Efficient Estimation of Word Representations in Vector Space

This paper introduces the Continuous Bag-of-Words model and the Continuous Skip-gram model for natural language processing and compares them to more traditional models. The goal of the paper is to introduce techniques that can be used to train models over much larger datasets of words. These new techniques will be focused on building vectors that can be used to relate similar words and then using those results to build the NNLM.

The first model they look at for comparisons is Feedforward Neural Net Language Model. The feedforward NNLM uses a projection layer between input and hidden layers. They use a Huffman binary tree representation of the vocabulary with a hierarchical version of softmax to keep the Feedforward NNLM efficient. The other model they look at for comparisons is the Recurrent Neural Net Language Model. This NNLM does not use a projection layer but uses a recurrent matrix in the hidden layer that connects the hidden layer to itself, this allows for a sort of short-term memory, updating the hidden layer with the current input for each time step. They combine these models and their new proposed models with parallel training. This allows them to run the models on distributed processors with DistBelief. They use an adaptive learning rate procedure with mini-batch gradient descent procedure, which is called Adagrad.

The first new model they propose is called Continuous Bag-of-Words. They use a model similar to the feedforward NNLM, but it uses a shared projection layer, projecting all words into the same position, therefore the order of words does not matter. It is called continuous because it uses a continuous distributed representation of the context. The second new model is called Continuous Skip-gram. This model tries to predict words based on another word in the sentence. Each word is used as an input for a log-linear classifier with a continuous projection layer, that allows them to predict words a certain range before and after their current word. They found that increasing the range both increases complexity and the quality of the resulting vectors.

They compared the ability of the resulting vectors from using each model to predict similar words. They used 2 sets of words with common relationships to one another and tried to predict the second word by subtracting the first from the second and adding the other set's first word to get a resulting vector close to the second word they were trying to predict. They used several different dimensionalities of the word vectors and sizes of the training dataset. Finding that at a certain point, increasing only 1 of those at a time resulted in diminishing returns. The found that Skip-gram worked significantly better than CBOW or either NNLM on semantic accuracy. They also found all but RNNLM were similar for syntactic accuracy. The dataset used was 320M words with 82K vocabulary, and 640 hidden units in the parallel training. When comparing to publicly available word vectors they found a very large improvement in semantic training and a large improvement in syntactic training over other models.

They also showed that they can improve on the Microsoft Research Sentence Completion Challenge by using Skip-gram and RNNLMs. They had a few percentage points improvements over the existing RNNLMs. They were able to show that they can train word vectors with very simple architectures and therefore can train models over larger vocabularies and datasets while still achieving similar accuracy.

Distributed Representations of Words and Phrases and their Compositionality

This paper focuses around the extensions of original implementation of Skip-Gram model and reduce the training time. The paper talks about uses the distributed representation to improve the vector representations.

Two significant changes are introduced to facilitate the desired improvements. Authors put in use the sub-sampling of frequent words which helps in speeding up the training time and improve representation for less frequent words. To further improve the model, they introduce a simplified version of Noise Contrastive Estimation (NCE) which works more efficiently than complex Hierarchical softmax. Author also believes using vectors to represent the whole phrases makes the Skip-gram model considerably more expressive. In this approach we consider phrases as a single token during training.

Skip-gram is used to predict the context of the word for a given word. We take a word as an input and return list of words that form the context for the given word. It is basically reverse of the Continuous Bag of Words. The main thought behind the Skip-Gram model is this: it takes each word in a big corpus (we will think of it as the middle word) and moreover takes independently the words that incorporate it inside a described 'window' to then deal with a neural association that ensuing to getting ready will expect the probability for each word to truly appear in the window around the middle word.

In hierarchical softmax, we modify the regular softmax algorithm so that the can reduce the computational complexity from O(n) to O($\log_2$ n). This drastically reduces the number of operations making the model even faster and more efficient. This is achieved with the use of binary tree and leaves of this tree represent the probability of words. Thus instead of a linear output model we have a binary tree which acts as the output layer of the network.

