Prawing word2vec

Kai Sasaki

Training Algorithm

- * CBOW
- * Skip-gram
- * Hierarchical Softmax
- * Omit Negative Sampling this time: (

- * Continuous Bag of Words
 - * Disregard grammar and work order
 - * Share the weight of each words
 - * Training around words

syn0[i-1]

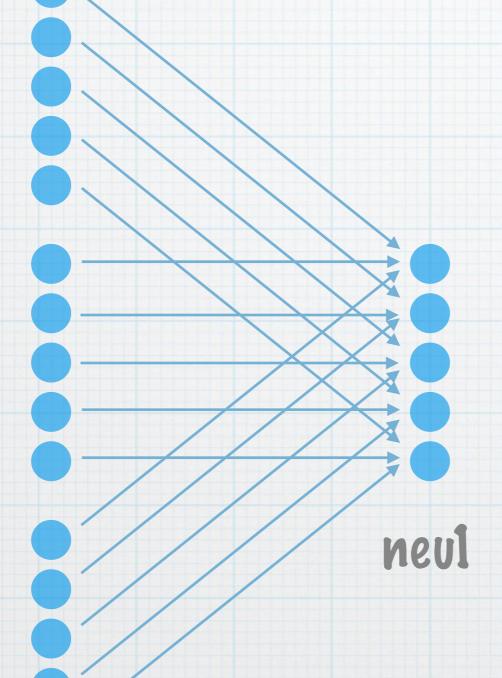
synoli]

syn0[i+1]

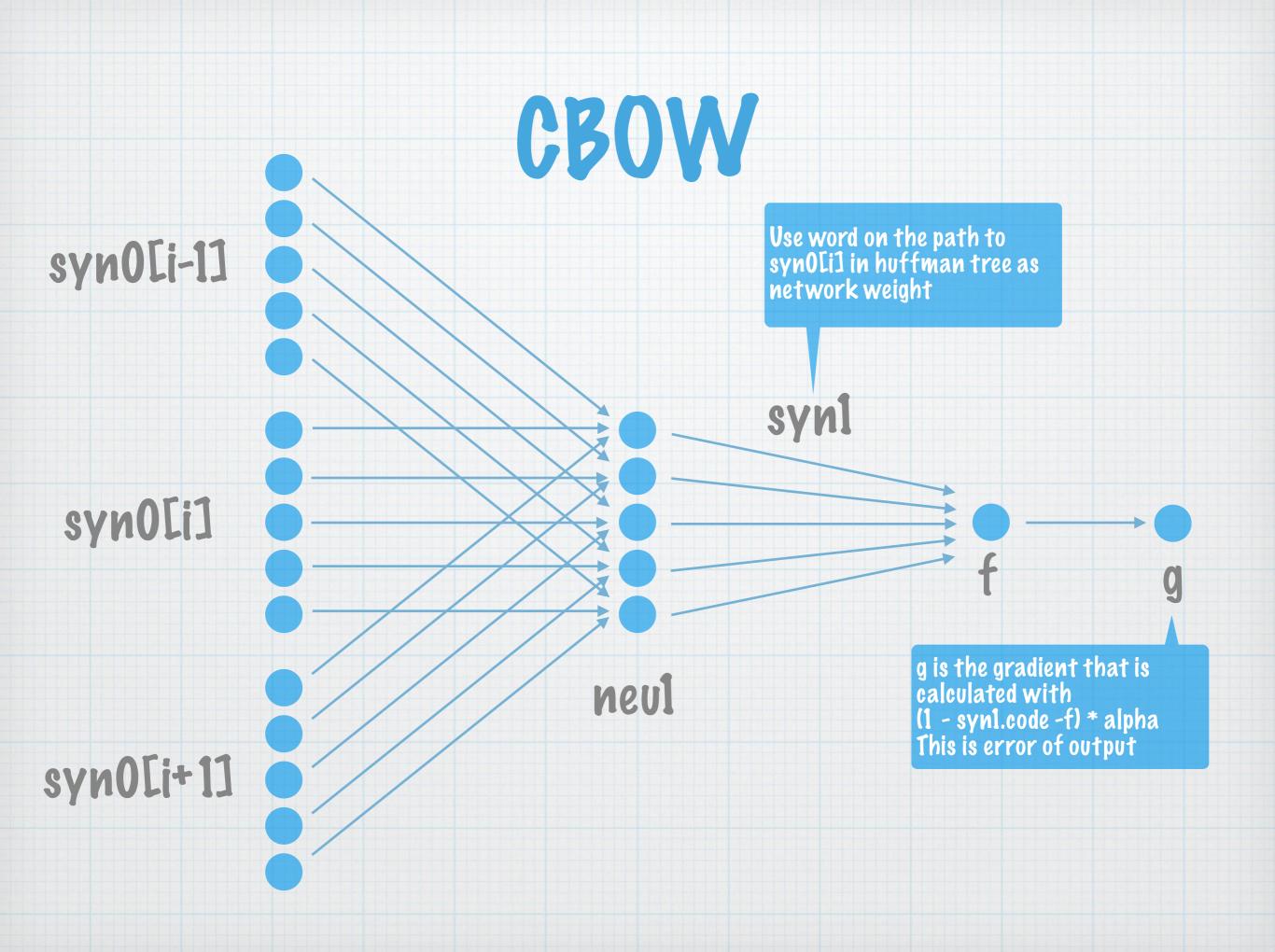
syn0[i-1]

