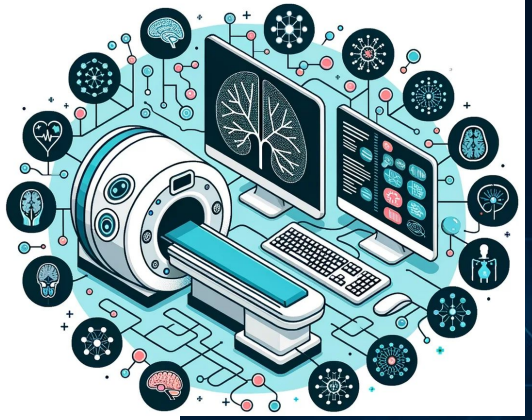
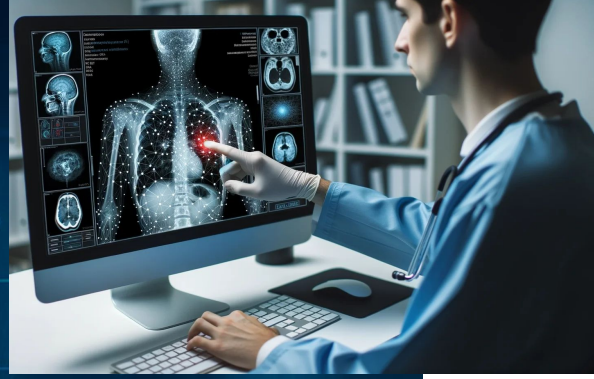




Frontiers in Generative AI for Medical Imaging and Healthcare

Session 3

Retrieval-Augmented Generation & Clinical NLP

Sep 20 2025

Recurrent Neural Networks

output distribution

$$\hat{\boldsymbol{y}}^{(t)} = \text{softmax}\left(\boldsymbol{U}\boldsymbol{h}^{(t)} + \boldsymbol{b}_2\right) \in \mathbb{R}^{|V|}$$

hidden states

$$\boldsymbol{h}^{(t)} = \sigma\left(\boldsymbol{W}_h\boldsymbol{h}^{(t-1)} + \boldsymbol{W}_e\boldsymbol{e}^{(t)} + \boldsymbol{b}_1\right)$$

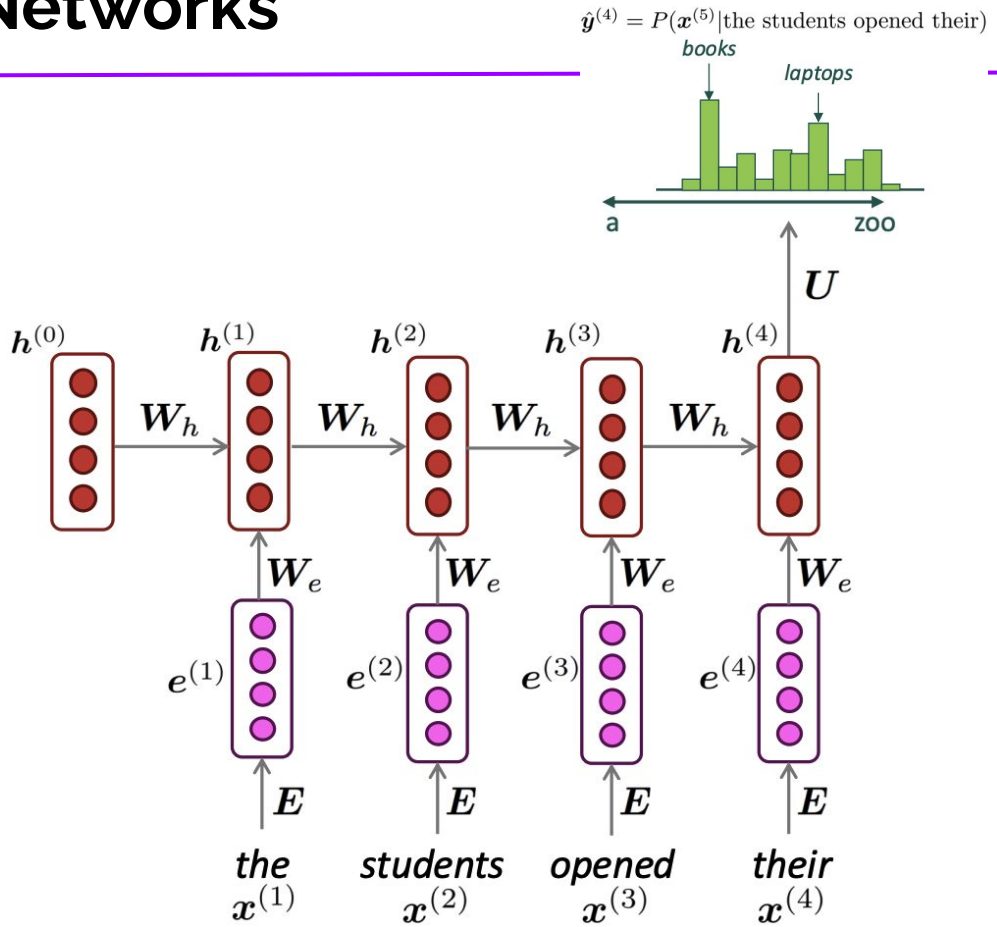
$\boldsymbol{h}^{(0)}$ is the initial hidden state

