Learnings:

Think about a crowded mall

This simple line was used to demonstrate how language is not just to transfer infomation from one person to another, but it is also used to refer to data which is common knowledge and known by both parties. So if both parties have a lot of general knowledge, a large amount of data can be talked on just by just pointing to it.

WordNet:

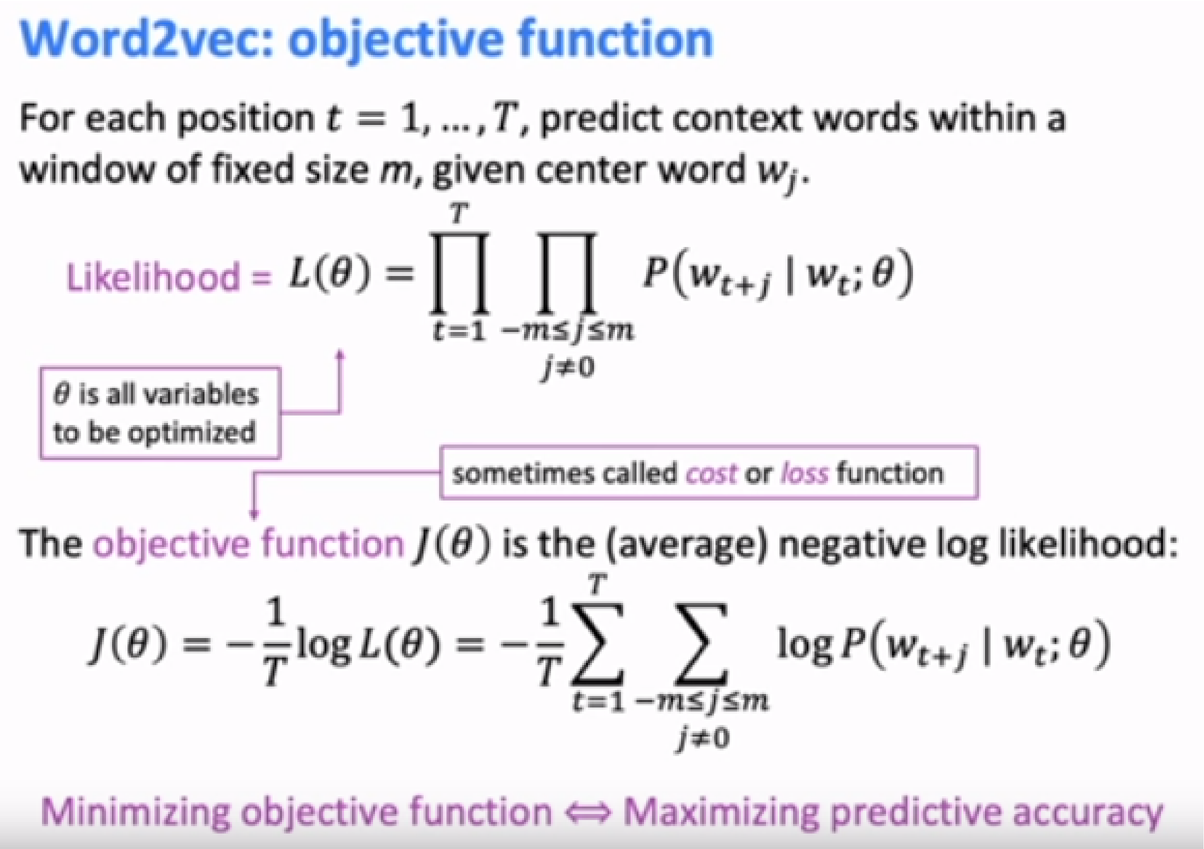
It is a database for english language which not only has synonyms for words but also has relation between words. It is generally known to not work effectively as it has man drawbacks like requiring a lot of manpower to update and adapt and it is also not possible to keep up to date with new words.

Representing words:

Words can be represented as a vector with 1 on a position corresponding to its location, but the problem with this is that such a vector will be very long and such a representation will not help in understanding its meaning. So instead of representing words absolutely, words are referred to by a vector where each value is in a range and words with similar values in similar positions have similar meanings.

Also note: we don't use only one vector to represent every word. Every word is represented by two vectors, V vector when it is the center word , U vector when it is the outside word.

