Word embeddings

- Convert text into numerical vectors (for example, necessary for NNets)
- First idea: one-hot encoding

- Example: let's consider the vocabulary
- Vocabulary: 'table', 'chair', 'small', 'plate', 'window',

- Here is how some of those words can be represented
- 'table' = [1, 0, 0, 0, 0]
- 'chair' = [0, 1, 0, 0, 0]

- It's a great starting point (equal distance between words). But it's sparse, as well as requiring very long vectors for each word (as long as the vocabulary size)

- These are ways to represent words in a multi-dimensional space.

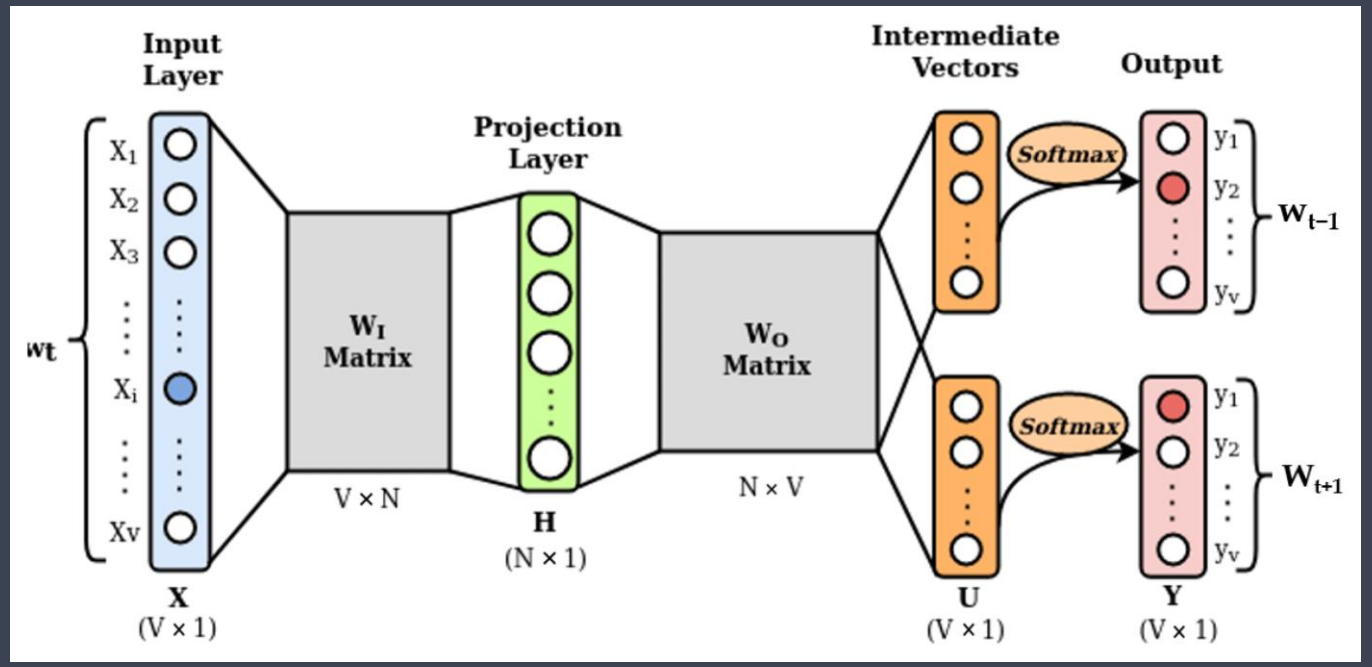
Word embeddings

- Like one-hot encodings, word embeddings are ways of representing words as numerical vectors, in a multi-dimensional space. Distance between words in that space is meaningful. It can capture the semantic distance between two concepts. For example, ‘dog’ close to ‘cat’, but far from ‘airplane’)
- Two main types: frequency-based and prediction-based

- TF-IDF is an example of frequency-based embedding (see previous lecture)

- Prediction-based embeddings are derived by training predictive models. For example, models that attempt to predict a target word based on the surrounding context (e.g., ‘sit’ and ‘eat’ to predict ‘table’ in the sentence ‘Let’s sit on the *table* and eat’).

Word embeddings



- One type of Word2Vec embedding is derived with the Skip-gram architecture.
- Each target word wt, represented as a one-hot encoding X, is used to predicted its surrounding context words (e.g., the previous and following content words w_{t-1} and w_{t+1}). All words are encoded in one-hot vectors.
- The mapping from target to context words is performed via a hidden/projection layer H (e.g., N=300), where the dimensionality of the word is reduced from the one-hot vector to a smaller vector (e.g., N=300). H captures the semantic relationships between words. H is a word-embedding.

Image from https://www.mdpi.com/2078-2489/12/1/38

Word embeddings

- That's fantastic. We can now represent words in reasonable sized multi-dimensional spaces, where distances between words represent semantic distances
- Caveat: distances are based on the assumption that similar words co-occur more often than unrelated words
- Issue: what happens for words that carry multiple distinct meanings? For example, bank (bank of a river vs. banking institution). We only get one embedding for each word, so that embedding will be a mix of the two.

- Solutions: extending Word2Vec by considering context, leading to separable embeddings for distinct concepts (even when the word is the same) – e.g., see BERT