word embeddings

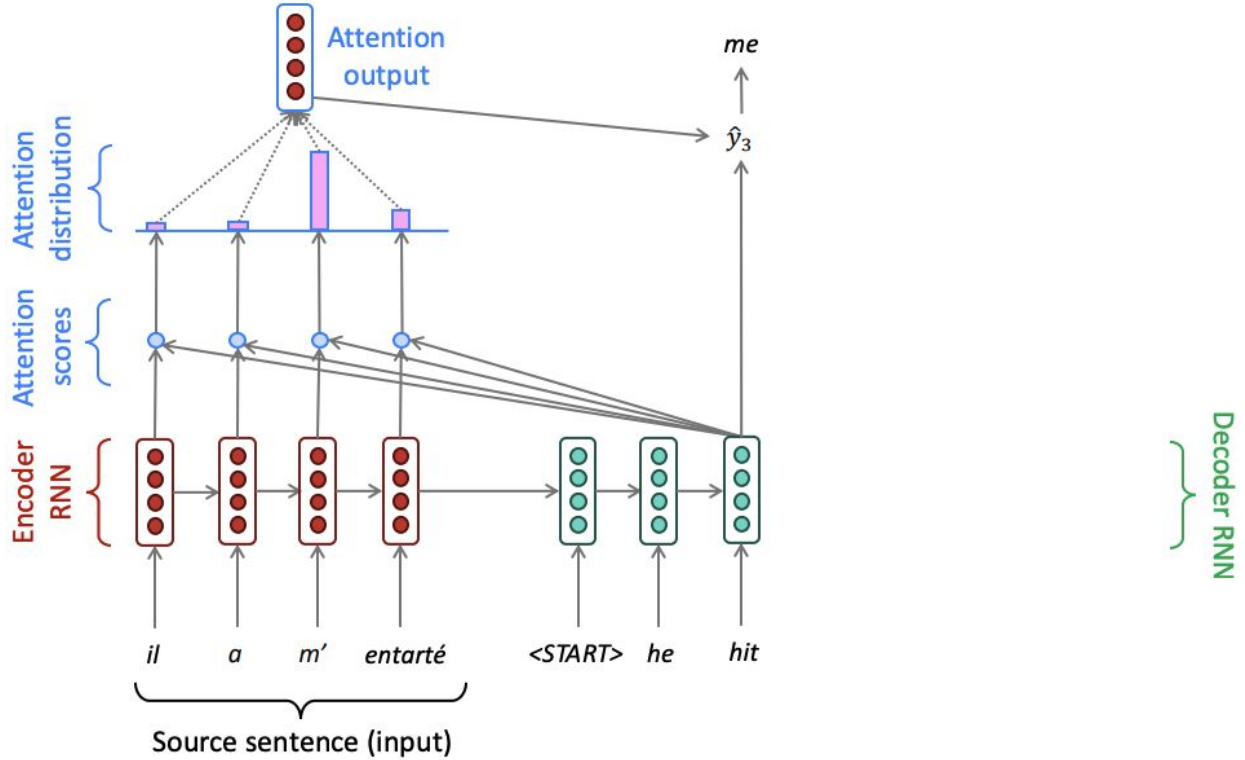
$$\boldsymbol{e}^{(t)} = \boldsymbol{E}\boldsymbol{x}^{(t)}$$

words / one-hot vectors

$$\boldsymbol{x}^{(t)} \in \mathbb{R}^{|V|}$$



Attention in RNNs



Self-attention: Keys, Queries, and Values

Let $\boldsymbol{w}_{1:n}$ be a sequence of words in vocabulary V, like *Zuko made his uncle tea*.

