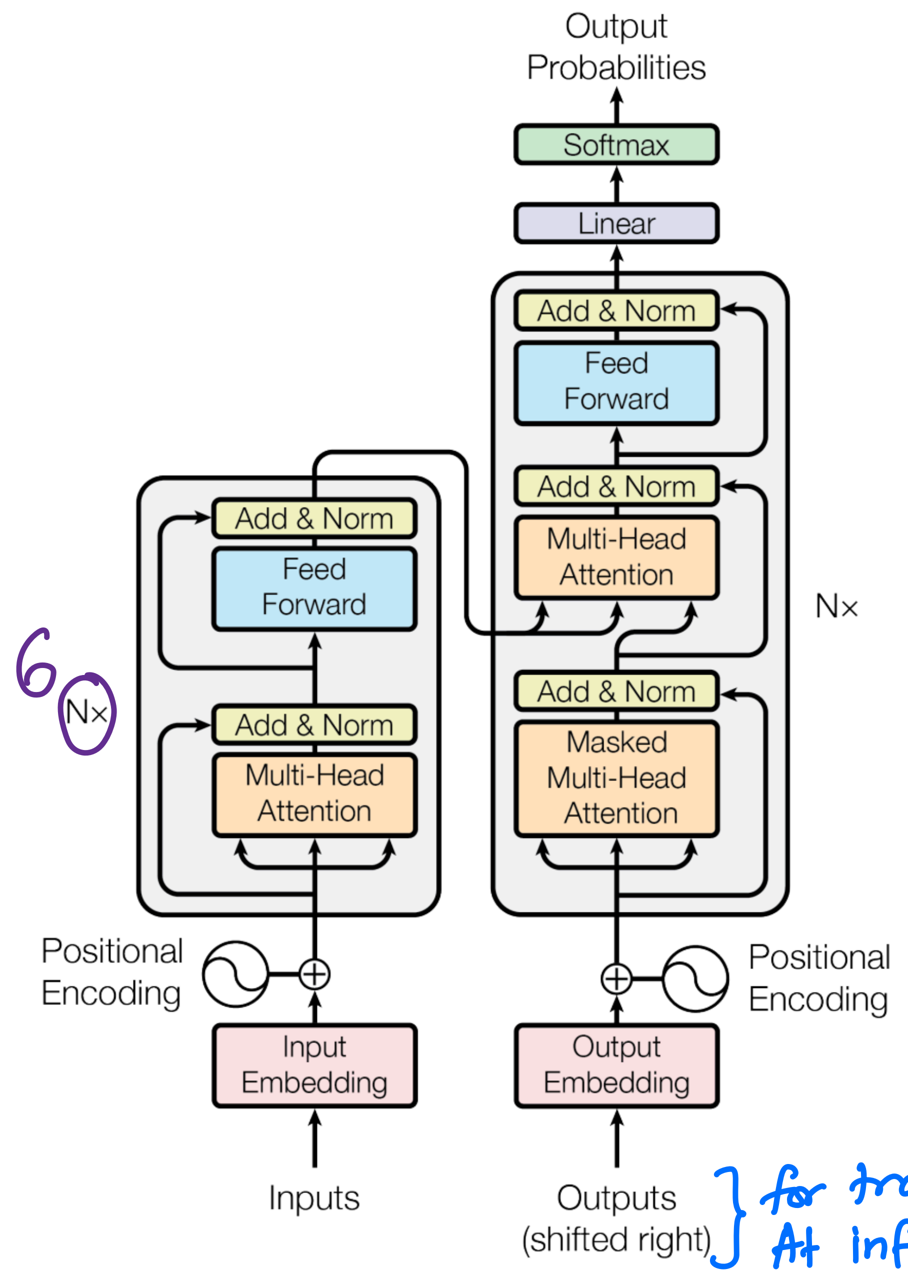
Lecture 19

- Attention recap.
- Transformer architecture
- Embedding layer
- Positional encoding.