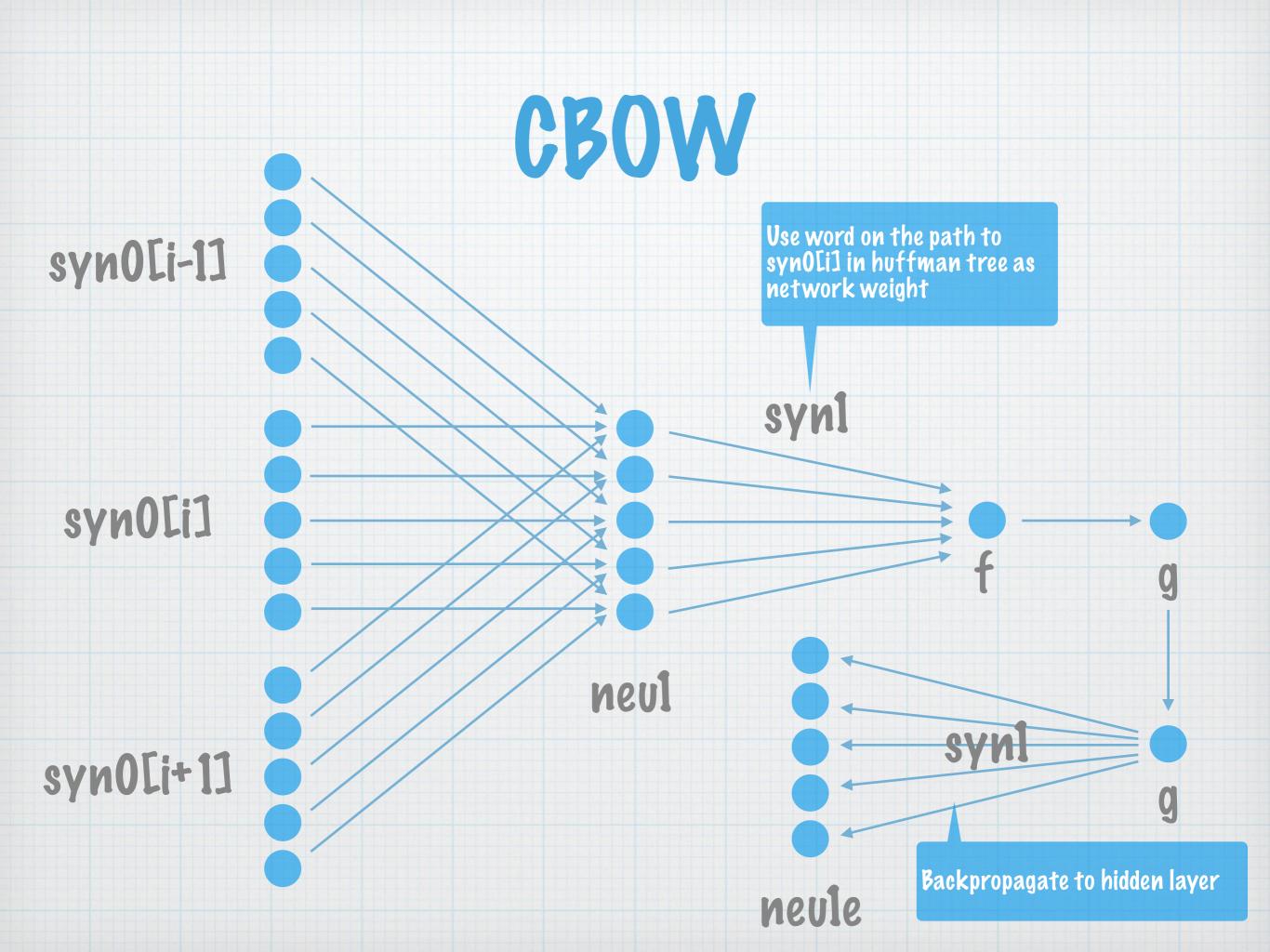
syn0[i]

