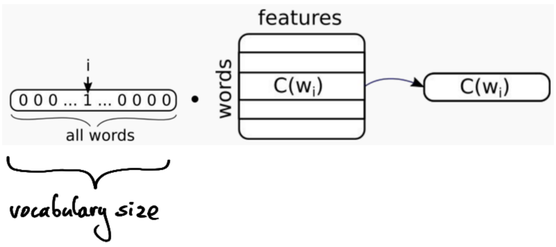
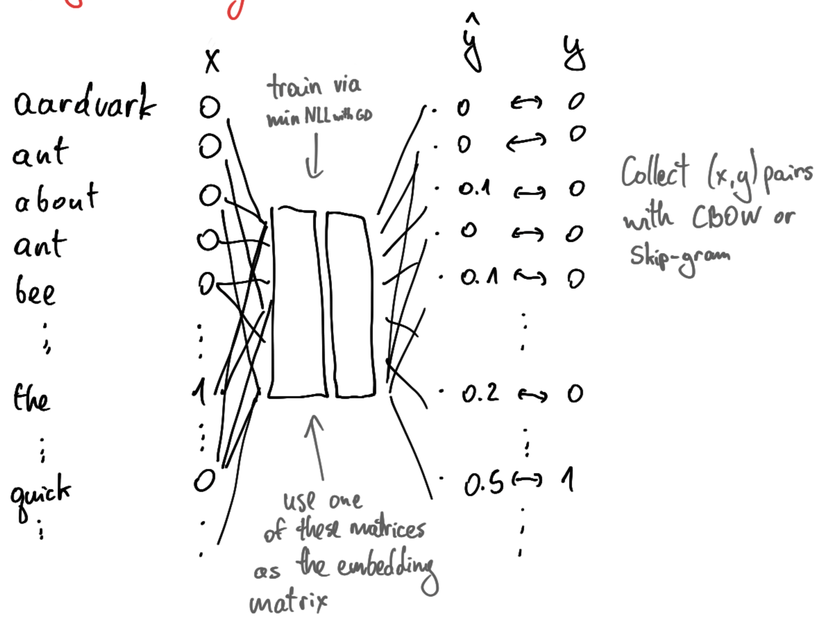
Embeddings



To get an embedding matrix:



Collect (x,y) pairs with CBOW or Skip-gram

To make a tokenizer via the BPE (byte pair encoding) algorithm:

The quick brown fox jumps over the lazy dog

↑ start off by splitting the text into letters (bytes) to be the tokens

- repeat:
  - find most statistically probable pair
  - merge it to form a new token (merge it in the dataset too)

until probability of most common pair is below a threshold (or have enough tokens)

Attention Mechanism

$$\text{Attention}(Q,K,V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V$$

↳ QK^T acts as a similarity search

↳ $\cdot V$ acts as a 'fuzzy' database indexer

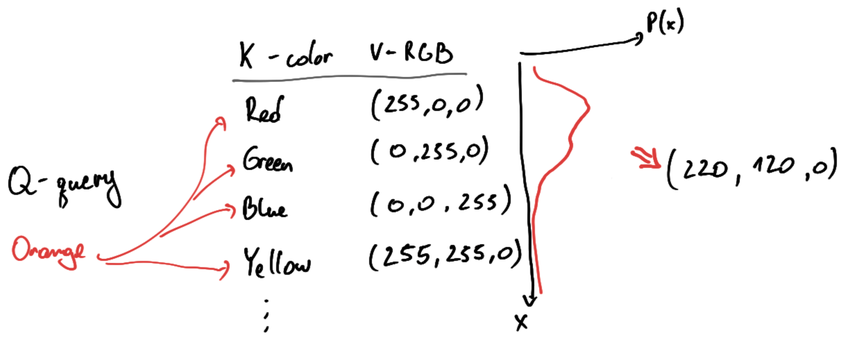
$Q = XW_q$

$K = XW_k$

$V = XW_v$

$X \in \mathbb{R}^{S \times L}$ S - sequence length, L - vocabulary length

$W_q \in \mathbb{R}^{L \times d_k}$ d_k - hidden layer dim



Key	Value
[10, 0, 0]	[1, 0, 1]
[0, 10, 0]	[10, 0, 2]
[0, 0, 10]	[100, 5, 0]
[0, 0, 10]	[1000, 6, 0]

Examples:

$Q = [0, 10, 0]$

Match with $[0, 10, 0] \Rightarrow V = [10, 0, 2]$

$Q = [0, 0, 10]$

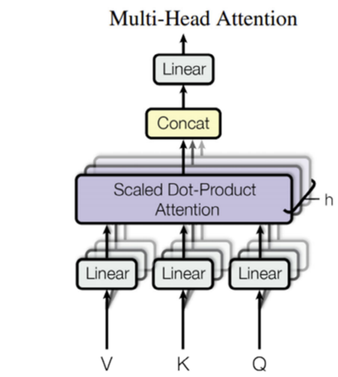
Matches two rows $\Rightarrow V = 0.5 \cdot [100, 5, 0] + 0.5 \cdot [1000, 6, 0]$

$= [550, 5.5, 0]$

$Q = [10, 10, 0]$

Matches $[10,0,0]$ and $[0,10,0] \Rightarrow V = 0.5 \cdot [1,0,1] + 0.5 \cdot [10,0,2]$

$= [5.5, 0, 1.5]$



Positional Encoding