Combining Convolution and Recursive Neural Networks for Sentiment Analysis

This paper addresses the problem of sentence-level sentiment analysis. In recent years, Convolution and Recursive Neural Networks have been proven to be effective network architecture for sentencelevel sentiment analysis. Nevertheless, each of them has their own potential drawbacks. For alleviating their weaknesses, we combined Convolution and Recursive Neural Networks into a new network architecture. In addition, we employed transfer learning from a large document-level labeled sentiment dataset to improve the word embedding in our models. The resulting models outperform all recent Convolution and Recursive Neural Networks. Beyond that, our models achieve comparable performance with state-of-the-art systems on Stanford Sentiment Treebank.

CNN Long Short-Term Memory Networks

Input with spatial structure, like images, cannot be modeled easily with the standard Vanilla LSTM.

The CNN Long Short-Term Memory Network or CNN LSTM for short is an LSTM architecture specifically designed for sequence prediction problems with spatial inputs, like images or videos.

In this post, you will discover the CNN LSTM architecture for sequence prediction.

After completing this post, you will know:

- About the development of the CNN LSTM model architecture for sequence prediction.
- Examples of the types of problems to which the CNN LSTM model is suited.
- How to implement the CNN LSTM architecture in Python with Keras.

How to implement Sentiment Analysis using word embedding and Convolutional Neural Networks on Keras

IMDb has released a database of 50,000 movie reviews classified in two categories: Negative and Positive. This is a typical sequence binary classification problem.

In this article, I will show how to implement a Deep Learning system for such sentiment analysis with ~87% accuracy. (State of the art is at 88.89% accuracy).
Here we used a 3-layered convolution neural network with 2 dense layers.

Why Convolutional? Because it works. Convolutional layers are really powerful to extract higher level feature in images. And quite amazingly, they actually work in most 2D problems. Another big reason that should convince you is the training time, CNN train 50% to 60% faster than LSTMs on this problem.

Convolutional Neural Networks for Sentence Classification

We report on a series of experiments with convolutional neural networks (CNN) trained on top of pre-trained word vectors for sentence-level classification tasks. We show that a simple CNN with little hyperparameter tuning and static vectors achieves excellent results on multiple benchmarks. Learning task-specific vectors through fine-tuning offers further gains in performance. We additionally propose a simple modification to the architecture to allow for the use of both task-specific and static vectors. The CNN models discussed herein improve upon the state of the art on 4 out of 7 tasks, which include sentiment analysis and question classification.

Code: https://github.com/dennybritz/cnn-text-classification-tf

Text Classification Using CNN, LSTM and visualize Word Embeddings

This is a part of series articles on classifying Yelp review comments using deep learning techniques and word embeddings. In the last part (part-1) of this series, I have shown how we can get word embeddings and classify comments based on LSTM. In this part, I use one CNN layer on top of the LSTM for faster training time.

Followings are the list of brief contents of the different part :

Part-1: In this part, I build a neural network with LSTM and word embeddings were leaned while fitting the neural network on the classification problem.
Part-2: In in this part, I add an extra 1D convolutional layer on top of LSTM layer to reduce the training time.
Part-3: In this part-3, I use the same network architecture as part-2, but use the pre-trained glove 100 dimension word embeddings as initial input.
Part-4: In part-4, I use word2vec to learn word embeddings.