Deep Learning for Document Classification

Amlan Kar, Sanket Jantre

Problem Statement

Explore how a CNN can work with pre-trained semantic embeddings to model data for various Document Classification tasks. We specifically look to produce classifiers for sentiment analysis and try to fine-tune our pre-trained vectors to produce robust task specific vectors.

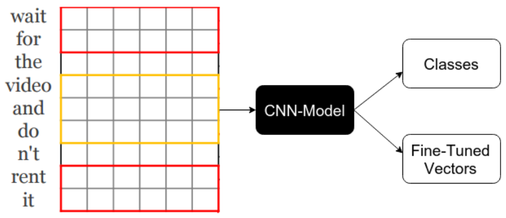

Fig.1: Problem Statement

Theory

CNN:
A Convolutional Neural Network is an artificial neural network which work by sliding windows through it's input looking for local features. These have shown to work extremely well for Image recognition tasks and recently have shone in NLP as well.[1]

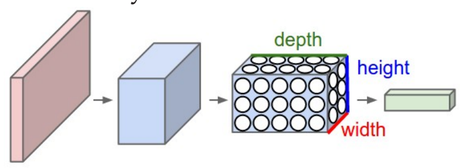

Fig.3: CNN Illustration [a]

[a]Image taken from www.cs231n.github.io

Semantic Embeddings:
Recent advances such as word2vec, GloVe,[2] skip-thoughts[3] map words or sentences to high dimensional real valued vectors such that syntactic relation between the words are preserved. These have been shown to have strong semantic similarity properties as well.

Dropout:[4]
A way to prevent neural nets from overfitting. Basically every node in the neural net is given a probability with which it could be present in the net during a training epoch. It has been shown to act as an excellent regularizer for neural nets.

Results

We trained our CNN for 25 epochs on the Pang-Lee movie review dataset[5] and the Hindi-700 movie review dataset created by Pranjal Sharma for his M.Tech Thesis. We report 10-fold CV results below and compare with the state of the art.

Method	Accuracy
Socher2012	79.0
Dong2014	79.5
Kim2014	81.5
This Method	**81.8**

Table 1: Sentiment Classification on Pang-Lee dataset

Method	Accuracy
Pranjal2014	**0.91**
Kalchbrenner2014[a]	0.71
This method	0.70

Table 2: Sentiment Classification on Hindi-700 dataset

[a]The experiment was carried out by Jayesh K.G and Arpit S. as a course project for CS365

Methodology

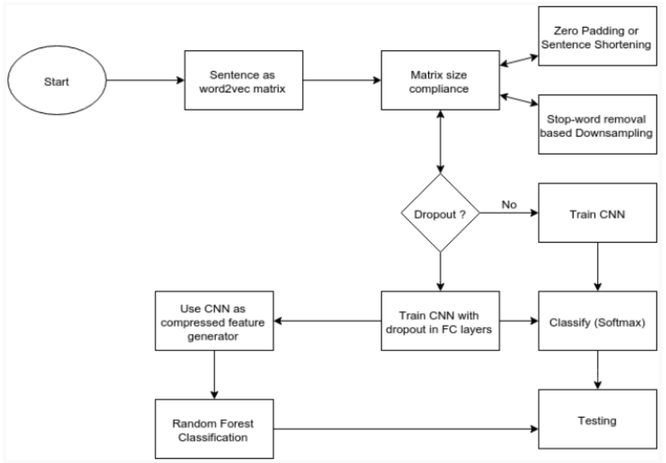

Fig.3: Flowchart

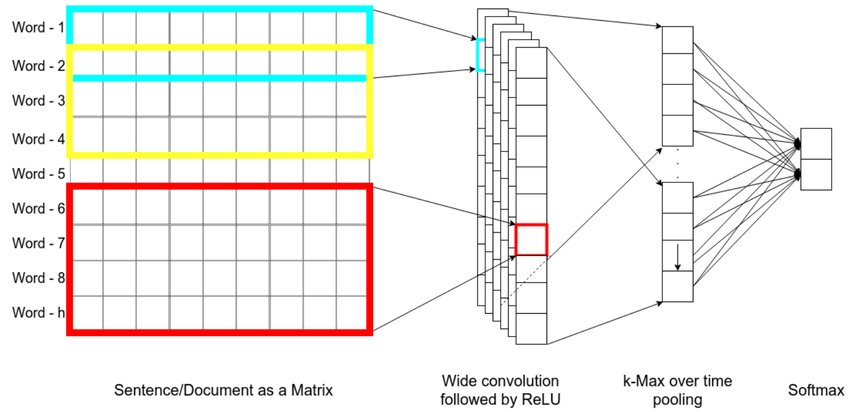

Fig.4: ConvNet Structure

Conclusions

1. The accuracies obtained for a simple CNN are surprisingly high. This leaves high expectation for the time when a deeper CNN could be trained with a much bigger dataset.
2. The Hindi classification task doesn't perform well. This could be attributed to the lack of availability of word vectors as good as the english-300 word vectors. In fact, 1/3rd of the vocabulary of the 700 dataset was missing in the word vector vocabulary.

Future Work

1. Use this approach for multi-class document classification.
2. Deeper network with a huge dataset.
3. Use the CNN as a feature generator to train other standard classifiers.

References

[1] Yoon Kim. Convolutional neural networks for sentence classification. *EMNLP 2014*, 2014.

[2] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, pages 3111–3119, 2013.

[3] Ryan Kiros, Yukun Zhu, Ruslan Salakhutdinov, Richard S Zemel, Antonio Torralba, Raquel Urtasun, and Sanja Fidler. Skip-thought vectors. *arXiv preprint arXiv:1506.06726*, 2015.

[4] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: A simple way to prevent neural networks from overfitting. *The Journal of Machine Learning Research*, 15(1):1929–1958, 2014.

[5] Bo Pang, Lillian Lee, and Shivakumar Vaithyanathan. Thumbs up?: sentiment classification using machine learning techniques. In *Proceedings of the ACL-02 conference on Empirical methods in natural language processing-Volume 10*, pages 79–86. Association for Computational Linguistics, 2002.