Data 188 Introduction to Deep Learning
Spring 2026 Eric Kim

Discussion 09

This week we continue our study of transformers by focusing on the encoder-decoder setting.

1. Cross-Attention Shapes

Recall that in multi-head cross-attention, the queries come from the target sequence, while the keys and values come from the source sequence. For head $h \in \{1, \dots, H\}$,

$$Q_h = X_{\text{tgt}} W_h^{(q)}, \qquad K_h = X_{\text{src}} W_h^{(k)}, \qquad V_h = X_{\text{src}} W_h^{(v)},$$

$$\mathrm{H}_h = \mathrm{Softmax}\left(\frac{Q_h K_h^\top}{\sqrt{d_q}}\right) V_h,$$

and the full multi-head output is

$$Y = \mathrm{Concat}(\mathrm{H}_1, \dots, \mathrm{H}_H)\, W^{(o)}.$$

In the figure below, we will focus on a *single* cross-attention block (that is, one head). As a result, each target token computes attention weights over the source tokens, and these weights depend on the particular input sequences in the batch.

Assume $X_{\text{tgt}} \in \mathbb{R}^{B \times n_{\text{tgt}} \times d}$, $X_{\text{src}} \in \mathbb{R}^{B \times n_{\text{src}} \times d}$, and assume the query/key dimension is d_q (so $d_k = d_q$) and the value dimension is d_v.

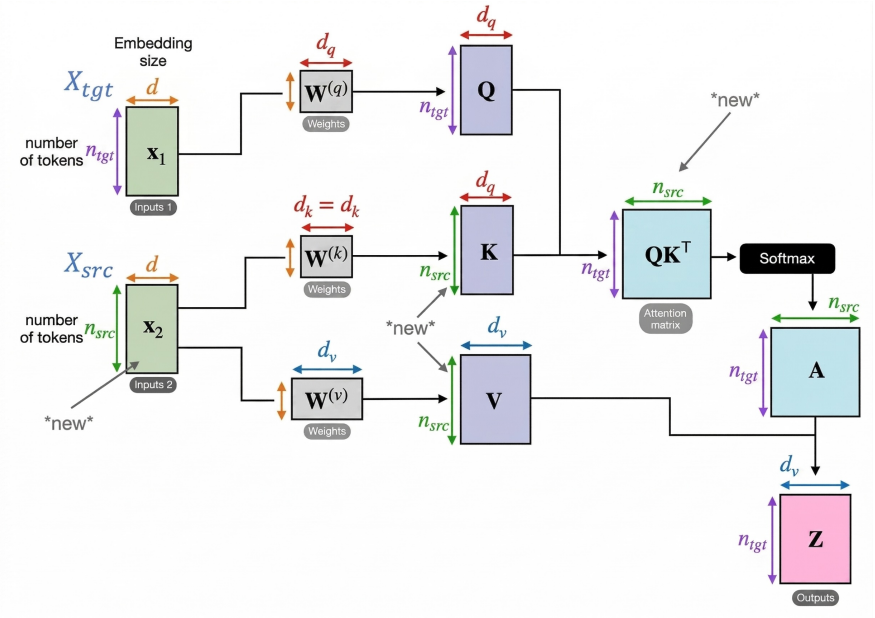

Figure 1: Cross-attention block.

You may annotate the figure as you work, but please write your final answers below.

(a) What are the shapes of the projection weight matrices $W^{(q)}$, $W^{(k)}$, and $W^{(v)}$?

(b) What are the shapes of the projected activations Q, K, and V?

(c) What are the shapes of the score matrix $QK^\top$, the attention matrix

$$A = \mathrm{Softmax}\left(\frac{QK^\top}{\sqrt{d_q}}\right),$$

and the final output

$$Z = AV?$$

Then, in one or two sentences, explain what the entries of A represent. Why can A be viewed as a set of *data-dependent* weights?

2. Decoder Block Shapes in an Encoder-Decoder Transformer

The figure below zooms in on one decoder block inside an encoder-decoder transformer for machine translation.

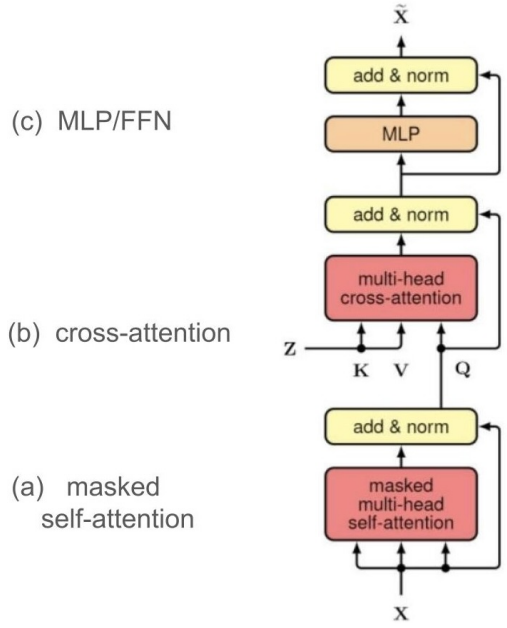

Figure 2: One decoder block in an encoder-decoder transformer.

You may annotate the figure as you work, but please write your final answers below.

Assume LayerNorm and residual connections do not change tensor shapes.

For both attention sublayers, assume query/key dimension d_q (so $d_k = d_q$), value dimension d_v, and an output projection back to model dimension d.

