HW1 Report

Yuxiang Zhao: yuxiang2

Introduction

In this assignment, I build a word-level CNN classifier to distinguish different types of text without pre-trained word embeddings, and my highest accuracy on validation dataset is **82.89%.** The training text includes 253909 sentences from 16 different categories including "Agriculture", "food and drink", "Music", etc. My model is similar to Kim's implementation (Convolutional Neural Networks for Sentence Classification 2014). One of the main challenge in this assignment is that the model easily overfits the training set, especially without fixed pre-trained word embeddings. To overcome this challenge, I added various pre-processing and regularization techniques, and my model can stabilize around 82% validation accuracy.

Preprocessing

The training text contains a large set of vocabularies, including words in different forms, capitalized words, proper nouns, numbers, and foreign words. A large set of vocabulary will result a complex model, and thus, overfits the training data and increases the training time. To combat these, I apply various steps in preprocessing.

Firstly, I convert all characters into lower cases, so that "Dogs" will be converted to "dogs". Secondly, I lemmatize words, so that "dogs" will be converted to "dog". Thirdly, I convert all numbers into integer of their log values, that is

$$num = round(\log_{10} num)$$

After this operation, words like "1996" will be converted to "3", and "12" will be converted to "1". Lastly, I removed rare words that occur fewer than 20 times and replace with their POS tags (use nltk library), and that will remove most of the foreign words and proper nouns. During validation and test, unknown words are also replaced with their POS tags.

A comparison between original text, and text after preprocessing

Original Text: *On 25 November 1974 , at the age of 26 , Drake died from an overdose of approximately 30 amitriptyline pills , a prescribed antidepressant .*

Preprocessed: *on 1 november 3 , at the age of 1 , drake die from an overdose of approximately 1 NOUN pill , a prescribed NOUN .*

Model Structure

The first challenge of building a CNN classifier is putting sentences of various lengths into mini-batches. A simple solution is to pad all sentences into the same length. However, this requires a significant amount of padding, which may leads poor performance on shorter sentences. I rank sentences according to their lengths, and build mini-batches only with sentences of similar lengths, so there is no extensive paddings. However, the downside of this approach is that it becomes hard to keep track of which sentences have been trained, and which have not. Therefore, instead of training all the sentences once before the second epoch, I just count the number of sentences have been trained.

I have tried various neural network structure, but I have found that even a very simple model is capably of overfitting the training set, so I used a very simple model with only two CNN layers with kernel size 3 and batch normalization.

However, even this simple model can overfit the training set after around 2 epochs (500000 training sentences), So I added 2 regularization techniques. Firstly, I randomly replace 15% of the words from the training set with OOV symbol, Secondly, I add a dropout layer with drop probability=0.5 after the embedding layer. With these two regularization techniques, my model is much more stable, and its validation accuracy is stable around 82% even after 10 epochs (2500000 training sentences) of training.

Model

Embedding: 12136 words into 128 dimensions

Dropout: p=0.5

CNN 128 to 256 dimensions, Kernel_size = 3, ReLU activation, Stride = 1, Batch Norm

CNN 256 to 256 dimensions, Kernel_size = 3, Stride = 2 (use this to replace max pool)

CNN 256 to 512 dimensions, Kernel_size = 3, ReLU activation, Stride = 1, Batch Norm

