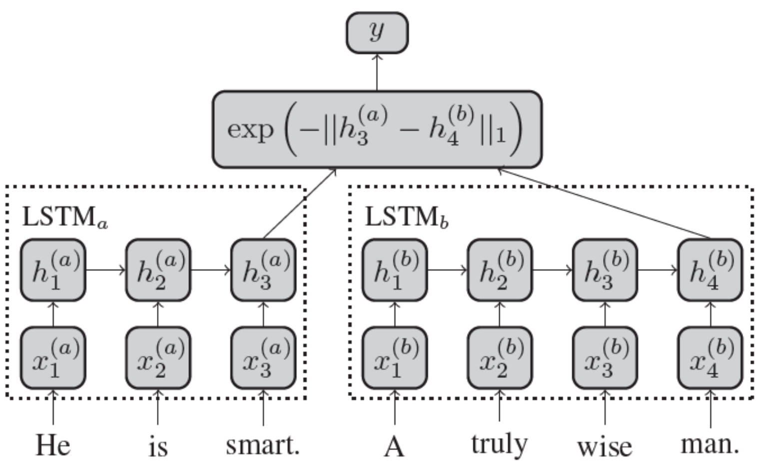

Figure 1: lstm architecture

each of them process a sentence in a given pair and predict whether $LSTM_a = LSTM_b$ based on the similarity between the vectors. In each LSTM, the word vector is employ to the hidden layer and calculation is done and output is passed to the next hidden layer to remember the previous context and so on. In above figure, the hidden layer which is core part of our neural network. It performs operation listed below on each word vector.

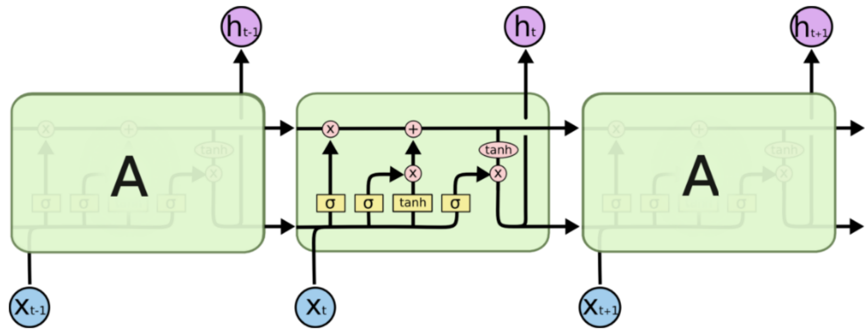

Figure 2: repeating module of hidden layer

- The first sigmoid function is used to decide which information to keep and what to ignore

$$f_t = sigmoid(W_i x_t + U_i h_{t-1} + b_i)$$

- Now it's time to update the old cell state into new cell state. The last step already decided what to do, in this step we just actually need to do it.

$$i_t = sigmoid(W_f x_t + U_f h_{t-1} + b_f)$$

$$c_t^{'} = \tanh(W_c x_t + U_c h_{t-1} + b_c)$$

- In this step, we will multiply old state c_{t-1} with f_t to forget things we decided to forget earlier. Then we add to new candidate values

$$c_t = i_t \odot c_t^{'} + f_t \odot c_{t-1}$$

- Final step defines what we actually need to output based on out filtered cell state and then tanh(used to push values between -1 and 1) and multiply with sigmoid function to decide the parts we need to output.

$$o_t = sigmoid(W_o x_t + U_o h_{t-1} + b_o)$$

$$h_t = o_t \odot \tanh(c_t)$$

Above process are carried by both the LSTM and emploed to the Similarity Function given below, which calculates the Manhattan differnece between the ouputs.

$$g(h_{T_a}^{(a)}, h_{T_b}^{(b)}) = \exp(- \parallel h_{T_a}^{(a)} - h_{T_b}^{(b)} \parallel_1) \in [0, 1].$$

2.3 LSTM

2.3.1 Overview

In this approach, we present siamese adaption of Long Short-Term Memory(LSTM) network for labeled data contains pairs of variable-length sequences. This model is used to get the semantic similairty between the sentences using complex neural network. For this applications, we provide word-embedding vectors supplemented with synonymic information to the LSTMs, which use a fixed size vector to convert the syntactic meaning of the sentence.