(a) Recall the single head attention mechanism (these are the same equations you've seen before, but we've just added the subscript "self" to distinguish from the cross-attention sublayer):

$$
\begin{aligned}
Q_{\text{self}} &= XW_{\text{self}}^{(q)} \\
K_{\text{self}} &= XW_{\text{self}}^{(k)} \\
V_{\text{self}} &= XW_{\text{self}}^{(v)} \\
A_{\text{self}} &= \text{Softmax}\left(\frac{Q_{\text{self}}K_{\text{self}}^{\top}}{\sqrt{d_k}}\right) \quad \text{(Attention matrix)} \\
H_{\text{self}} &= A_{\text{self}}V_{\text{self}} \quad \text{(Attention head)} \\
Y_{\text{self}} &= H_{\text{self}}W_{\text{self}}^{(o)} \quad \text{(Attention block output, single head)}
\end{aligned}
$$

In the masked self-attention sublayer, what are the shapes of

$$W_{\text{self}}^{(q)},\ W_{\text{self}}^{(k)},\ W_{\text{self}}^{(v)},\ W_{\text{self}}^{(o)},$$

the activations

$$Q_{\text{self}},\ K_{\text{self}},\ V_{\text{self}},$$

the masked attention matrix A_{self}, and the projected sublayer output Y_{self} right before the first add & norm?

Assume the decoder takes target-side hidden states $X \in \mathbb{R}^{B \times n_{\text{tgt}} \times d}$ as input to this block.

(b) Assume the encoder has already processed the source sequence and produced $E \in \mathbb{R}^{B \times n_{\text{src}} \times d}$.

The cross-attention sublayer is the same as the previous part, except that the queries (Q) come from the decoder stream (same shape as Y_{self}), while the keys (K) and values (V) come from the encoder output E.

What are the shapes of

$$W^{(q)}_{\text{cross}},\ W^{(k)}_{\text{cross}},\ W^{(v)}_{\text{cross}},\ W^{(o)}_{\text{cross}},$$

the activations

$$Q_{\text{cross}},\ K_{\text{cross}},\ V_{\text{cross}},$$

the attention matrix A_{cross}, and the projected sublayer output Y_{cross} right before the second add & norm?

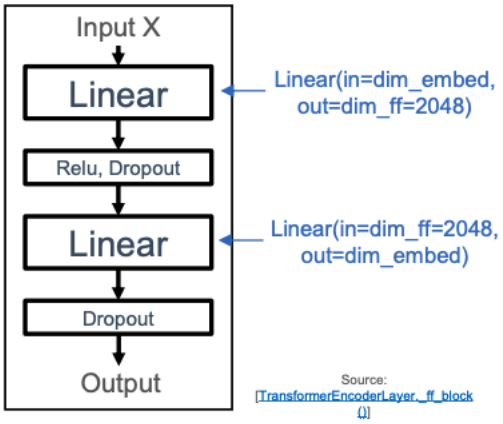

Figure 3: FNN Block (Lec 16 Slide 36).

(c) In the above **MLP/FFN sublayer**, the first linear layer expands from model width d to hidden width d_{ff}, a nonlinearity (ReLU) and dropout is applied, and the second linear layer projects back to width d.

Note: The input matrix X to the MLP block is the output of the previous add & norm (see Figure 3), which has the same shape as the output of the cross-attention block Y_{cross}. In other words, $X \in \mathbb{R}^{B \times n_{\text{tgt}} \times d}$.

$$Z = \text{Dropout}(\text{ReLU}(XW_1))$$
$$\tilde{X} = ZW_2$$

What are the shapes of the two MLP weight matrices, the hidden activation Z, and the final decoder-block output $\tilde{X}$?

3. Why Attention Is More Than a Linear Layer

Recall the equations for a single attention head:

$$Q = XW^{(q)}$$
$$K = XW^{(k)}$$
$$V = XW^{(v)}$$
$$H = \text{Softmax}\left(\frac{QK^\top}{\sqrt{d_q}}\right)V.$$

In this question, ignore batching and assume $X \in \mathbb{R}^{n \times d}$, where n is the sequence length.

(a) For this part only, ignore the softmax and normalization ($\sqrt{d_q}$) and set $W^{(q)} = W^{(k)} = W^{(v)} = I$. Show that the attention output simplifies to $H = (XX^\top)X$.

(b) Why does the expression $(XX^\top)X$ show that attention is *nonlinear* in X? Why does it also show that attention *mixes information across tokens*?

(c) Now restore the learned projections and the softmax. Rewrite the attention output in the form $H = A(X)XW^{(v)}$, where $A(X)$ is a function that takes in X as input and returns an $n \times n$ matrix that depends on the current input. **Hint:** $(AB)^\top = B^\top A^\top$ for any matrices A and B.
In what sense can attention be viewed as a *data-dependent* linear layer? How is this different from an ordinary linear layer XW?

4. Why the Decoder Looks the Way It Does

This final question focuses on several design choices that appear in encoder-only, decoder-only, and encoder-decoder transformers.

(a) Suppose we remove positional encodings entirely from a transformer. What kind of information would the model lose? Give one concrete example of two sentences that would become difficult to distinguish based on meaning.

(b) Give one common use case for each transformer variant below.
   i. encoder-only
   ii. decoder-only
   iii. encoder-decoder

(c) In the decoder, why do we apply a *causal mask* in the self-attention block for tasks such as machine translation?

(d) Do we also apply a *causal mask* in the cross-attention block of an encoder-decoder transformer for machine translation? Why or why not?

Contributors:

- Eric Kim.
- Terry Kim.

