# Deep Learning Assignment 4

**Peter Yun-shao Sung** yss265@nyu.edu

## 1 Approaches to RNN

### 1.1 Architecture

The initial model is based on lstm as building block, with 2 layers and 20 sequence length. Layers are horizontally considering the current given input (or word) and the states passed from previous sequence. Sequence is the stack of multiple layers, and is used for vertically passing states to the next layers. This way, model can not only predict the next word based on current word (horizontal), but the predictions can be altered based on the sequence of previous words (vertical). Regarding to each lstm layer, there are number of rnn_size lstm unit, and each of the unit perform the operation shown as figure 1.

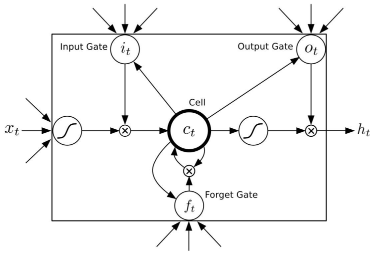

Figure 1: Operation of each of lstm unit

### 1.2 Learning Techniques

The two main learning techniques were the dropout and gradient clip. Dropout method was applied between each of lstm module and the final output. The inital module didnot used the dropout, and therefore we can observed that the perplexity of training improved well but not for the validation. This implies the concerns of overfitting and dropout might be a good option to improve.
Gradient clipping might be important method during the learning process too. Soft-clipping was initially applied to the model, which it observes the L2 norm of gradient and every gradient will multiply the portions it overpassed if the norm is higher than the threshold. The other method I used during this assignment is hard clipping, which we only cut the gradiend for only the one that pass threshold, instead of the every gradients. The inspiration is mainly from the curiousity of whether soft clipping downgrade the learning process for other gradient.

### 1.3 Training Procedure

Data was preprocessed in the shape of (lines,batch_size), which is basically to traing a batch size of words. During each forward or backward propagation, line $i$ and $i + 1$ is the corresponding current and next word. When feed in network, the $next\_h$ is the output from the lstm module which is in the shape of (batch_size, vocab_size), and further calculates the prediction of each word along the batch_size axis by LogSoftMax. The perplexity is the accumulated error calculated by ClassNLLCriterion, and we are optimizing the perplexity error metric.

29th Conference on Neural Information Processing Systems (NIPS 2016), Barcelona, Spain.

## 2 Softmax regression gradient calculation