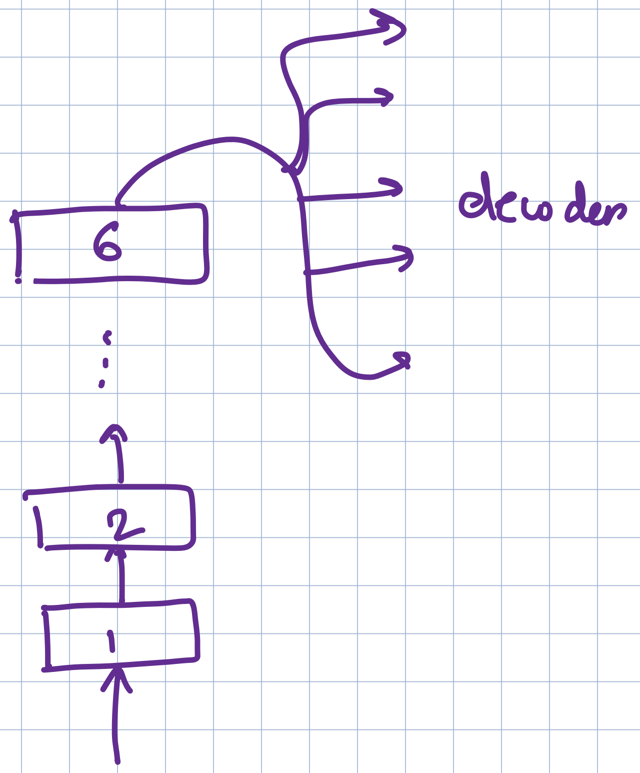
Attention is all you need.



} for training.
At inference, just put in <start>.

Figure 1: The Transformer - model architecture.

- Normalization layers are now typically before the attention block.



GPT-3 architecture (Decoder only).

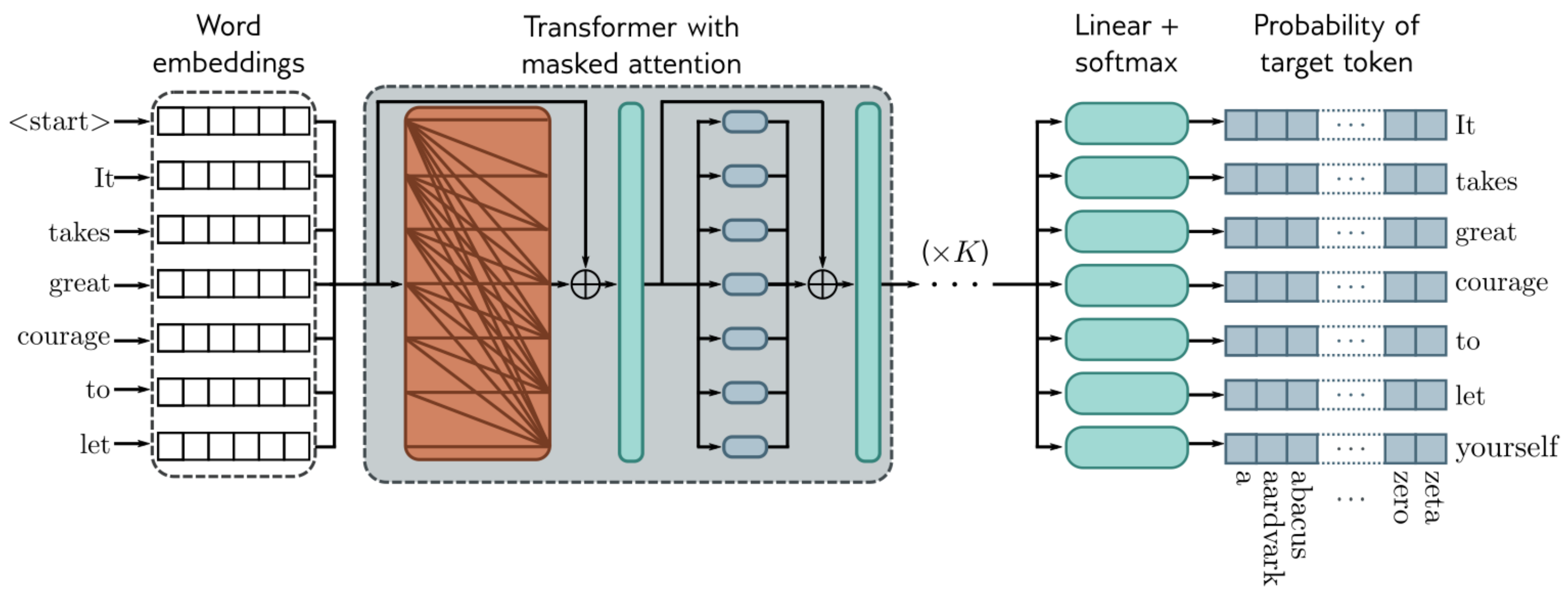


Figure 12.12 Training GPT3-type decoder network. The tokens are mapped to word embeddings with a special <start> token at the beginning of the sequence. The embeddings are passed through a series of transformer layers that use masked self-attention. Here, each position in the sentence can only attend to its own embedding and those of tokens earlier in the sequence (orange connections). The goal at each position is to maximize the probability of the following ground truth token in the sequence. In other words, at position one, we want to maximize the probability of the token It; at position two, we want to maximize the probability of the token takes; and so on. Masked self-attention ensures the system cannot cheat by looking at subsequent inputs. The autoregressive task has the advantage of making efficient use of the data since every word contributes a term to the loss function. However, it only exploits the left context of each word.