Mathematics in word to vectors:



Word2vec: objective function

For each position $t = 1, \ldots, T$, predict context words within a window of fixed size *m*, given center word w_j.

$$\text{Likelihood} = L(\theta) = \prod_{t=1}^{T} \prod_{\substack{-m \le j \le m \\ j \ne 0}} P(w_{t+j} \mid w_t; \theta)$$

θ is all variables to be optimized

sometimes called *cost* or *loss* function

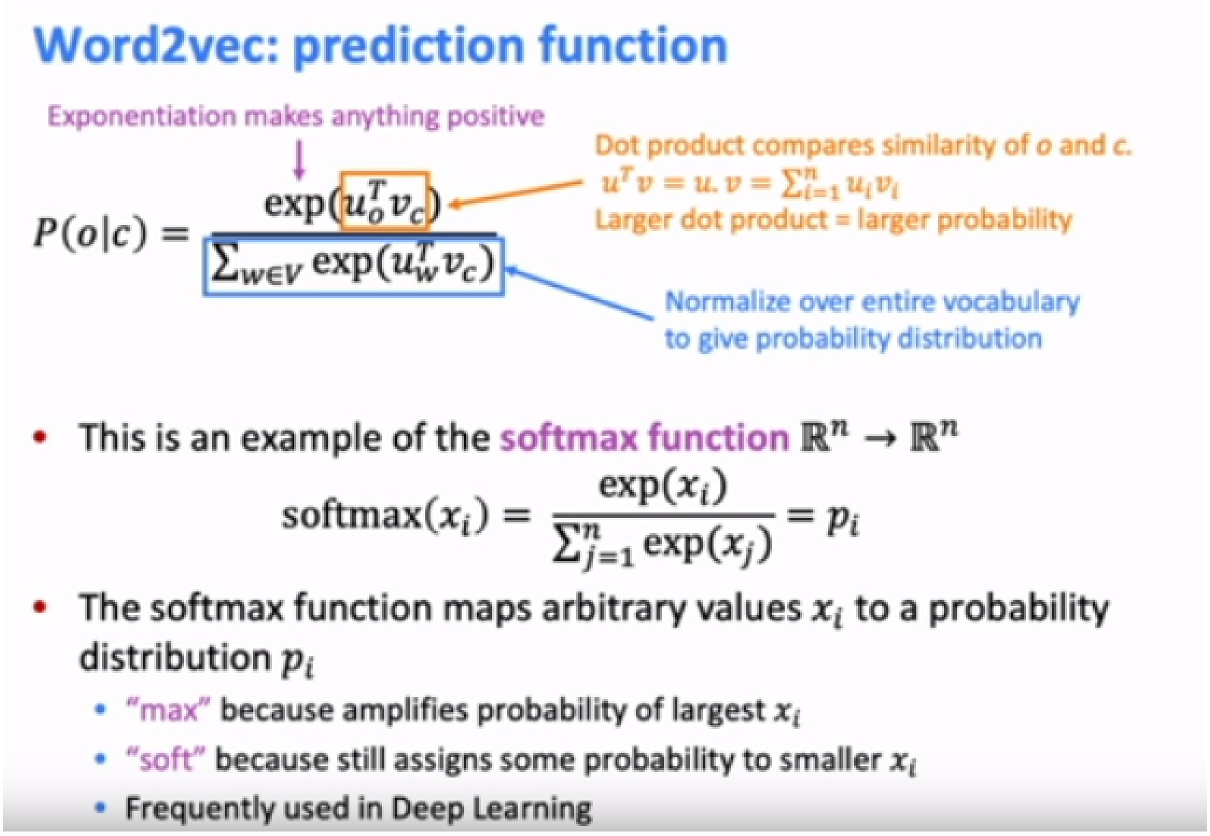
The objective function $J(\theta)$ is the (average) negative log likelihood:

$$J(\theta) = -\frac{1}{T} \log L(\theta) = -\frac{1}{T} \sum_{t=1}^{T} \sum_{\substack{-m \le j \le m \\ j \ne 0}} \log P(w_{t+j} \mid w_t; \theta)$$

Minimizing objective function ⟺ Maximizing predictive accuracy

IMPORTANT: Remember the j != 0 condition

; Symbol in probability means whatever b4 it depends on variables after it
Here theta is all the parameters in our model and only parameters here are the words



Word2vec: prediction function

Exponentiation makes anything positive

$$P(o|c) = \frac{\exp(\boxed{u_o^T v_c})}{\boxed{\sum_{w \in V} \exp(u_w^T v_c)}}$$

Dot product compares similarity of *o* and *c*.
$u^T v = u.v = \sum_{i=1}^{n} u_i v_i$
Larger dot product = larger probability

