Word Embedding (P-3)

- A LLM can only process a fixed number of tokens at a time this is called Context size. GPT = 2048 max

Ex: [The] [smartest] [person] [is] Vectors/tokens: Context size = 4

- this means as you go on and on with a conversation with a LLM it slowly starts to lose the context and thread of the conversation if you continue too long as it has only so much context from before (context)

* - Unembedding

- at the end we have a probability distribution of all possible next tokens. We use the last vector for this

- this involves 2 steps

Ex The smartest person is

Final layer ⇒ [[ ] [ ] [ ] [ ]] → { Einstein 78%, the 30%, would 20%, ⋮ }

1) We have a unembedding matrix that maps the very last vector in the context to a list of 50k values one for each token in vocab

2) [Unembedding Matrix W_u (50k rows (cols))] · [ ] = [ ] (start at random → learns in training)

[[ ] [ ] [ ] [ ]]

2) Then we use a softmax function to normalize it so all probabilities add upto 1 (probability distribution)

2) → Softmax ⇒ { einstein 0.78, the 0.30, would 0.20, ⋮ }

? What about the other vectors in the final layer why only the last one? They all have context and meaning so why not use them?

- this is because in the training process the { smartest 5.1, one 2.1, etc ⋮ } the smartest { person 2.1, one 5.1 } . . .

it's more efficient if you use each vector in the final layer to simultaneously make a prediction for what would come immediately after it

[[the] [ ] [smartest] [ ] . . . . [is] [ ]] . . .