DL for NLP, Coding Assignment
CSCI 544, Spring 2019

POS Tagging

Gokul Reddy Muddarla
muddarla@usc.edu

Introduction:
The Goal of this task is to predict the Parts of Speech (POS) tags for words in a sentence of any language, given some pre-tagged training data set. The proposed sequence to sequence model is a generalized framework which uses a bi-directional layer of recurrent neural network, then followed by a fully connected layer, and the output of this layer is passed through a CRF layer for predicting the POS tag for each word in a sentence.

Model:
Figure 1 shows the basic flow of the network used. The final output in the figure is logits tensor of size (batch_size x max_len x num_of_tags).

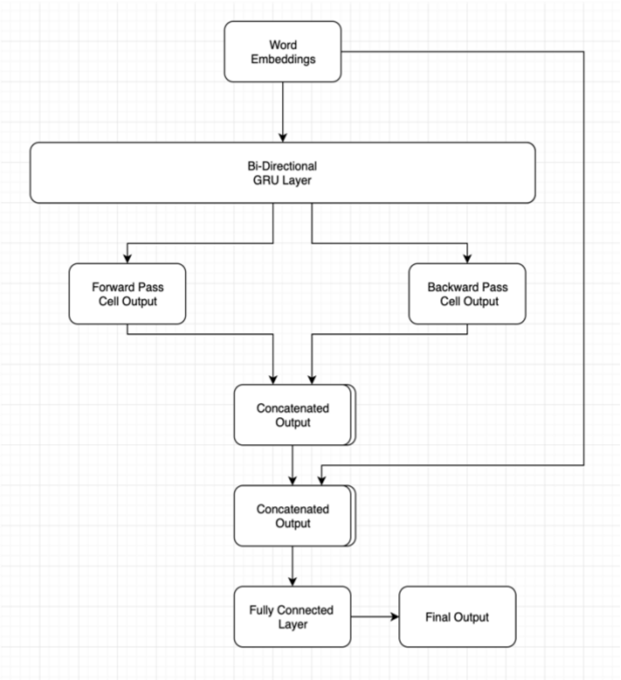

Figure 1

1. Embedding Layer:
   For every word in the language, this layer learns a vector representation for the same. The Embedding matrix learnt is of size – (vocabulary_size x size_of_word_vector). The proposed model uses word vector size as 10. A smaller vector size is used to learn quickly. The word embeddings of a sentence are looked from the embedding matrix and are passed to next layer.

2. Bi-directional RNN Layer:
   For sequence labeling tasks it is beneficial to have access to both past (left) and future (right) contexts. The basic idea is to present each sequence forwards and backwards to two separate hidden states to capture past and future information, respectively. Hence the model uses a bi-directional RNN layer with an instance of GRU Cell. The state size of the cell used is 25. A smaller state size value was giving optimal results, given the time constraints.

3. Concatenation Layer 1:
   The previous layer outputs two hidden state vectors. Each of which represents one from forward pass and other from backward pass of bi-directional RNN layer. To capture information from them, the proposed model just concatenates both the state representations.

4. Concatenation Layer 2:
   The model then concatenates the output of previous layer with the initial word embedding matrix passed into the bi-directional layer. The intuition being, the initial word representation itself would carry some information which would help in prediction. The final output matrix after this layer is of size – (batch_size x max_len x (2*state_size + size_of_word_vector)).

5. Fully Connected Layer:
   To predict the likelihood of tags for a word, the output of previous layer is passed through a fully connected layer, such that the output dimensions are- (batch_size x max_len x number_of_tags). To prevent over-fitting a L2 regularization term is added with a coefficient which is was tuned to obtain

optimal results. This output is then passed through a CRF Layer.

Learning and Prediction

Conditional Random Fields (CRF) Layer:

For sequence labeling tasks, it is beneficial to consider the correlations between labels in neighborhoods and jointly decode the best chain of labels for a given input sentence. For example, in POS tagging an adjective is more likely to be followed by a noun than a verb. Hence the proposed model uses CRF layer to predict the tags, which accounts the same.

The proposed model computes the maximum conditional likelihood estimation of tag sequences. The goal is therefore, to maximize the total likelihood estimation. Hence, the negative of the likelihood calculated is added to the total loss. The CRF layer learns a transition matrix of tags, which is of the size (number_of_tags x number_of_tags), and could be used while predicting the tags on unseen data (i.e., while CRF decoding).

Parameter optimization is performed with minibatch Adam Optimizer with batch size 10. The optimizer minimizes the total loss including the regularization loss added in the fully connected layer of the model. Optimal results were found by choosing an initial learning rate ($\eta0$) of 0.006, and the learning rate is updated on each epoch of training as $\eta t = \eta0/(1 + \rho t)$, with decay rate $\rho = 0.5$ and t is the number of epoch completed. Tuning of all the hyper-parameters was done incrementally from a certain value in order to obtain results with high accuracy within given time constraints.

Now, to predict the tags on unseen data, the model uses the transition matrix of tags which it learned during training (CRF Layer) and uses it for prediction. The prediction or decoding of tags from CRF layer can be solved efficiently by adopting Viterbi algorithm. This helped increase in accuracy as the model considered the transition probabilities of tags as well.

Results:

After training the model for about 4 epochs, very high accuracy results were observed.
The accuracy for Japanese data set was 95% and for Italian dataset an accuracy of 95.65% was obtained.