# How to make a model learn Word Representations in FastText

FastText Tutorial – We shall learn how to make a model learn Word Representations in FastText by training word vectors using Unsupervised Learning techniques.

## Learn Word Representations in FastText

For training using machine learning, words and sentences could be represented in a more numerical and efficient way called Word Vectors. FastText provides tools to learn these word representations, that could boost accuracy numbers for text classification and such.

## Input Data

Unlike supervised learning, unsupervised learning doesn't require labelled data. So, any of the word dumps could be used as input data to train the model for learning word representations. For example, you may find many dumps from wiki at [https://dumps.wikimedia.org/enwiki/latest/], if you want to try training your model with huge amount of data corpus.

For this tutorial, we shall use sample data as shown below :

```
Text Classification is one of the important NLP (Natural Language Processing) task with

FastText provides "supervised" module to build a model for Text Classification using Sup

To work with fastText, it has to be built from source. To build fastText, follow the fas
```

But, please remember that, for any useful model to be trained, you may need lot of data corpus w.r.t your use case, at least a billion words.

## Train model to learn Word Representations

To train word vectors, FastText provides two techniques. They are

- Continuous Bag Of Words (CBOW)
- SkipGram

# Training Continuous Bag Of Words (CBOW) Model

Following is the syntax to train word vectors using CBOW model.

```
$ ./fasttext cbow -input <input_file> -output <output_file>
```

## Example

We shall use the data in a text file that is provided in the input data section, as training data.

```
$ ./fasttext cbow -input wordRepTrainingData.txt -output cbowModel
```

```
$ ./fasttext cbow -input wordRepTrainingData.txt -output cbowModel
Read 0M words
Number of words:  2
Number of labels: 0
Progress: 100.0%  words/sec/thread: 33  lr: 0.000000  loss: 0.000000  eta: 0h0m
```

**cbowModel.bin** is created after training.

# Training SkipGram Model

Following is the syntax to train word vectors using CBOW model.

```
$ ./fasttext skipgram -input <input_file> -output <output_file>
```

## Example

We shall use the data in a text file that is provided in the input data as training data.

```
$ ./fasttext skipgram -input wordRepTrainingData.txt -output cbowModel
```

```
$ ./fasttext skipgram -input wordRepTrainingData.txt -output skipGramModel
Read 0M words
Number of words: 2
Number of labels: 0
Progress: 100.0% words/sec/thread: 27 lr: 0.000000 loss: 0.000000 eta: 0h0m
```

**skipGramModel.bin** is created after training.

## Print Word Vectors

Once the model is generated, we shall have a look on how to calculate word vectors for some input words :

Example : Calculate word vector for the word “Classification”

```
$ echo "Classification" | ./fasttext print-word-vectors cbowModel.bin
Classification -0.0016351 -0.00038951 -0.00069403 -0.00055687 4.6813e-05 0.00084484 -0.0
```

## Print Sentence Vectors

We could also calculate sentence vectors using the CBOW and SkipGram models that we generated.

Example: Calculate sentence vector for the sentence “Text Classification”

```
$ echo "Text Classification" | ./fasttext print-sentence-vectors cbowModel.bin
Text Classification -0.10849 0.0073465 0.010102 -0.063361 -0.059639 0.056901 -0.06169 -0
```

We have printed word and sentence vectors using CBOW model. You may try with SkipGram model as a practice. All you need to do is providing skipGramModel.bin instead of cbowModel.bin in the commands.

### Conclusion

In this FastText Tutorial, we have learnt to make a model learn Word Representations in FastText using Unsupervised Learning techniques – CBOW (Continuous Bag of Words) and SkipGram. And also calculate word vectors for words and sentences.

**FastText Tutorial**

✦ FastText Tutorial Home

✦ FastText - BUILD

✦ FastText - Train and Test Supervised Text Classifier

➩ **Learn Word Representations in FastText**

**FastText Python**

✦ FastText Python - Learn Word Representations