To evaluate the model, authors used an internal google dataset with one billion words. They preprocessed the data and removed words with frequency fewer than five words. Authors concluded Negative Sampling (simplified NCE) out performed Hierarchical Softmax and performed even better than original (NCE). They also stated that vectors generated by skip-gram model are more suitable for linear analogical reasoning. To conclude, authors suggested that chosing methodology is based on task specific decision.

GloVe: Global Vectors for Word Representation

This paper attempts to combine the advantages of global matrix factorization with local context window methods that capture semantic and syntactic regularities. Their new proposed model produces a global vector space that scores well on word analogy tasks. They explain that the new model proposed in the other 2 papers is very useful and novel for its ability to create a more meaningful vector space for relating words to one another, but this new model loses the global nature of other models. Skip-gram cannot fully utilize statistics of the corpus because it is trained on local contexts rather than the global co-occurrence counts. This paper will then analyze the characteristics of the 2 models and attempt to create a new model that incorporates both. This model will be a global log-bilinear regression, with specific weighted least squares to create a global word-word co-occurrence count while still producing a meaningful vector space that can perform well on word analogy problems.

They begin by examining these two types of models. They examine first the matrix factorization methods and how the matrix is structured with words and co-occurrence. They also consider some methods to deal with the problems of this method such as the co-occurrence of very common words. They then examine shallow window-based methods. These models either try to guess the word's context from the word itself or to guess the word from its context. They demonstrated that models could learn linguistic patterns expressed as linear relationships of the word vectors. Since these methods do not rely on the co-occurrence statistics of the entire corpus, they cannot take advantage of the repetition of the data.

The GloVe model is the new model they propose. They create a matrix of word-word co-occurrence counts and can use ratios to determine if words are related, inversely related, or have no relation based on the ratios. The main drawback they find is their model weighs all co-occurrences equally and to prevent that they use a weighted least squares regression model. In their formulas they fix xmax to 100 and alpha to ¾ which turns out to be close to alpha in the other papers. Next, they explain the similarities between their function and the function used for skip-gram. They also explain some of the issues with the skip-gram model and how their usage improves upon it. They also explain that for the worst-case scenario, their algorithm is O(—V—2) complexity but really ends up more like O(—C—0.8) which is better than skip-gram.

They show that the GloVe model does better on semantics than other models and does better syntactically on the largest datasets. For this testing they trained models with a variety of corpora, they lowercase each corpus and tokenize with the Stanford tokenizer, building a vocabulary of the 400,000 most common words. They weight word occurrences with 1/d with d being the number of words apart they are. They used AdaGrad with a learning rate of 0.05 and set xmax and alpha to 100 and ¾ respectively. They run 100 iterations for vectors larger than 300 dimensions and 50 iteration for smaller vectors. They then combine the 2 resulting vectors for a small boost in performance.

They report their results and show that GloVe performs better even with smaller corpora or vector dimensions. They also show the results over 5 different word similarity datasets and that GloVe preforms better. They also show the results of a NER task with CRF-based models and how much better GloVe performs on that test. They then show that there are diminishing returns for vector dimensions over 200 and the syntactic tasks perform better with an asymmetric context. They show that larger corpora improve syntactic tasks but not semantic tasks and posit this is due to the quality of the corpora themselves. They explain that loading X with a 10-word symmetric context, 400,000 word vocabulary, and a corpus of 6 billion words takes 85 minutes on their 32 core Intel Xeon machine. And that same machine takes 14 minutes per iteration for 300-dimension vectors. That comes out to 1485 minutes or 24 hours and 45 minutes for 100 iterations plus the loading of X. They also talk about the limitations of the analysis, they set certain parameters for training the models to compare but the main variation for GloVe being iterations and for word2vec was epochs. Unfortunately, word2vec was written for a single epoch and expanding it was out of scope of this analysis. They talk about increasing the negative sampling and how that can improve word2vec and they graph the results based on training time. They find that skip-gram performs worse after too many negative samples and there are decreasing improvements for GloVe over a certain amount of iterations/time.

They conclude that GloVe captures the usefulness of both co-occurrence statistics on the corpus and meaningful linear substructures that recent models like word2vec have captured. GloVe is "a new global log-bilinear regression model for the unsupervised learning of word representations" and that it outperforms the other models in word analogy and similarity tasks and in named entity recognition.