CHINESE WORD SEGMENTATION

Unstacked Bi-LSTM Model

Abstract

Our task is to implement a Chinese word segmenter, reproducing the SOTA using Bi-LSTM network

Ahmed El Sheikh - 1873337

Elsheikh.1873337@student.uniroma1.it

Models

Both Bi-LSTM and Stacked Bi-LSTM networks were implemented, and most of the variables were replicated from the paper. Nesterov momentum (0.95) SGD optimizer, with categorical cross entropy for loss function 5 epochs for unstacked Bi-LSTM model, with batch size of 64

Datasets

Working on the All datasets concatenated, first, I had to simplify traditional datasets using 'hanzi-convert', reading dataset, producing input files (no spaces) and labels file. Sentences as per dataset are not of consistent length, thus I had to pad the sentences as per batch max length.
And produce from it training samples of unigrams and bigrams, so our input data were composed of [train_x_unigrams, train_x_bigrams]. And, for our labels, train_y was padded as well, and converted to one hot encoded.
This padding needed to be masked during the training period, which was done using `mask_zeros` attribute in keras, as well as, masking layer, and tf.sequence_mask.
In preparation for using the 4 datasets, I concatenated the 4 datasets in one huge file, vocab_size = 1,047,644.

My Approach

I started with using 'GloVe' Pre-trained Embeddings for both uni & bigrams instead of learning my own embedding layers.
Model had Early Stopping to avoid overfitting, as well as, 'ReduceLROnPlateau', so model don't get stuck on shoulder/saddle point.
As trail of improvisation, I tried to add kernel and bias regularizers as per LSTM layer in order to avoid overfitting, so I added L2 Regularizer with coeff = 0.01 for both.

GridSearch CV

Grid search algorithm supplied from Sklearn, could not be used with multiple input models, so I did the tuning manually, I only tried varying the learning rate And from graphs 2 & 3 it can be noted that lower learning rate basically requires more time to learn, which is not news.