For each $\boldsymbol{w}_i$, let $\boldsymbol{x}_i = E\boldsymbol{w}_i$, where $E \in \mathbb{R}^{d \times |V|}$ is an embedding matrix.

1. Transform each word embedding with weight matrices $\mathrm{Q}, \mathrm{K}, \mathrm{V}$, each in $\mathbb{R}^{d \times d}$

$\boldsymbol{q}_i = Q\boldsymbol{x}_i$ (queries) $\quad \boldsymbol{k}_i = K\boldsymbol{x}_i$ (keys) $\quad \boldsymbol{v}_i = V\boldsymbol{x}_i$ (values)

2. Compute pairwise similarities between keys and queries; normalize with softmax

$$\boldsymbol{e}_{ij} = \boldsymbol{q}_i^\top \boldsymbol{k}_j \qquad \boldsymbol{\alpha}_{ij} = \frac{\exp(\boldsymbol{e}_{ij})}{\sum_{j'} \exp(\boldsymbol{e}_{ij'})}$$

3. Compute output for each word as weighted sum of values

$$\boldsymbol{o}_i = \sum_j \boldsymbol{\alpha}_{ij} \boldsymbol{v}_i$$

Positional Embedding in Self-Attention

- Since self-attention doesn't build in order information, we need to encode the order of the sentence in our keys, queries, and values
- Consider representing each sequence index as a vector

$$\boldsymbol{p}_i \in \mathbb{R}^d, \text{ for } i \in \{1,2,\ldots,n\} \text{ are position vectors}$$

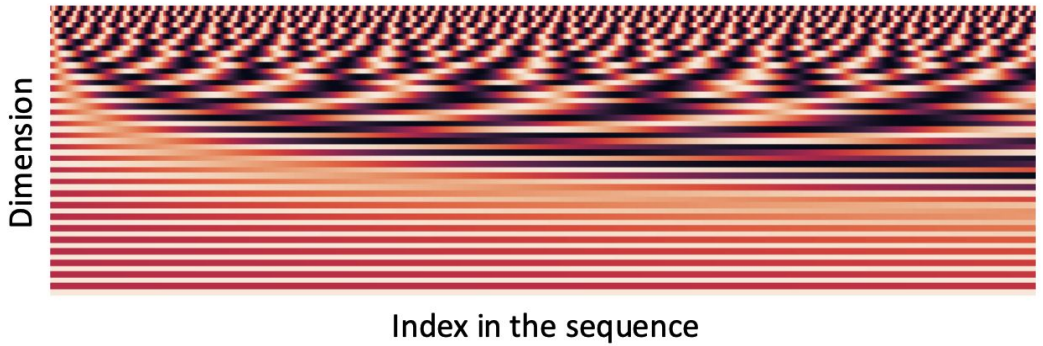
$$\tilde{\boldsymbol{x}}_i = \boldsymbol{x}_i + \boldsymbol{p}_i$$

In deep self-attention networks, we do this at the first layer! You could concatenate them as well, but people mostly just add...

Sinusoida Positional Embedding

- Sinusoidal position representations: concatenate sinusoidal functions of varying periods
- Periodicity indicates that maybe "absolute position" isn't as important
- It can extrapolate to longer sequences as periods restart!