Normalize over entire vocabulary
to give probability distribution

- This is an example of the softmax function $\mathbb{R}^n \to \mathbb{R}^n$

$$\text{softmax}(x_i) = \frac{\exp(x_i)}{\sum_{j=1}^{n} \exp(x_j)} = p_i$$

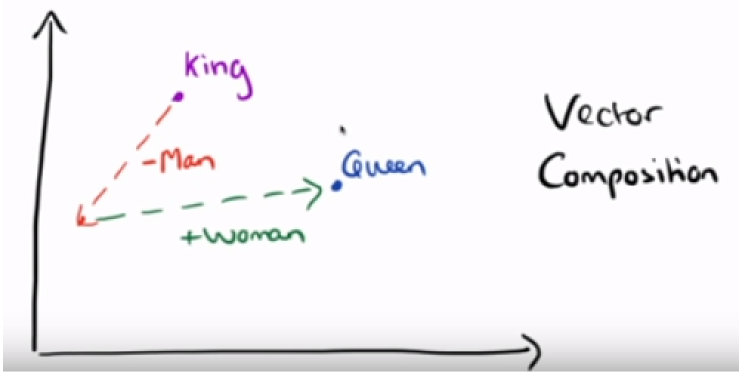
- The softmax function maps arbitrary values x_i to a probability distribution p_i
  - "max" because amplifies probability of largest x_i
  - "soft" because still assigns some probability to smaller x_i
  - Frequently used in Deep Learning

NOTE: O: outside word / context word
C: center word

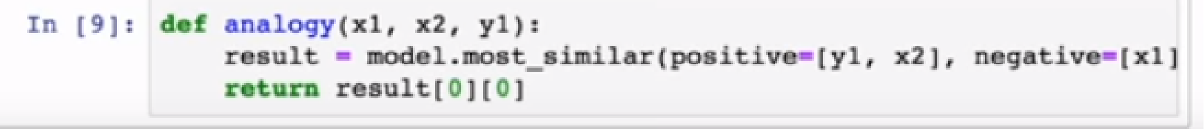
Vector composition:

It is one of the most highly celebrated abilities of word vetors

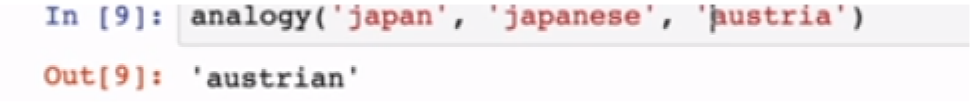
Example:



Example code:



```
In [9]: def analogy(x1, x2, y1):
            result = model.most_similar(positive=[y1, x2], negative=[x1]
            return result[0][0]
```



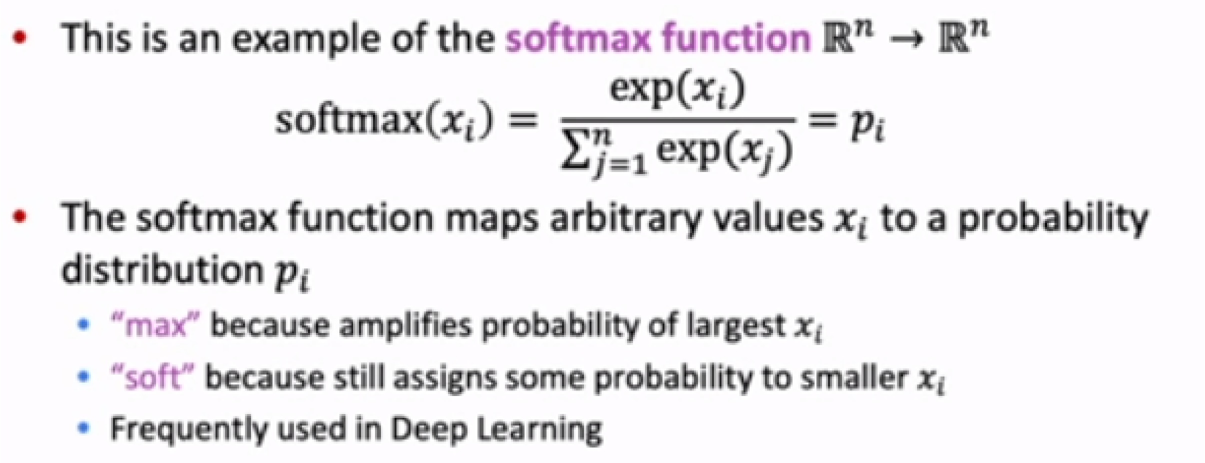
```
In [9]: analogy('japan', 'japanese', 'austria')

Out[9]: 'austrian'
```

Misc:

1. NLTK: natural language toolkit, used for doing simple tasks quicker, unlike full fledged NLP.

2. Always make sure if slides work correctly and screen does not jump to desktop before giving presentation.

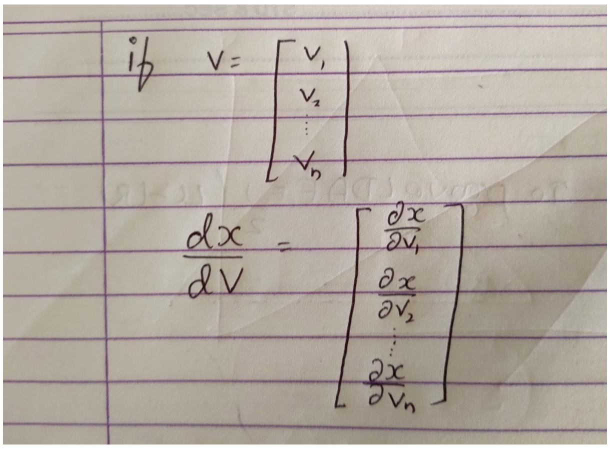
3. Corpus: large pile of text. (plural **corpora**)

4. 

- This is an example of the softmax function $\mathbb{R}^n \rightarrow \mathbb{R}^n$

$$\text{softmax}(x_i) = \frac{\exp(x_i)}{\sum_{j=1}^{n} \exp(x_j)} = p_i$$

- The softmax function maps arbitrary values x_i to a probability distribution p_i
  - "max" because amplifies probability of largest x_i
  - "soft" because still assigns some probability to smaller x_i
  - Frequently used in Deep Learning

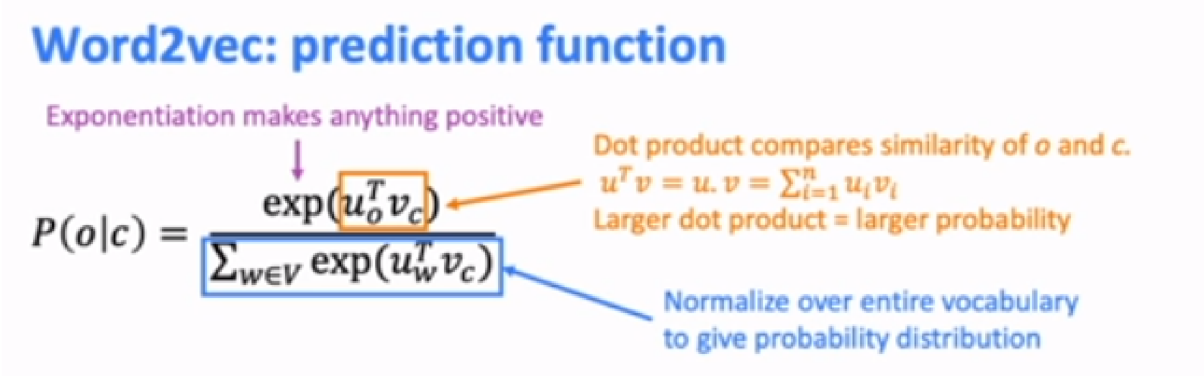
5. 

if $V = \begin{bmatrix} v_1 \\ v_2 \\ \vdots \\ v_n \end{bmatrix}$

$$\frac{dx}{dV} = \begin{bmatrix} \frac{\partial x}{\partial v_1} \\ \frac{\partial x}{\partial v_2} \\ \vdots \\ \frac{\partial x}{\partial v_n} \end{bmatrix}$$

Doubts:

1.



Word2vec: prediction function

Exponentiation makes anything positive

$$P(o|c) = \frac{\exp(\boxed{u_o^T v_c})}{\boxed{\sum_{w \in V} \exp(u_w^T v_c)}}$$

Dot product compares similarity of *o* and *c*.

$u^T v = u.v = \sum_{i=1}^{n} u_i v_i$

Larger dot product = larger probability

Normalize over entire vocabulary to give probability distribution

How does this represent probability uniformly?
For example let the center word X1 have its center word vector v close to the origin of the nd space, so the outside vector of an outside word near X1 is multiplied with it and the dot product is the numerator. This is smaller than the center word X2 which has its center vector farther from origin, so just because a word is far from origin (Note: a word's absolute positions are pretty random only the relative positions matter) its probability of having relation better ?

Solution: no, it is balanced by the denominator where center word vector Vc remains the same and if values of Vc is smaller then the denominator is also smaller.

2.
How did the creator of the neural network visualization manage to project points in 100-dimensions into 2-dimensions and it makes sense ?