Embedding Layer.

- Use Learned embeddings.
- How?

Assume fixed vocabulary size, e.g. 32,000.
Embedding dimension = 1000 say.

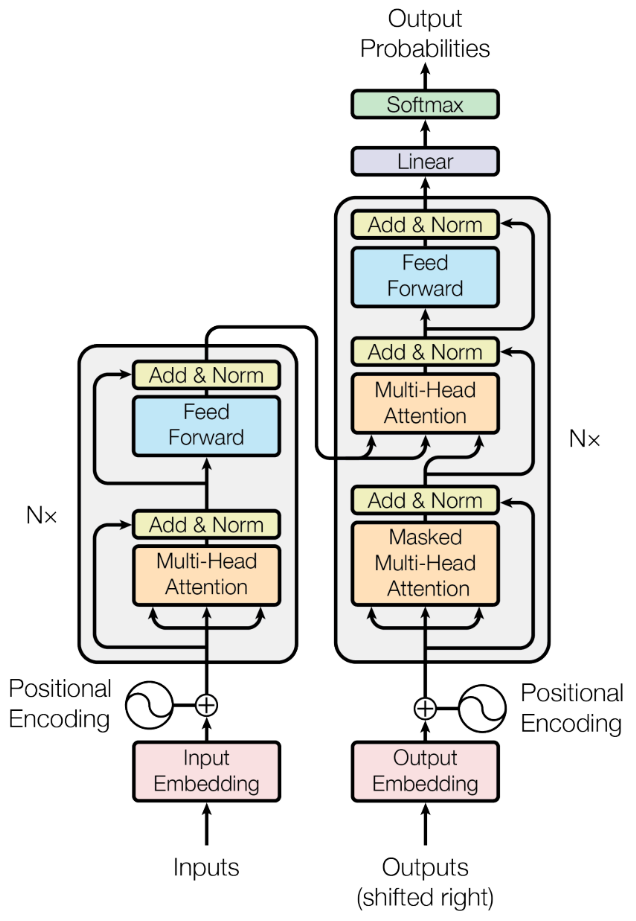


Figure 1: The Transformer - model architecture.

Input is a sequence of tokens from your vocabulary.

→ One hot encoded vectors in $\mathbb{R}^{32,000}$.

→ Need 1000 dimensional vectors.

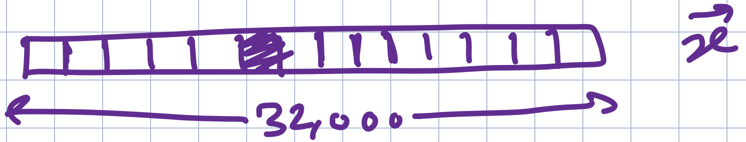
→ Use a linear layer and learn this.

- At output: Need to get scores that are compatible with a softmax.

Need to take 1000-dim vector to 32,000 scores → Softmax → Probability.

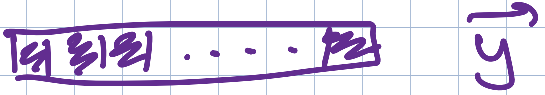
→ Again Linear layer

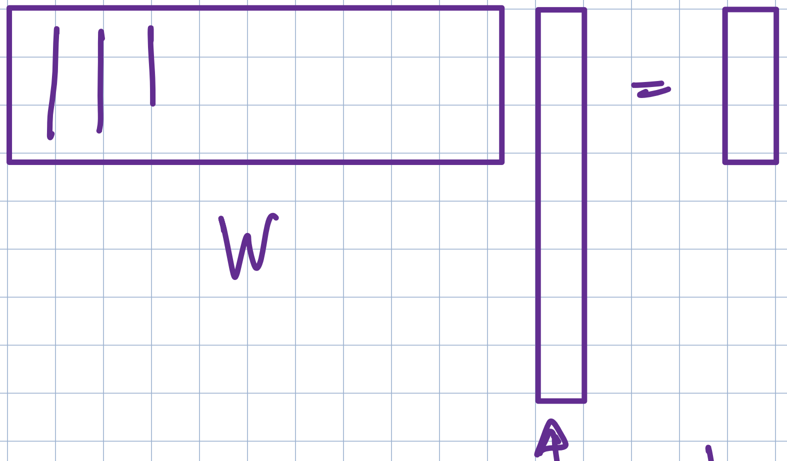
Input : RMS-norm $= \frac{1}{\sqrt{32{,}000}}$