syn0[i+1]



CBOW Use word on the path to syn0[i] in huffman tree as network weight syn0[i-1] syn0[i] neul syn0[i+1]



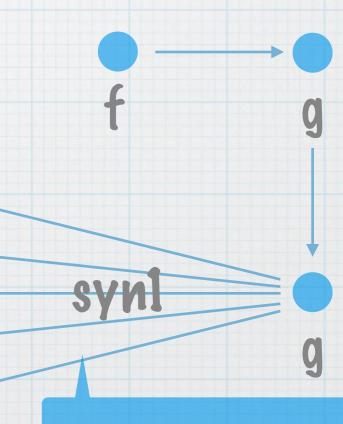


syn0[i-1]

Adding error back to each words in window

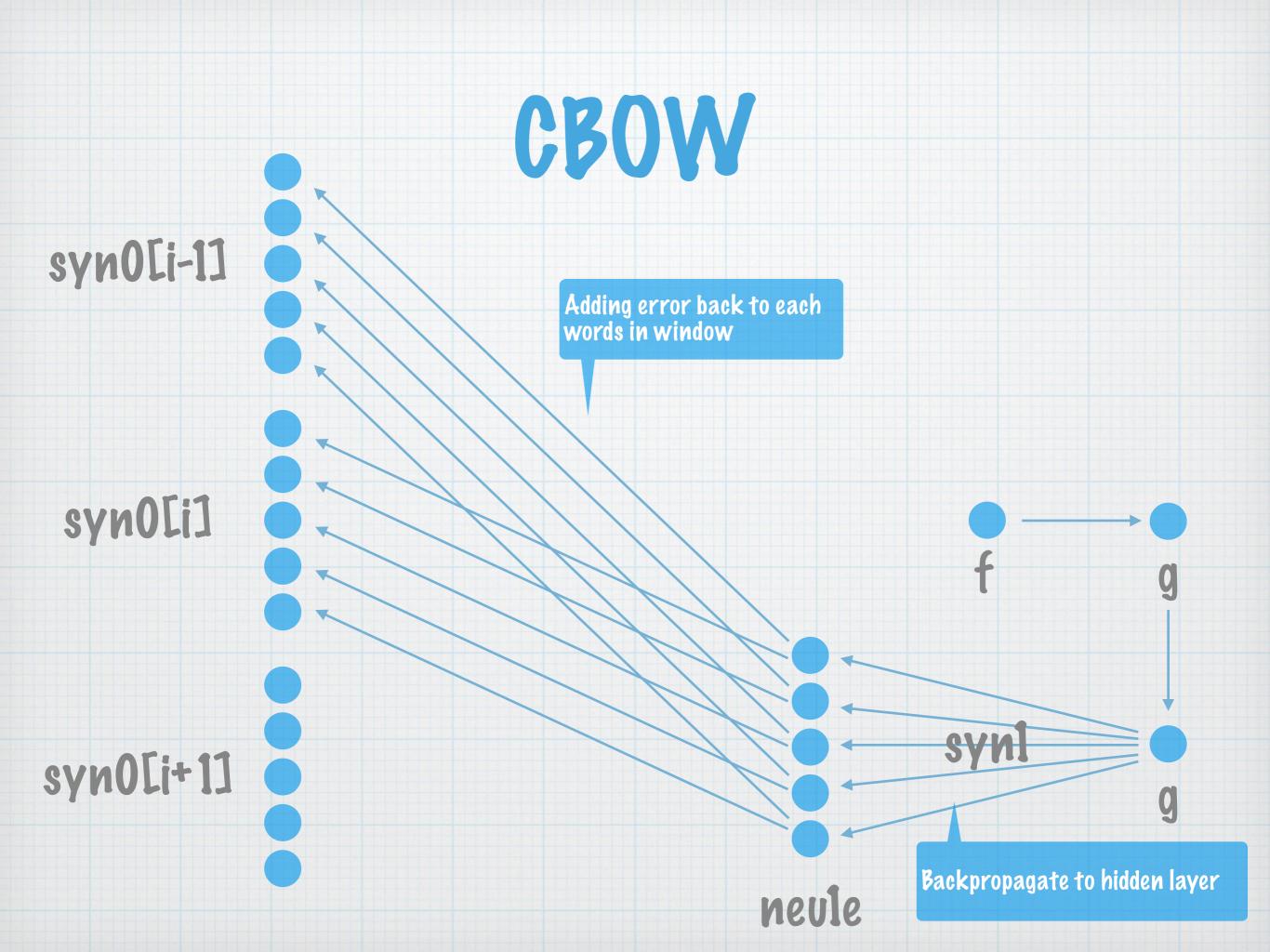
syn0[i]

