NLP Assignment

14th March

E044 Tanish Surana

What is RNN?

- These are neural networks working on the problem of sequence problems in data
    - That is data is dependent on the previous value (also on future) of data
- HMM is also useful here but it suffers from a lack of accuracy as there we have a Markov assumption
- These are memory-based networks as we feed in the output of the previous calculation as input to the newer calculation
- There are many types of RNN
    - One to one
        - Classification of text
    - Many to one
        - Fill in the blanks
    - One to many
        - Context prediction
    - Many to many
        - Machine translation
        - All these are the application in nlp
- Mainly these are used for short term memory-related task

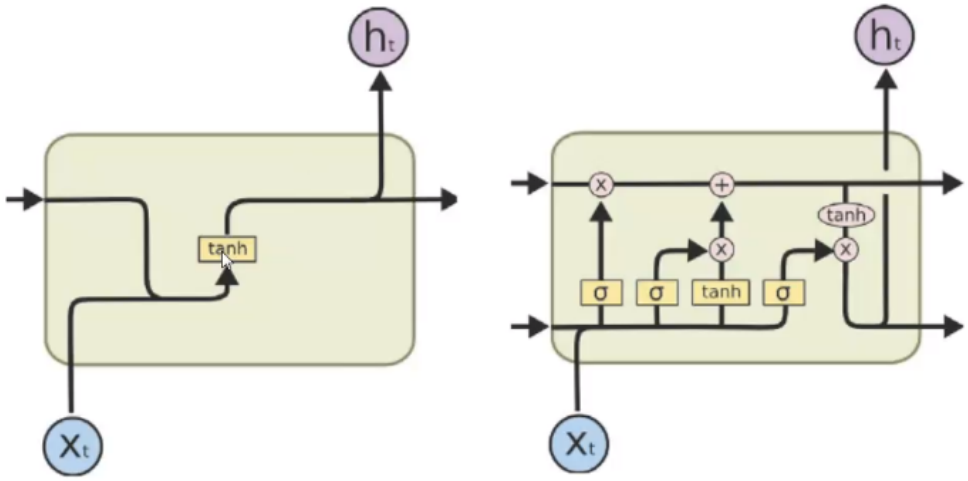
Why RNN sucks?

- There are two problems with rnn
    - They only remember stuff for short term
        - I.e

        - I like pizza. Sent2,.. Sent3 and my favourite cusine is ____
            - Rnn won't be able to fill in Italian here
        - So we need a model which remembers stuff for long term
    - The second problem is it suffers from vanishing gradient
        - I.e. while training the gradient of weight wrt loss becomes zero
        - The reason for this is the gradient of current weight also depends on the previous ones, thus making a product chain of weights.
            - 0.1x0.1x0.1 = 0.001
            - Vanishing gradient

Why LSTM?

- Lstm or long short term memory solves both the problems of RNN
- They have a cell state which acts as a long term storage
- Also, the gradient of the weights here does not have the weight component in a product chain
- Brief working of lstm
    - There are 3 gates
        - Forget gate
            - There removes stuff from the cell state
        - Input gate
            - This gate adds stuff to the cell state for long term storage
        - Output gate
            - This gate uses the thing stored in the cell state to give output for the current input



(a) RNN (b) LSTM

-
    - You can see the architecture of both lstm and rnn
        - Since RNN uses tanh function it again is a contributor to the vanishing gradient problem
- Now coming to the applications of lstm
    - The previous problem of long term memory is solved here
        - I like pizza … sent2… sent3…. My favourite cuisine is ____
            - Lstm would store the word pizza in the cell state and then it would be able to use that to predict Italian as my favourite cuisine
    - The main application of lstm
        - Machine translation
        - The above example
        - Speech recognition
        - Sentiment analysis
- Also similar to rnn we have one to one, many to one, many to many and one to many architecture

Why lstm sucks

- Lstms are prone to overfitting
- They take longer to train
- Dropout to prevent overfitting is difficult to implement in lstm

- Weight initialization has a larger impact on the training of lstm

Why GRU over lstm?

- This not only solves the problem of rnn but also solves most of the problems with lstm
    - Long term memory and vanish gradient issues of rnn
- It has 2 gates
    - Reset and
        - How much info has to be removed
    - Update gate
        - How much info has to be passed to the next calculation
        -
- It is less complex than lstm
- It is mostly used over lstm when you have fewer data available to you
- It has no memory state unlike lstm
- It is more efficient than lstm as it is less complex, i.e. faster training,
- Disadvantage of GRU
    - Lstm is better when more data is available
    - Lstm remembers longer sequences
- Performance of both lstm and GRU are on par with each other
    - Selecting when to use which depends on the time available, dataset and type of problem (does it has longer sequences?)