Mid Report Summaries

# GloVe

## Technical Contributions

* A log-bilinear model that merges local context window methods with global matrix factorization
* Weighted least squares model trained on global word-word co-occurrence
* The goal of training is to learn word vectors such that their dot product equals the logarithm of the words' probability of co-occurrence.
* Training:
  + Build a co-occurrence matrix (X) with a fixed window size, where Xij represents how often the word i appears in the context of the word j
  + Uses co-occurrence ratios between two words in a context (assumes this is strongly connected to the meaning of the words)

## Strengths

* The model generates accurate word representations (75% performance on analogy task)
* Outperformed other methods on the word analogy task (Mikolov et al. (2013a))
* Uses global count statistics in addition to local word information

## Weaknesses

* Requires a lot of memory - needs dimension reduction
* Empirical results show that the performance of GloVe is very similar to that of word2vec, despite the different approaches of the models and the results showed in the paper (which show GloVe outperforming word2vec in both training times and accuracy)

## Improvements

* RGloVe: Uses cosine similarity between entity vectors instead of dot product to measure the entity occurrences, which more easily reaches a local optimum. Results on a corpus of Sina News show RGloVe outperforming GloVe.

# 

# fastText

## Technical Contributions

* Modified a linear classifier to work with a large dataset
* Uses rank constraint on a linear classifier
* Probability distribution calculated using a softmax function
* Improved running time using hierarchical softmax, based on Huffman coding tree - complexity went down from O(kh) to O(h log2 k)
* Using depth-first search increases speed by not exploring small probability branches
* Bag of n-grams used to retain information about local word order
* Uses skip-gram with negative sampling (sub-words are positive examples and random samples from dictionary are negative examples)

## Strengths

* Relatively much faster than conventional neural networks (both training and testing)
* Accuracy on par with neural network-based models, further improved by adding bigram information
* N-grams help capturing the meanings of shorter words, including prefixes and suffixes

## Weaknesses

* Cannot capture different contexts

## Improvements

* FastText.zip: An improved version of fastText designed to fit in limited memory. Employs a method based on product quantization to store embeddings. By further We consider the problem of producing compact architectures for text classification, such that the full model fits in a limited amount of memory. After considering different solutions inspired by the hashing literature, we propose a method built upon product quantization to store word embeddings. By further avoiding quantization artefacts, reduces memory requirements by half with only a small loss in accuracy

## Questions

* Are the methods of evaluation used good enough to evaluate deep learning models? - *from the paper’s conclusion*