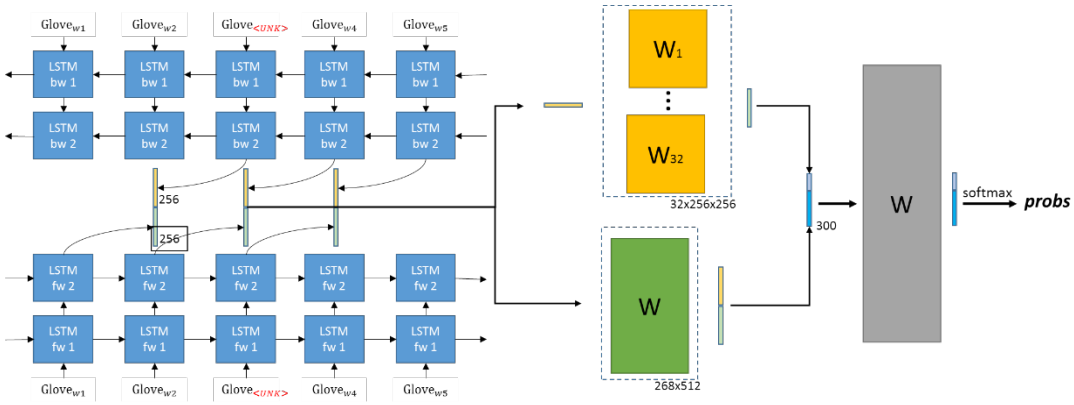
Environment

Linux, Intel i7, GTX 960M(4G memory)

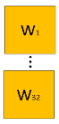
Package: tensorflow, nltk, genism, numpy

Model description



glove (https://nlp.stanford.edu/projects/glove/)

Performance

1. Subsample target frequent target word $P(w_i) = (\sqrt{\frac{z(w_i)}{0.001}} + 1) \cdot \frac{0.001}{z(w_i)}$ (~10%up)
   (http://mccormickml.com/2017/01/11/word2vec-tutorial-part-2-negative-sampling/)
2. Finetune pretrained embedding (~5%up)
3. Stack up to 2 BiRNN (~5%up)
4. Increase diversity of feature interaction by adding  (~5%up)
   (李弘毅老師 3/24 上課內容)
5. Summation of log probability of neighbor words (5 words left and 5 words right) in inference time (~5%up)

Experiment settings and results

 # Model Structure

Voc_Size = 60000

Hidden_Size_per_Single_LSTM_Cell = 256

Basic_Cell = tf.contrib.rnn.BasicLSTMCell(Hidden_Size, forget_bias=0.1)

Num_Stack_Layer = 2

Pre_Trained_Ebd = glove

Training

Train_Iters = 90000 # ~ 7 epochs = ~ 15 hours

Batch_Size = 64

Loss = “Sampled_Softmax_Loss”

Sampled_Softmax_Loss_Num_Sampled = 1024

Optimizer_Type = “Adam”

Learning_Rate = 0.01

Learning_Rate_Decay_Step = 39000 # ~3 epochs

Finetune_Embedding_Rate = 0.05 * current_learning_rate

SubSample_Rate = 0.001 # $P(w_i) = (\sqrt{\frac{z(w_i)}{0.001}} + 1) \cdot \frac{0.001}{z(w_i)}$

Results

Public set: 0.62308 **Private set: 0.65385**