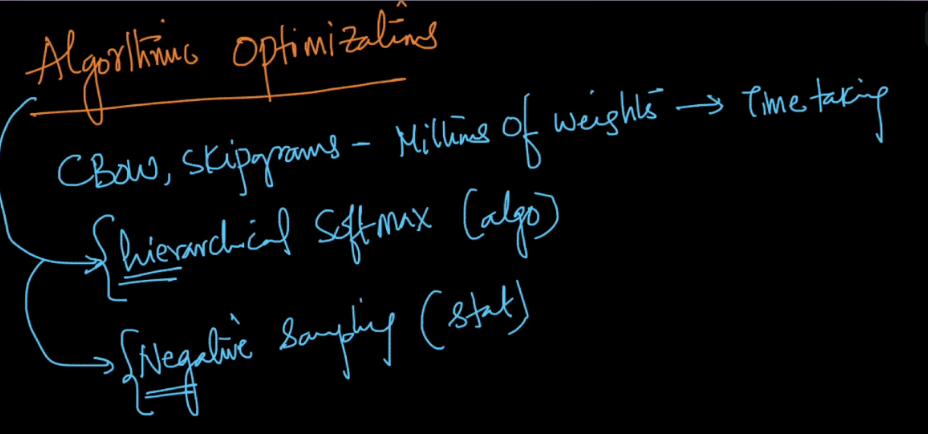
Since there are millions of weights in CBOW and skipgram so it will tak a huge time to train, therefore we need some optimization.

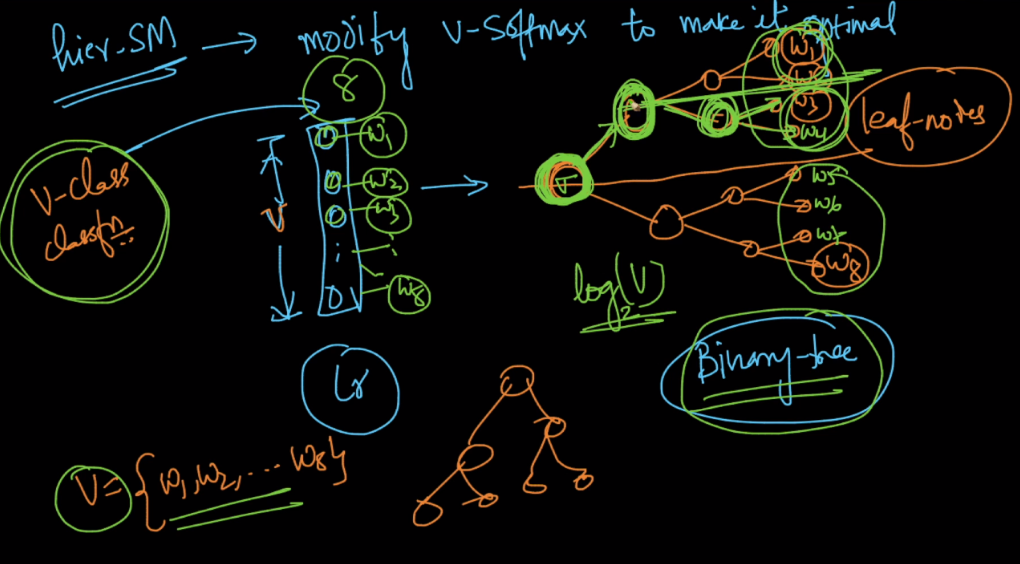


**Hierarichal softmax(algorithm):** let’s take CBOW here we are using softmax classifier to predict one word out of thousands of words, that means it’s finding the probability of all thousands of word, and whichever has bigger prob, will be treated as output.

So the optimization idea is lets say we have only 8 words in vocab. So we create a binary tree where all the words present on leaf nodes. Now instead of finding probability of all 8 words we find the probability at first node of tree, that wheter word belongs to left or right side of the tree.

Suppose left has more prob, now we find prob for 2nd node, now suppose right child has more prob, now we find prob for 3rd selected node, and this will give one word out of two.

So here we are finding only 3 probabilities instead of prob for all 8 words. That means only log2(v) probabilities.

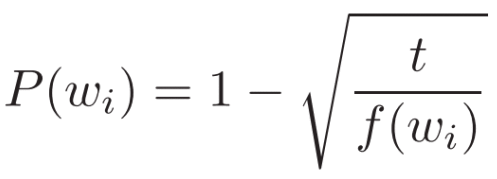


**Negative Sampling(statistic based):**

The idea is we don’t update all the weights instead we update weights of only a sample of words per iteration.

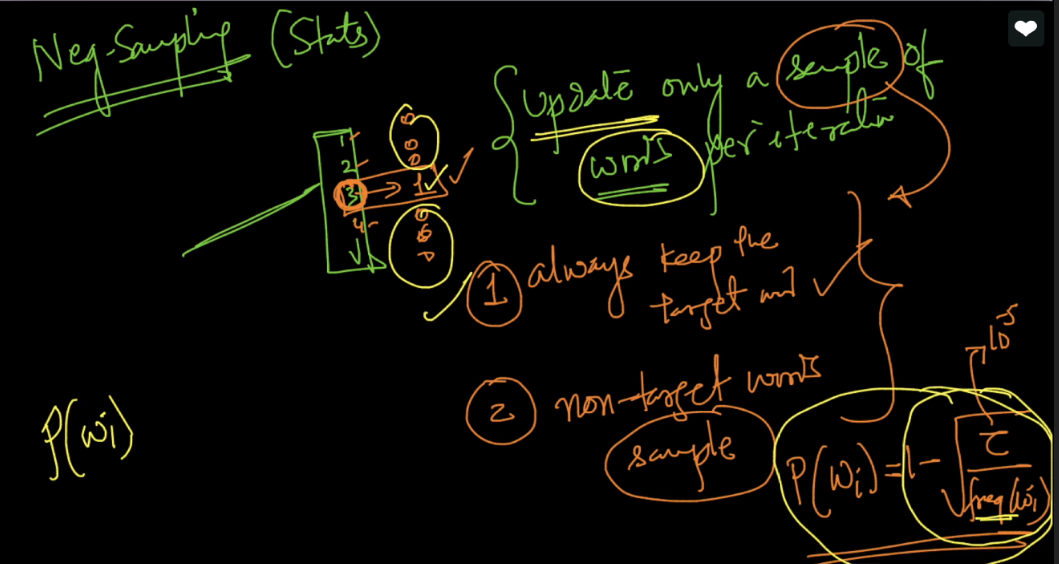
It updates weights of following words.

1. Always update weights of the target word.
2. And for non target word we select a sample of words using formula



Here f(w) is the frequency of the word, there if frequency is high then the under root term becomes smaller 1-underroot becomes greater.

The idea is words like “in”, “the”, and “a” provide less information value than rare words, so these words will be given lower probabilities and hence won’t be present in selected sample.



<https://blog.acolyer.org/2016/04/21/the-amazing-power-of-word-vectors/>