Global Max Pool

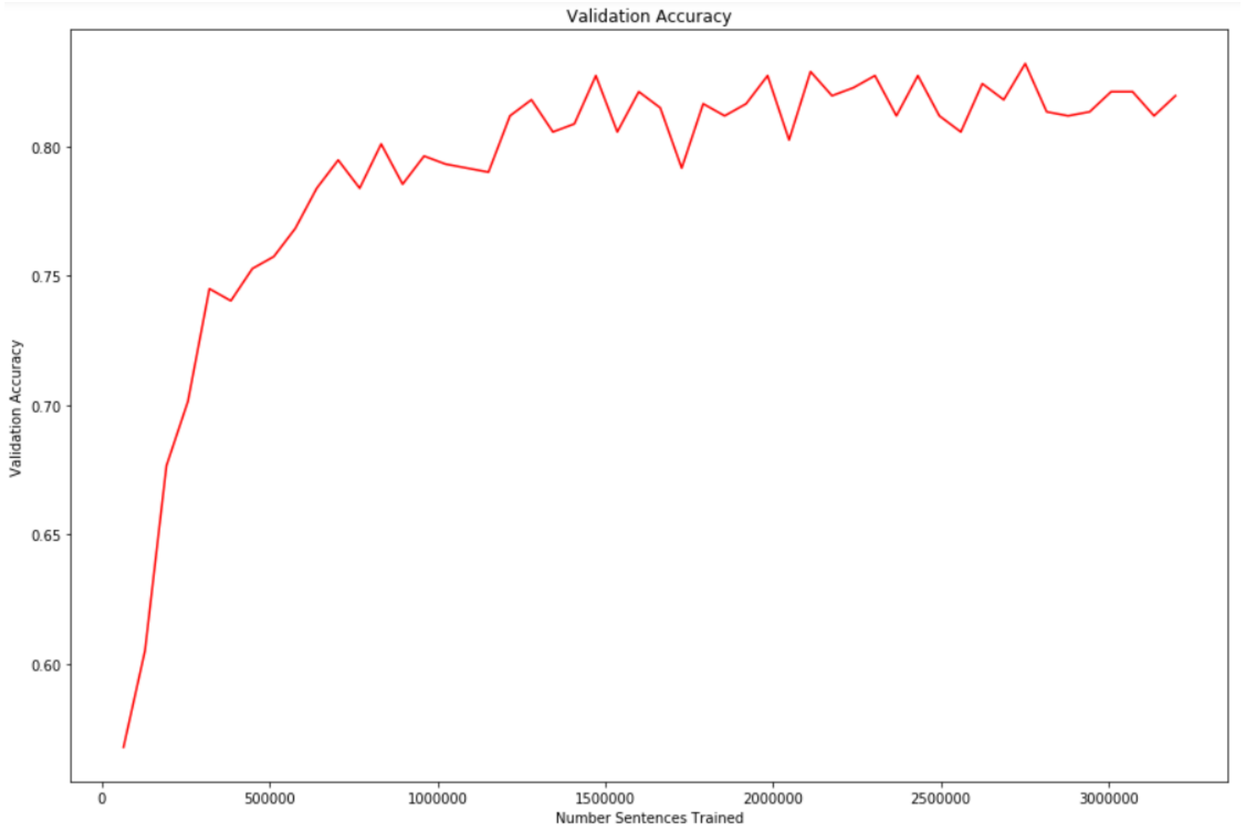
Linear Layer 512 dimensions to 512 dimensions

Linear Layer 512 dimensions to 16 dimensions

Hyperparameters

Batch Size = 32, Lr = 0.001, Random Drop Words = 15%, Adam Optimizer

Result



Validation Accuracy stabilizes around 82%. Highest Validation Accuracy = 82.89%. 3000000 sentences is about 11 epochs. However, some sentences are trained more than 11 times, and others are trained fewer than 11 times.

Analysis

Convolutional layer with kernel size 3 is able to capture features of trigrams. With two such convolutional layers and a pooling layer of stride of 2, the model is able to capture 7-grams. With more than 10000 unique words, 7-grams are enough to overfit all training examples. Therefore, preprocessing and regularization techniques to prevent overfitting is vital in this kind of task. Without these techniques, my model reaches about 71% after 2 epochs (500000 training sentences) and then its validation accuracy starts decreasing. With preprocessing, random words drop, and dropout, my model will not overfit even after 11 epochs.

Sample Validation Set Mistakes

Text	Ground Truth	Predicted Label
Later on , SpongeBob accidentally swears again .	Media and Drama	Social Sciences and Society
It is the world 's only two-row stationary Dentzel menagerie in existence .	Geography and places	Engineering and technology

His work features controversial and provocative views , written in a direct , often confrontational style .	Social sciences and society	Media and drama
Churchill was fired on July 24 , 2007 , leading to a claim by some scholars that he was fired because of the " Little Eichmanns " comment .	Social sciences and society	History

Among these mistakes, even a human may be confused sometimes. Many sentences provided contain too little information to determine its label (the 1st example). Other times, the ground truth label and the predicted label are indeed very similar, and may even confuse humans (2nd and 3rd example). Overall, the model successfully classify most of the texts, and a large portion of the mistakes it makes are unavoidable.

References

Kim, Yoon. "Convolutional neural networks for sentence classification." *arXiv preprint arXiv:1408.5882* (2014).