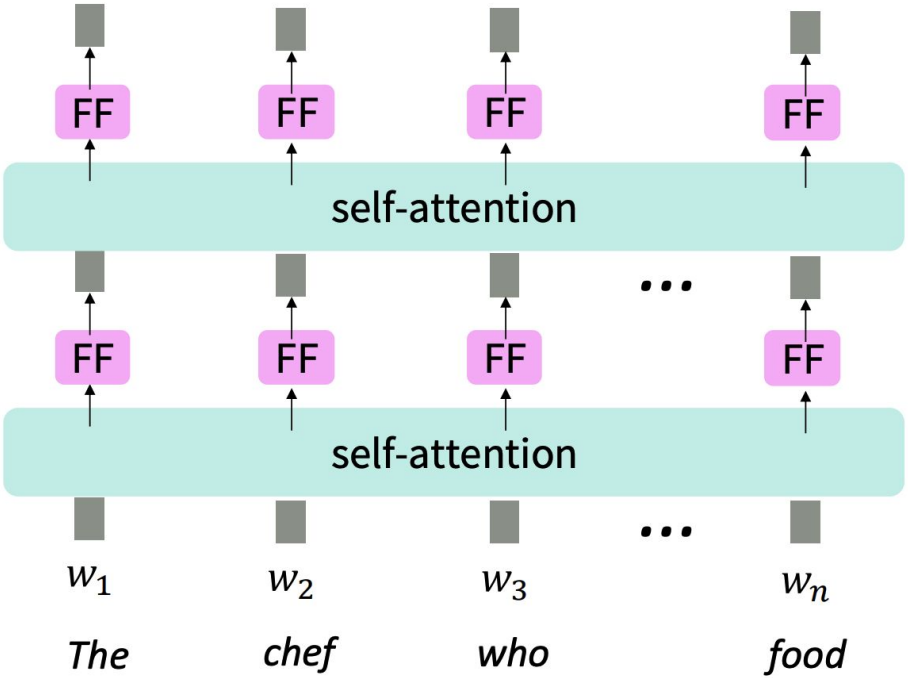
$$\boldsymbol{p}_i = \begin{pmatrix} \sin(i/10000^{2*1/d}) \\ \cos(i/10000^{2*1/d}) \\ \vdots \\ \sin(i/10000^{2*\frac{d}{2}/d}) \\ \cos(i/10000^{2*\frac{d}{2}/d}) \end{pmatrix}$$



Non-Linearity in Self-Attention

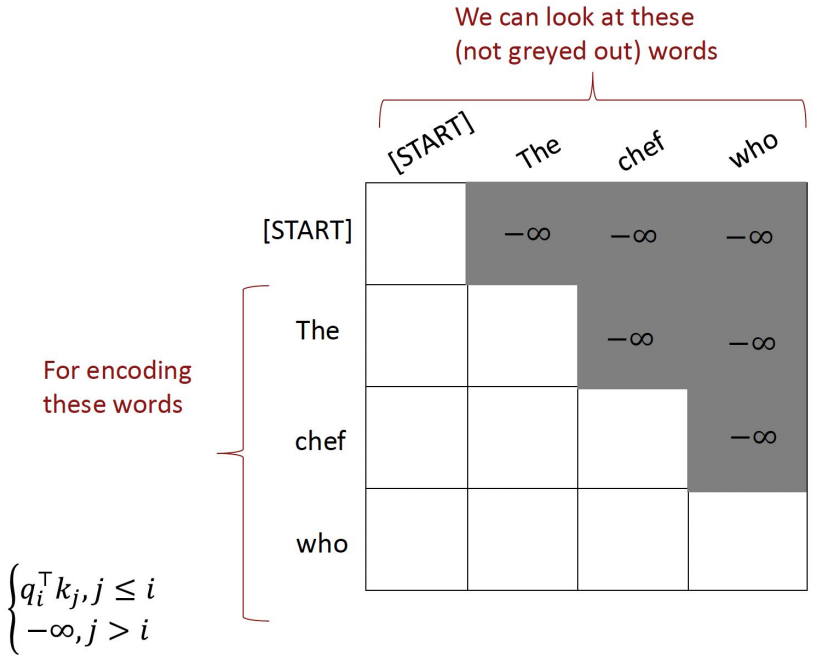
- Easy fix: add a feed-forward network to post-process each output vector.

$$m_i = MLP(\text{output}_i)$$
$$= W_2 * \text{ReLU}(W_1\ \text{output}_i + b_1) + b_2$$



