Embeddings

Learn the basics of word embeddings and why they're used.

Chapter Goals:

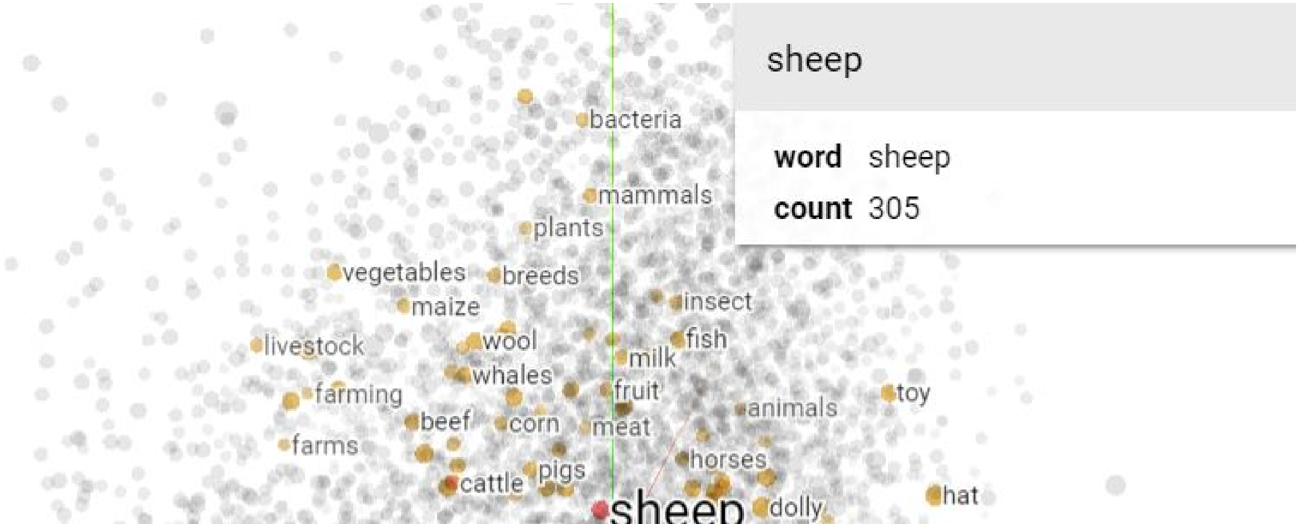
- Learn about word embeddings and why they're used
- Create a function that retrieves a target word and its context

A. Word representation

So far, the way we've represented vocabulary words is with unique integer IDs. However, these integer IDs don't give a sense of how different words may be related. For example, if a vocabulary gave the words "milk" and "cereal" the IDs 14 and 103, respectively, we would have no idea that these two words are often used within the same context.

The solution to this problem is to convert each word into an *embedding vector*. An embedding vector is a higher-dimensional vector representation of a vocabulary word. Since vectors have a sense of distance (as they are just points in a higher-dimensional space), embedding vectors give us a word representation that captures relationships between words.

Below is an example of word embedding vectors from the TensorFlow Embedding Projector. Note that the words most similar to "sheep" have shorter pointwise distances.



sheep

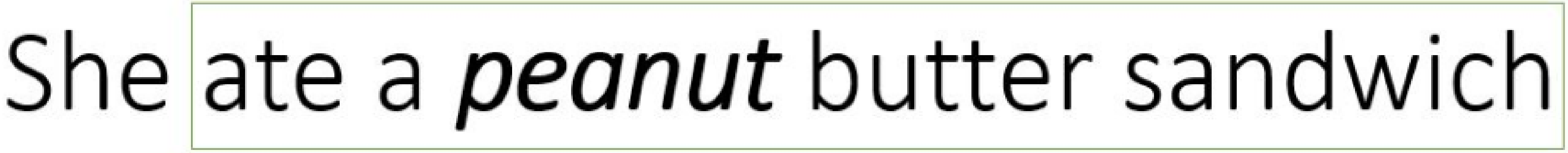
word sheep
count 305

When creating embedding vectors for your vocabulary, something to consider is how large the vectors are (i.e. the number of dimensions). Larger vectors are able to capture more relational tendencies between words, and are therefore better if you have a large vocabulary size. However, they also use up more computational resources and are more likely to overfit on smaller vocabularies.

A general rule of thumb is to set the number of dimensions in your embedding vectors equal to the 4th root of the vocabulary size. For example, if your vocabulary consists of 10000 words, the rule says your embedding vectors should have 10 dimensions. However, similar to the batch size or number of hidden layers in a neural network, you should try different embedding sizes to figure out which is best for the task at hand.

B. Target-context

The basis for embedding vectors comes from the concept of a *target* word and its *context window*. For each word of each tokenized sequence in the text corpus, we treat it as a *target*, and the words around it as the *context window*. This is how we define embedding vector relations; words that often appear near each other tend to be related.



A target word ("peanut") and context window (outlined in green).

We refer to the size of the context window as the *window size*. Since we want the context to be symmetric on both sides of the target word, the window size should be an odd number. In the above example, the window size is 5 (two words to the left and right of the target word).

We'll be learning how to convert target words and context windows into data that can be used to train an embedding vector model in the next chapter.