We want RMS-norm for input to attention heads to be 1.

$W\vec{x} = \vec{y}$



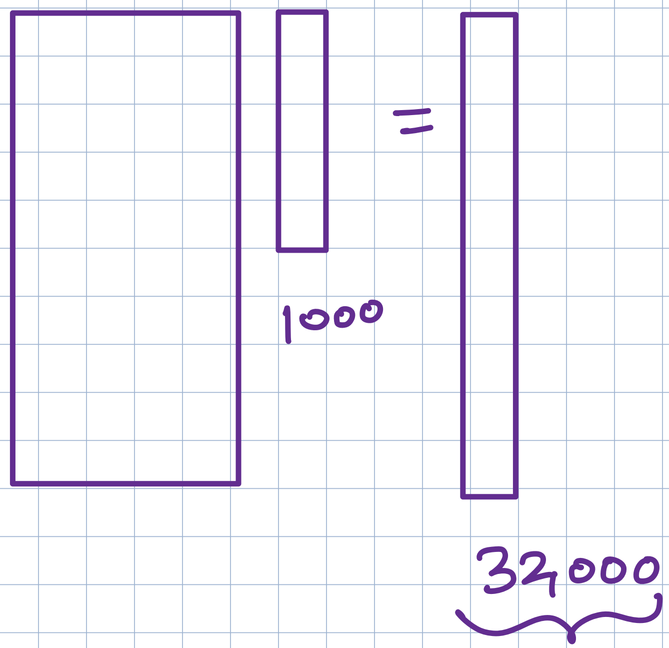


one hot

– so picks out one column of W.

So want each column of W to be RMS norm 1.

For "de"-embedding:



$$Q\vec{\alpha} = \vec{\beta}.$$

$$\|\vec{\alpha}\|_{RMS} = 1.$$

want to make sure that entries are not too big.

Every entry of $Q \approx \frac{1}{\sqrt{1000}}$

$$q_{i_1}^2 \cdot 1 + q_{i_2}^2 \cdot 1 + \cdots + q_{i_n}^2 \cdot 1 \approx 1$$

$$\Sigma q_i^2 = 1 \Rightarrow 1000\, q_i^2 = \overset{1}{\cancel{1000}} \Rightarrow q_i^2 = \frac{1}{1000} \Rightarrow q_i \approx \frac{1}{\sqrt{1000}}.$$

Positional Encoding.

① Attention is all you need.

Add positional encodings.

Sinusoidal pattern.

→ No longer used.

② NoPE.

→ No position encoding.

③ Learned Relative positional encoding.

Basic attention: Score $s_i = \frac{\langle \vec{q_t}, \vec{k_i} \rangle}{\sqrt{d}}.$

q_t at time t

k_i is key at pos i

$$P_{t,i} = \frac{e^{s_i}}{\sum_j e^{s_j}}$$

$$P_{t,i} = \frac{e^{s_i + \boxed{}}}{\sum_j e^{s_j + \boxed{}}}$$

$b_{\text{shift}} \rightarrow t-i$

$$P_{t,i} = \frac{e^{s_i + b_{t-i}}}{\sum_j e^{s_j + b_{t-j}}}$$

learn these!

This changes the attention matrix!

4 RoPE

Modify attention calculation to capture relative position.

$$\langle \vec{q}_t, \vec{k}_i \rangle$$

$d = 2.$

$$\begin{bmatrix} q_{t1} \\ q_{t2} \end{bmatrix}$$

$$R_t = \begin{bmatrix} \cos w_t & -\sin w_t \\ \sin w_t & \cos w_t \end{bmatrix}$$

$$\langle R_t \vec{q}_t , R_i \vec{k}_i \rangle$$

$$= \vec{q}_t^{\,T} \underbrace{R_t^T R_i} \vec{k}_i$$