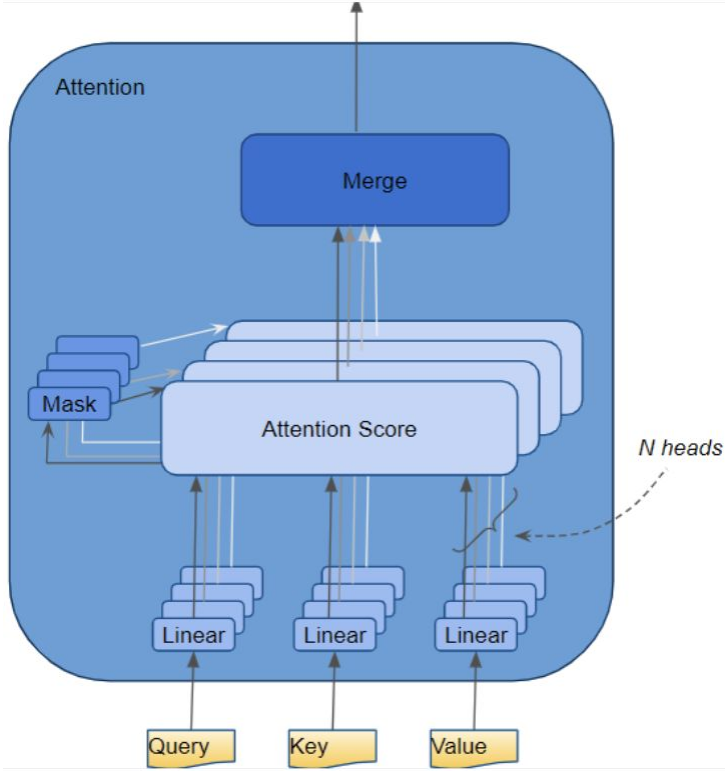
Causal Masking in Self-Attention

- For causality, we need to ensure not to peek at the future.
- At each timestep, we could change the set of keys and queries to only include past words!



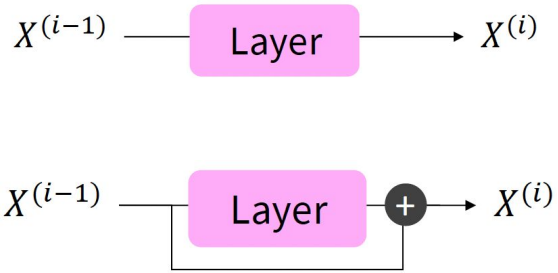
Multi-Head Self-Attention Layer

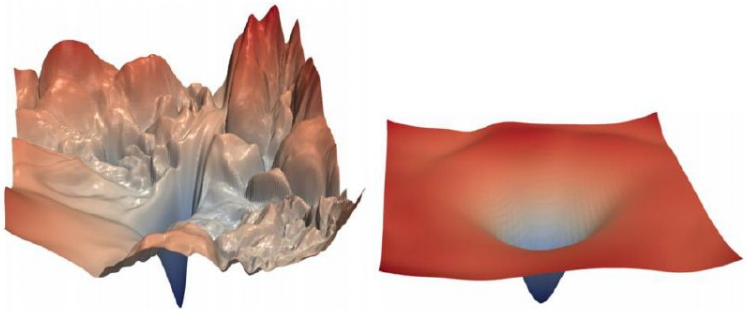
- The Attention module splits its Query, Key, and Value parameters N-ways and passes each split independently through a separate head.
- Calculations are combined together to produce a final attention score.
- Greater power to encode multiple relationships and nuances for each word.



Residual Connections

- A trick to help models learn better!
- Gradient is 1 through residual connection
- Bias toward identity function.





[no residuals] [residuals]

[Loss landscape visualization, Li et al., 2018, on a ResNet]

Layer Normalization

- A trick to help models train faster.
- Cut down on uninformative variation in hidden vectors by normalizing to unit mean and standard deviation within each layer.

