**Day 32**

**What to do?**

Learn about word2vec embeddings and how to build them using CBOW and Skip – Gram.

**Word2vec:**

As it was seen yesterday, word embeddings are a way to represent a word numerically. However, to implement it, word2vec technique is used. Usually in word embedding, unique words are one hot encoded. In other words, they are assumed to be independent of one another. Let us say that there are two sentences “Amazing food” and “Great food”. We as humans know that “amazing” and “great” are not so different semantically, but word embeddings do not. It would create two different vectors. Hence, word2vec is utilized to create dependence among similar semantic words. It can be built using either CBOW or Skip – Gram.

**CBOW:**

Consider the text “Amazing food”. Say the target word is “food”, and the feature are “amazing”. To predict the target word with only one-word feature, the one – hot encoded vector is given as input and passed through hidden layer with no activation functions. The output layer is given softmax activation with the same size as input.

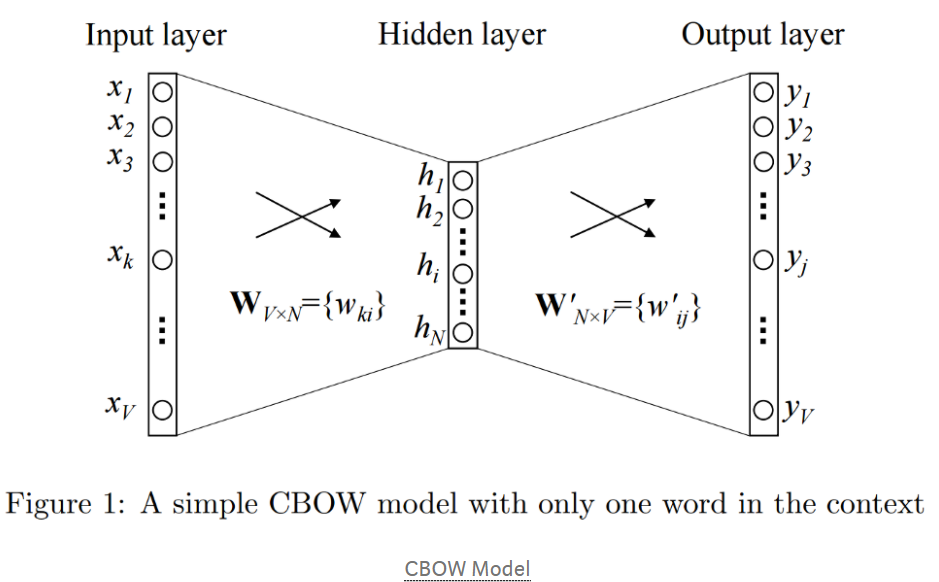


Fig 1a. CBOW with one feature

If there are more than two words (more than two features), then the process repeats, as follows.

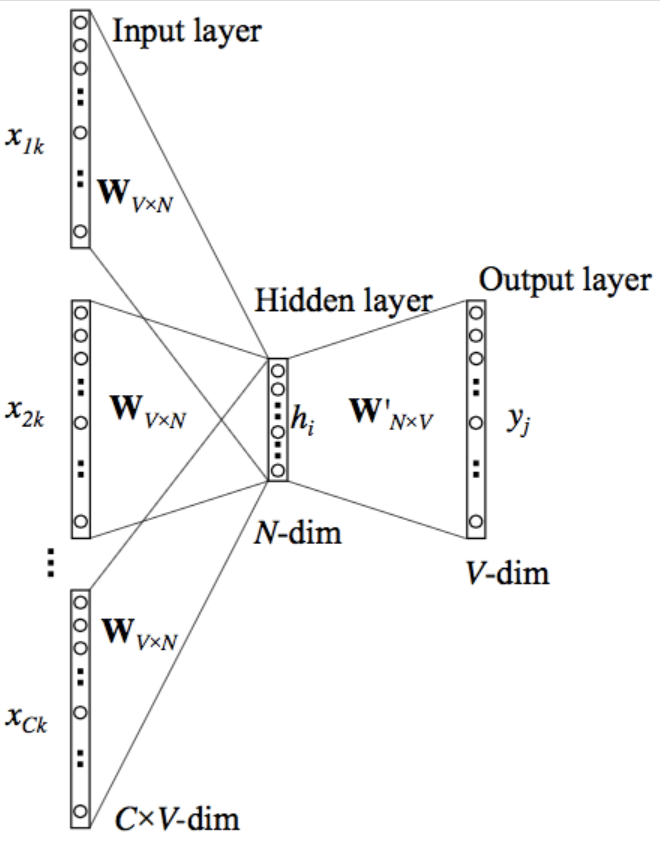


Fig 1b. CBOW with more than one feature

This technique works well with large data and for more frequent words.

**Skip – Gram:**

This technique is the reverse of CBOW. So, the target feature is given as the input to predict the context. Say there are 3 features and 1 target: Have a great day. The word vector “day” is given as input, and the three-word vectors “Have a great” is to be predicted by the predicted probabilities.

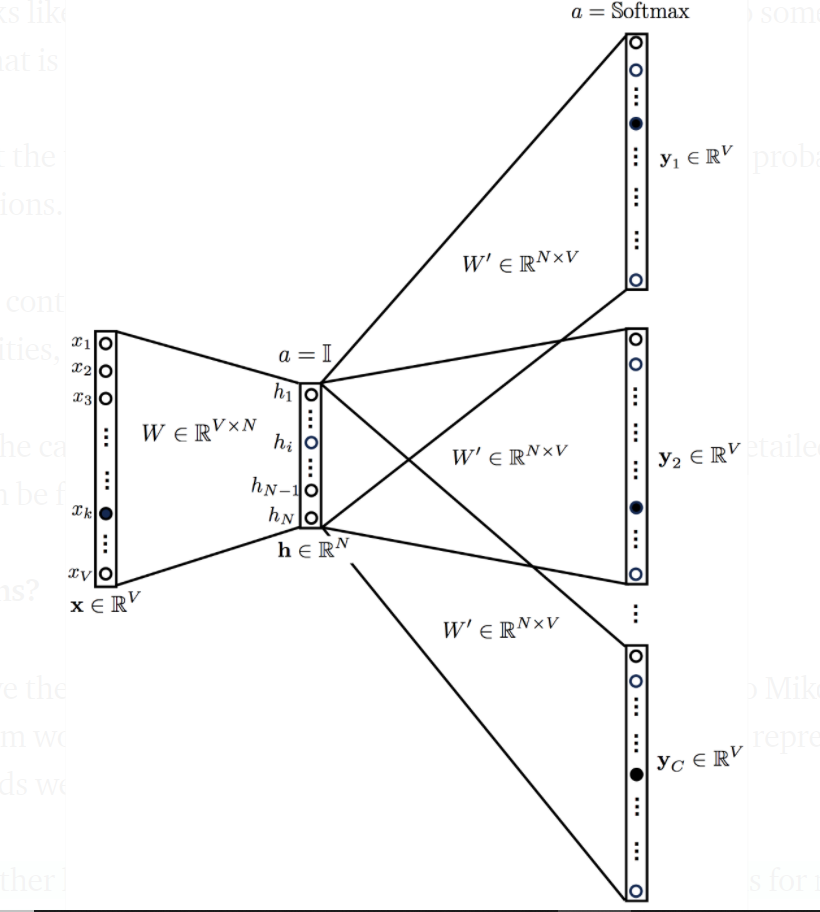


Fig 2. Skip – gram network

This technique works well with small amount of data and with rare words.