2.3.2 Model

It's a supervised learning model, where each data consists of pair of sequences $(x_1^{(a)}, ..., x_{n_a}^{(a)}), (x_1^{(b)}, ..., x_{n_b}^{(b)})$ of fixed size vectors along with a single label y(human labeled) for the pair. Note that sequences may be differenent length. These data will be pass to the model with a purpose of learning the semantics. In the above diagram, there are two networks $LSTM_a$ and $LSTM_b$,

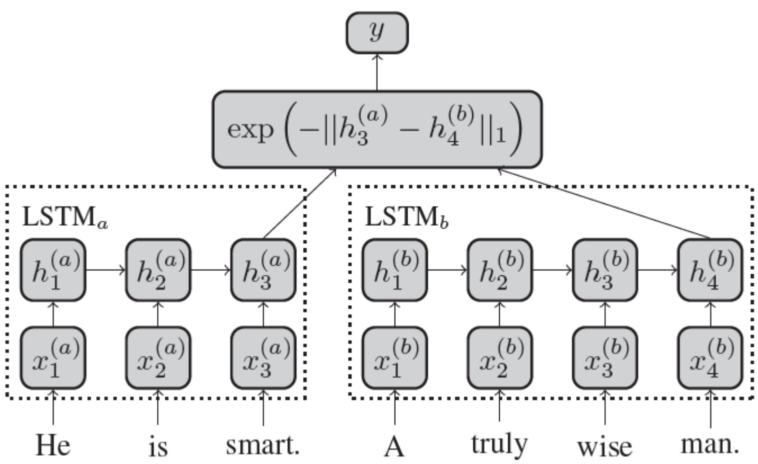

Figure 3: lstm architecture

each of them process a sentence in a given pair and predict whether $LSTM_a = LSTM_b$ based on the similarity between the vectors. In each LSTM, the word vector is employ to the hidden layer and calculation is done and output is passed to the next hidden layer to remember the previous context and so on. In above figure, the hidden layer which is core part of our neural network. It performs operation listed below on each word vector.

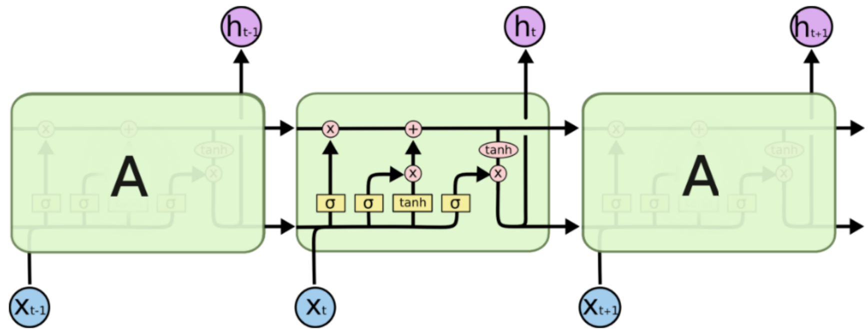

Figure 4: repeating module of hidden layer

- The first sigmoid function is used to decide which information to keep and what to ignore

$$f_t = sigmoid(W_i x_t + U_i h_{t-1} + b_i)$$

- Now it's time to update the old cell state into new cell state. The last step already decided what to do, in this step we just actually need to do it.

$$i_t = sigmoid(W_f x_t + U_f h_{t-1} + b_f)$$

$$\grave{c}_t = \tanh(W_c x_t + U_c h_{t-1} + b_c)$$

- In this step, we will multiply old state c_{t-1} with f_t to forget things we decided to forget earlier. Then we add to new candidate values

$$c_t = i_t \odot \grave{c}_t + f_t \odot c_{t-1}$$

- Final step defines what we actually need to output based on out filtered cell state and then tanh(used to push values between -1 and 1) and multiply with sigmoid function to decide the parts we need to output.

$$o_t = sigmoid(W_o x_t + U_o h_{t-1} + b_o)$$

$$h_t = o_t \odot \tanh(c_t)$$

Above process are carried by both the LSTM and emploed to the Similarity Function given below, which calculates the Manhattan differnece between the ouputs.

$$g(h_{T_a}^{(a)}, h_{T_b}^{(b)}) = \exp(- \parallel h_{T_a}^{(a)} - h_{T_b}^{(b)} \parallel_1) \in [0, 1].$$

3 Result

How do they perform

4 Discussion

4.1 Conclusion

4.2 Limitation & Improvement