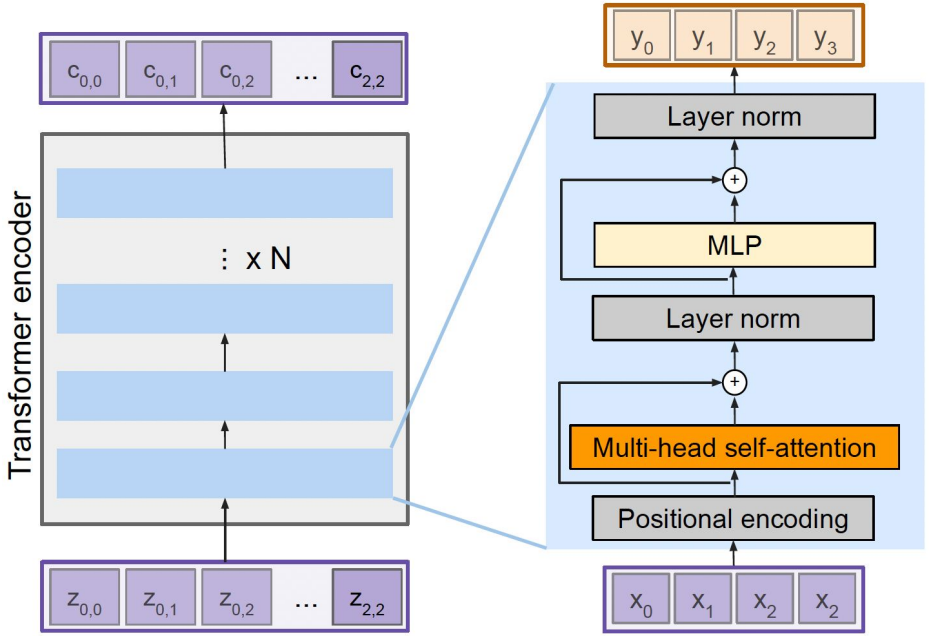
$$\text{output} = \frac{x - \mu}{\sqrt{\sigma} + \epsilon} * \gamma + \beta$$

Normalize by scalar mean and variance

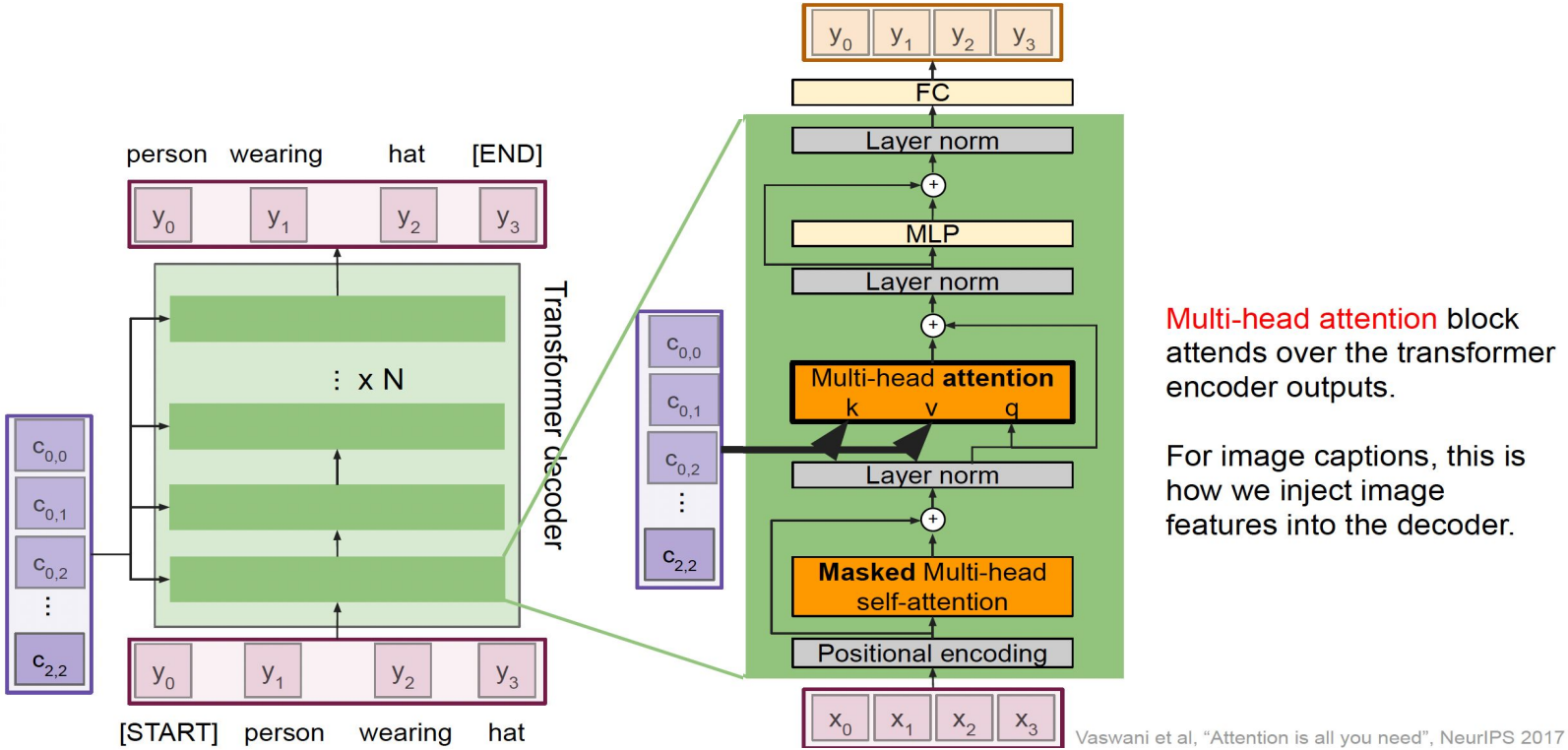
Modulate by learned elementwise gain and bias