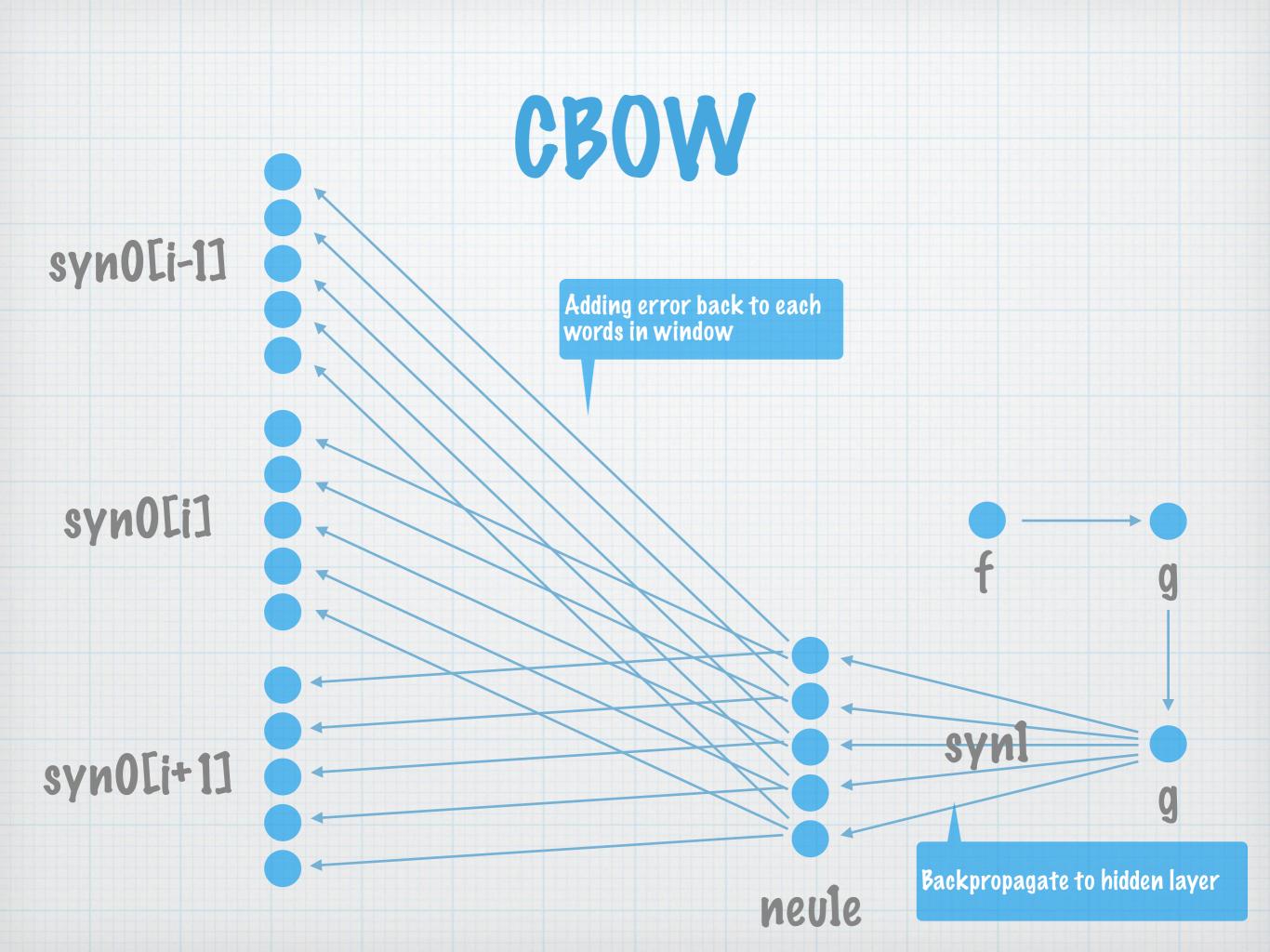
syn0[i+1]



neule

Backpropagate to hidden layer





syn0[i-1]

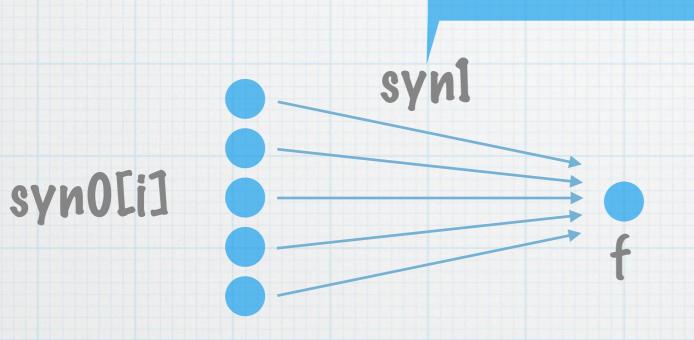
syn0[i]

They are the conclusive continuous representations that are available through training

syn0[i+1]

- * Reverse format of CBOW
- * Predict representations of word that is put around the target words

