Deep context vectors (word2vec)

Approaches

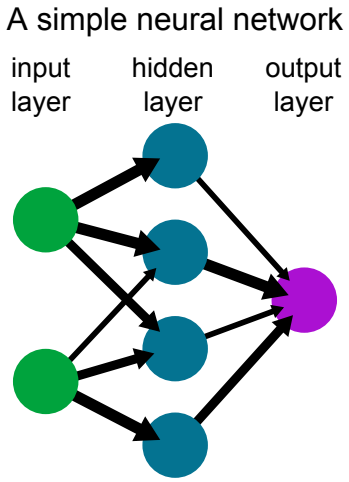
- continuous bag of words (CBOW)
    - predict word from context
- skip-gram
    - predict context from word

Aside: one-hot encoding

1. Create a vector of zeros with length $|V|$
2. When representing word i, set $w_i = 1$

$$[0, 0, ..., 0, 1, 0, ..., 0, 0]$$

Aside: counting layers



This network is sometimes described as having 1, 2, or 3 layers:

- 1 "hidden layer"
- 2 layers of weights
- 3 layers of neurons

we aren't going to decide what these weight are, what we get to do is set up the structure such that it can capture the information we want then, set up the inputs and outputs, and train it trough a bunch of data and assuming that we can achieve the goal that we set out to

Skip-gram

input: one-hot-encoded word

1 hidden layer (~300 nodes)

output: "context vector"

CBOW

input: "context vector"

1 hidden layer (~300 nodes)

output: one-hot-encoded word

Pre-trained embeddings

- gensim: https://radimrehurek.com/gensim/models/word2vec.html
- pretrained models: http://vectors.nlpl.eu/repository/