Transformer Encoder

- Position representation
  - Specify the sequence order, since self-attention is an unordered function of its inputs.
- Nonlinearities
  - Frequently implemented as a simple feedforward network.
- Masking
  - Keep information about the future from “leaking” to the past.



Transformer Decoder



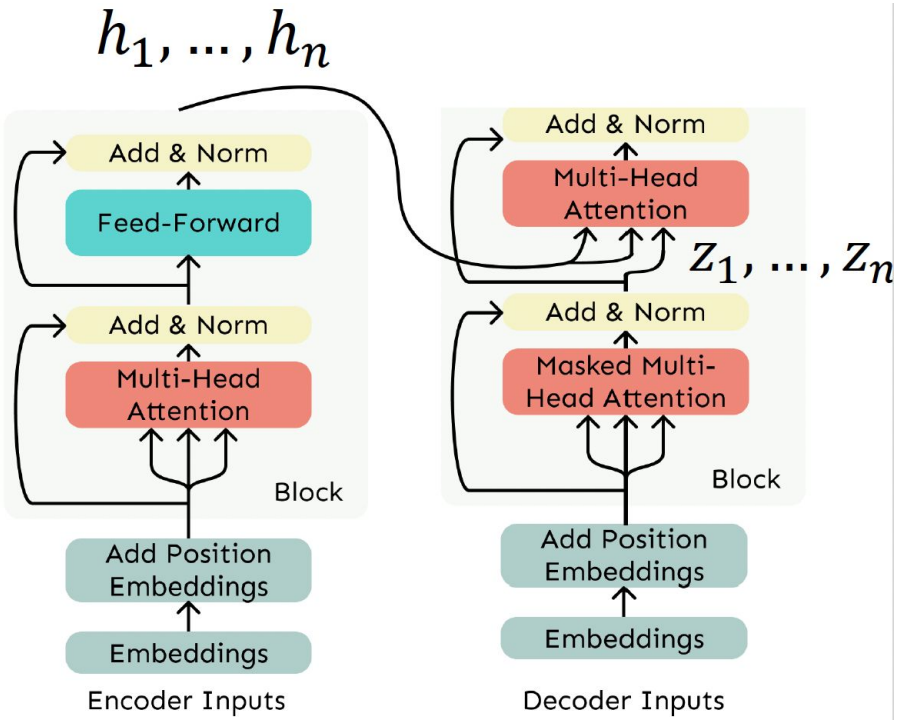
Vaswani et al, "Attention is all you need", NeurIPS 2017

Multi-head attention block attends over the transformer encoder outputs.

For image captions, this is how we inject image features into the decoder.

Cross Attention

- Self-attention:
  - Keys, queries, and values from same source
- Cross-attention
  - The keys and values are from encoder (like a memory)
  - The queries are from the decoder




Inside an
LLM