1 | Overview

Attention is a process for interpreting context of inputs, something computers and networks aren't usually great at (see interpreting sentences like "The animal was tired so it didn't cross the road" and understanding the "it").

Attention (in the context of NLP) allows better encoding of words by looking at the rest of the words in the sentence so that machines can understand context.

2 | Calculation

To begin, we calculate query, key, and value vectors by multiplying the word embeddings (vector representations of words) by weight matrices *Note: Where the hell do these come from? Ask Jack?*. Then we calculate an attention score by multiplying query by key, dividing by the squareroot of the key vector dimensionality (a trick to help with gradients), take softmax, and finally multiply by value vector.

Alternatively this can be represented by $Z = \text{softmax}(\frac{Q \cdot K^T}{\sqrt{d_k}})V$, where Z is the attention output. Calculation of the key/value/query vectors and their involvement is intuitive to some degree because the weights should signify importance of words, etc.

3 | Multi-Headed Attention

Different version of attention with even more weird representations like query/key/value.

4 | Positional Encodings

Positional encodings, vectors representing where the word is in the sentence are added to the original word embedding to allow a notion of time/position in the encoding process.