Use word on the path to synO[i] in huffman tree as network weight

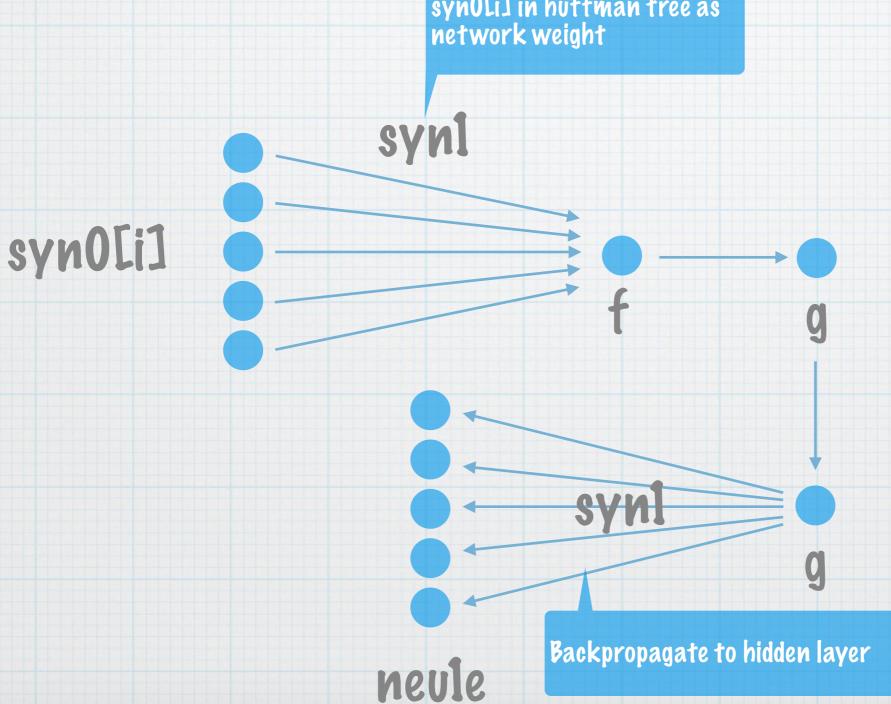


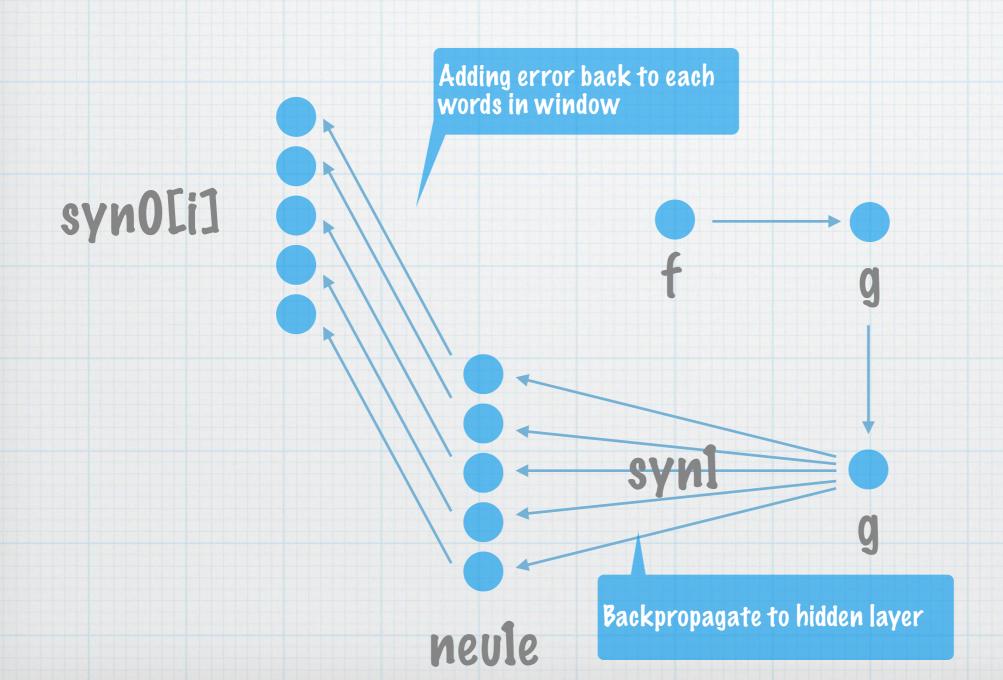
Use word on the path to synO[i] in huffman tree as network weight

syn0[i]

g is the gradient that is calculated with (1 - syn1.code -f) * alpha This is error of output

Use word on the path to synO[i] in huffman tree as network weight





Reference

* https://code.google.com/p/word2vec/