

"Today is a _____"

1) Tokenize

"Today is a _____"
↓

[Today, is, a]
↓
1x t # input tokens

2) Embed

Vocabulary = $\begin{bmatrix} a, \text{aa}, \dots, zzz \\ 1 & 2 & \dots & 9 \\ 3 & 2 & \dots & 9 \end{bmatrix}$ $\begin{matrix} \uparrow \# \text{words} \\ \downarrow v \times d \end{matrix}$
Tokens = [Today, is, a] $\begin{matrix} \uparrow \text{dimension of embedding} \\ \downarrow 1 \times t \end{matrix}$

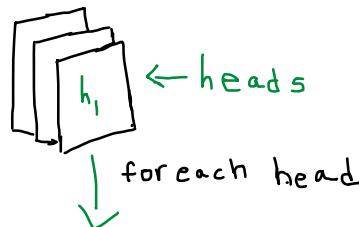
token embedding = $\begin{bmatrix} 4 & 1 & 3 \\ 3 & 5 & 6 \\ 7 & 8 & 0 \end{bmatrix}$ $\begin{matrix} \uparrow d \times t \\ + \end{matrix}$

position embedding = $\begin{bmatrix} 1 & 1 & 1 \\ 0 & 1 & 3 \dots \\ 3 & 5 & 4 \end{bmatrix}$ $\begin{matrix} \uparrow t \times d \\ \downarrow \# \text{tokens model supports} \end{matrix}$

Input X = $\begin{bmatrix} 5 & 2 & 5 \\ 3 & 6 & 6 \\ 10 & 17 & 7 \end{bmatrix}$ $t \times d$

4) Attention

X $t \times d$



$W_Q^{(i)}, W_K^{(i)}, W_V^{(i)}$ $\begin{matrix} \uparrow d \times d_h \\ \downarrow \# \text{dims/head} \end{matrix}$ → Learned Proj. Matrices
calculate Q, K, V

$Q^{(i)} = XW_Q^{(i)}$ → Query "What am I looking for?"
 $K^{(i)} = XW_K^{(i)}$ → Key "What do I offer?"
 $V^{(i)} = XW_V^{(i)}$ → Value "If selected, what will I pass?"
 $t \times d_h$

$$\text{Attention}^{(i)}(Q, K, V) = \text{softmax}\left(\frac{Q^{(i)}(K^{(i)})^\top}{\sqrt{d_h}}\right)V^{(i)}$$

$t \times t$
scaling

"From head h's POV,
what tokens should each token pay
attention to"

Fine Tuning

In attention Layers

- $W_a' = W_a + \Delta W_a$
- $W_k' = W_k + \Delta W_k$

$$\cdot \quad W_v' = W_v + \Delta W_v$$

In FFT,

Full Matrix is updated

$$\Delta W \in d \times d$$

In PEFT,

$$\Delta W = B \cdot A \quad r \ll d$$

$d \times r \quad r \times d$

A = "which directions in input to pay attention to"

B = "how to inject that adjustment into the output space"

Original weights are frozen

Why?
W_v'

ΔW is usually very low-rank or close to low-rank.

That is, the changes needed to adapt a pretrained model to a new task often lie in a small subspace.

Most of the full $d \times d$ matrix doesn't need large changes; only a few directions matter.