**Efficient Estimation of Word Representations in Vector Space**

**Technical Contribution**

This paper introduces two new model architectures for learning distributed representation of words that improve on the accuracy and minimize computational costs. The aim of these models is to be able to represent the data as precisely as neural networks, but without the complexity of the hidden non-linear layer, in order to be trained on the data more efficiently. The two model architectures are Continuous Bad-of-Words (CBOW) and Skip-gram. This paper also compared the results of the two proposed methods with two previously used architectures: NNLM and RNNLM.

In the Continuous Bag-of-Words Model, the hidden layer is removed and the projection layer shared for all words. Thus, all words are projected on the same position.

In CBOW, a log linear classifier is used with a certain number of words before the target word and the same number of words after the target word which are used as the context to project the target word(i.e. the current word is predicted based on the context).

In the Continuous Skip-gram Model, unlike CBOW, the current word is used to predict the context. In other words, each current word is used as an input to a log-linear classifier with continuous projection layer to predict words within a certain range before and after that current word.

**Comparison between the model architectures revealed:**

* NNLM vectors performed significantly better than the RNNLM vectors as the vectors in RNNLM are directly connected to a non-linear hidden layer.
* The CBOW architecture worked the same as the NNLM on the semantic task but worked better at the syntactic one.
* The Skip-gram model works slightly worse on syntactic task than the CBOW model but works much better than all the models on the semantic task.

(Comparison performed on the Semantic-Syntactic Word Relationship test set, using 640-dimensional word vectors)

**Strengths**

* The CBOW and the Skip-gram models are easier to train and they take a lot less time because of the much lower computational costs as they do away with the non-linear hidden layer like the previous architectures.
* These two models can be trained on huge datasets with billions of words with millions of words in the vocabulary.
* Increasing the range (words before and after the target word) improves quality of words in Skip-gram, but it also increases the computational complexity.
* The CBOW works better than Skip-gram at syntactic tasks but Skip-gram outperforms CBOW on the semantic tasks.

**Weaknesses**

* These word2vec models (CBOW and Skip-gram) are context-independent which means that words like **cell** which could mean prison, phone, etc are represented as a single vector.
* Skip-gram takes longer time to train in comparison to CBOW (on single machines).
* Inability to handle out-of-vocabulary words

**Improvements**

* Improve the model to capture the different contexts that a word may appear in.