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.

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]

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.

Table 1: Sentiment Classification on Pang-Lee dataset

<table>
<thead>
<tr>
<th>Method</th>
<th>Accuracy</th>
</tr>
</thead>
<tbody>
<tr>
<td>Socher2012</td>
<td>79.0</td>
</tr>
<tr>
<td>Dong2014</td>
<td>79.5</td>
</tr>
<tr>
<td>Kim2014</td>
<td>81.5</td>
</tr>
<tr>
<td>This Method</td>
<td>81.8</td>
</tr>
</tbody>
</table>

Table 2: Sentiment Classification on Hindi-700 dataset

<table>
<thead>
<tr>
<th>Method</th>
<th>Accuracy</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pranjal2014</td>
<td>0.91</td>
</tr>
<tr>
<td>Kalchbrenner2014*</td>
<td>0.71</td>
</tr>
<tr>
<td>This method</td>
<td>0.70</td>
</tr>
</tbody>
</table>

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

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