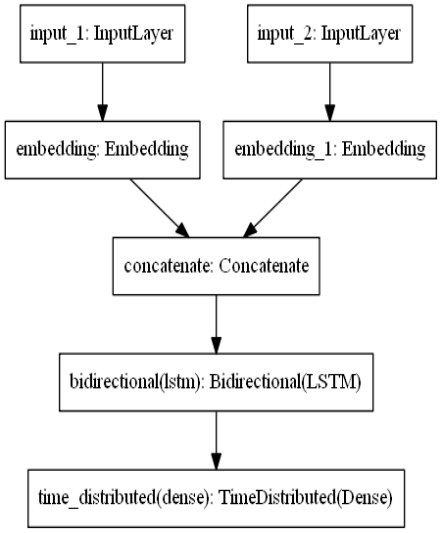

Figure 1: Unstacked Bi LSTM model

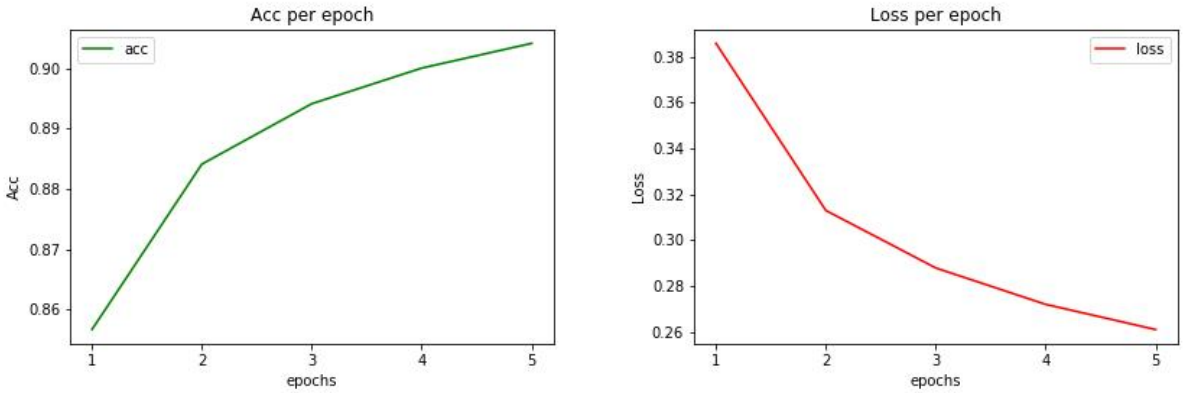

Figure 2:Acc & Loss as per 5 epochs of unstacked model training no regularizer

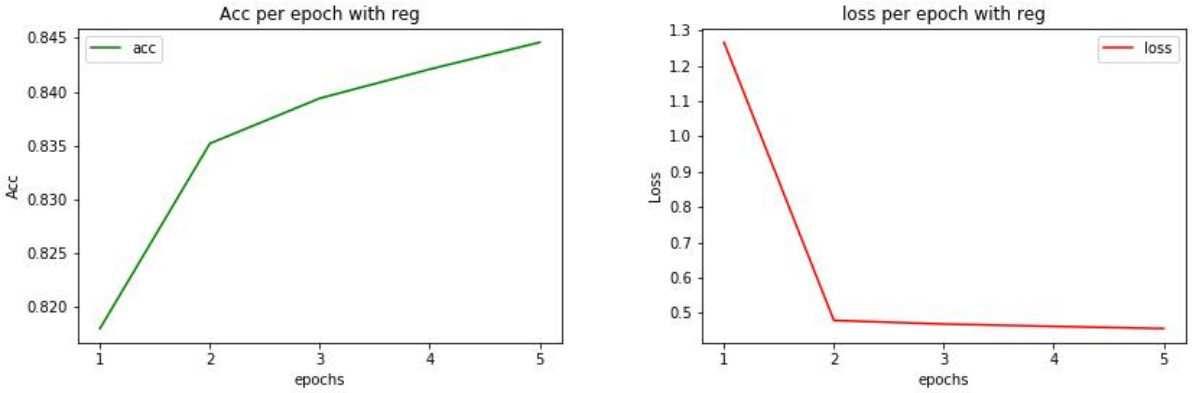

Figure 3:Acc & Loss as per 5 epochs of unstacked model training regularizer

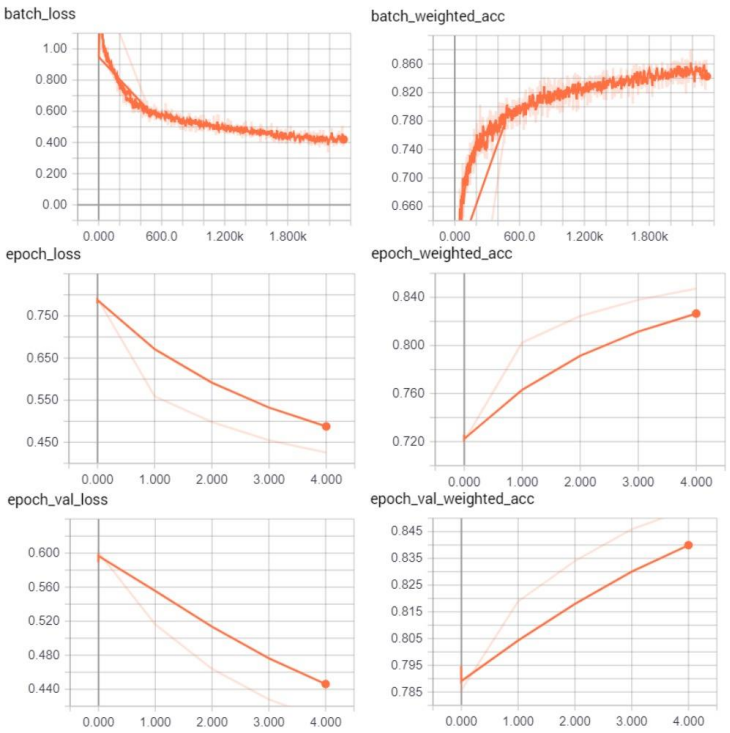

Figure 4: ACC & Loss using Learning Rate 4e-3

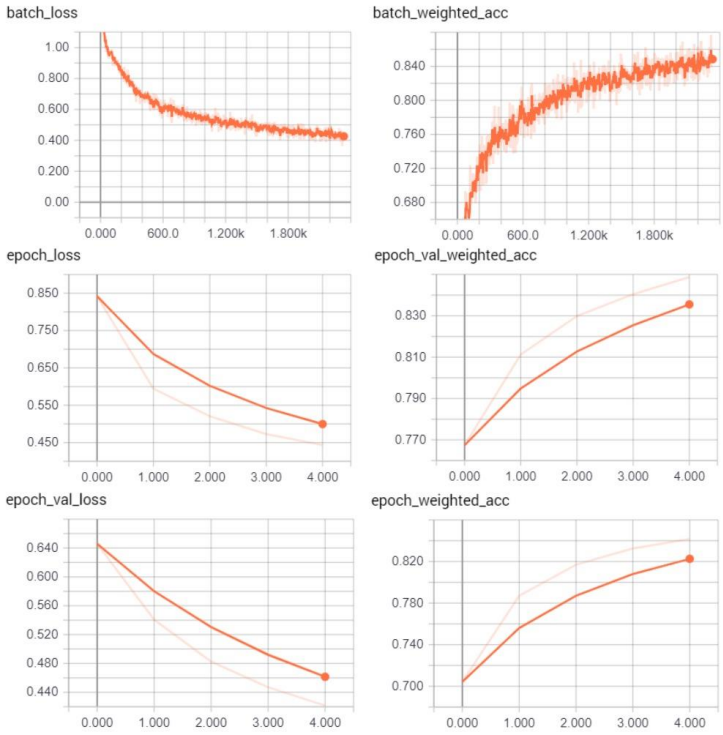

Figure 5:ACC & Loss using Learning Rate 3e-3