

CPEN 400D: Deep Learning

Lecture 5: Transformers

Renjie Liao

University of British Columbia

Winter, Term 2, 2022

Outline

- Applications and Challenges of Sequence Modeling
- Transformers
 - Positional Encoding
 - Encoder
 - Multi-head Self-Attention
 - Decoder
- Limitations & Variants
 - Pre-norm vs. Post-norm
 - Vision Transformer
 - Swin Transformer

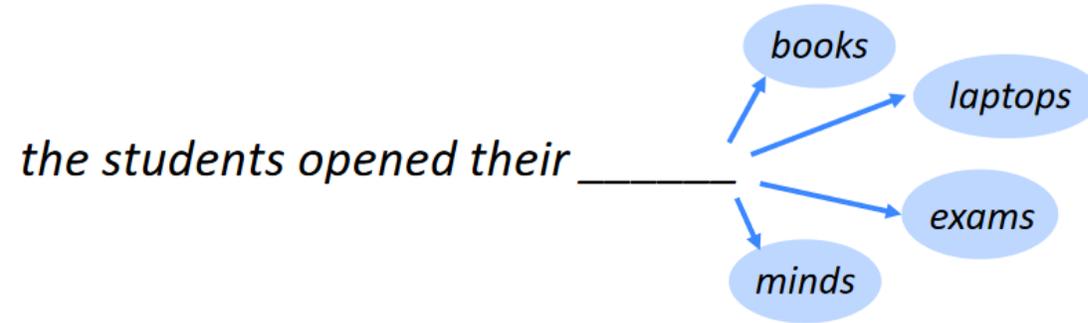
Outline

- **Applications and Challenges of Sequence Modeling**
- Transformers
 - Positional Encoding
 - Encoder
 - Multi-head Self-Attention
 - Decoder
- Limitations & Variants
 - Pre-norm vs. Post-norm
 - Vision Transformer
 - Swin Transformer

Deep Learning for Sequences

- Language Models

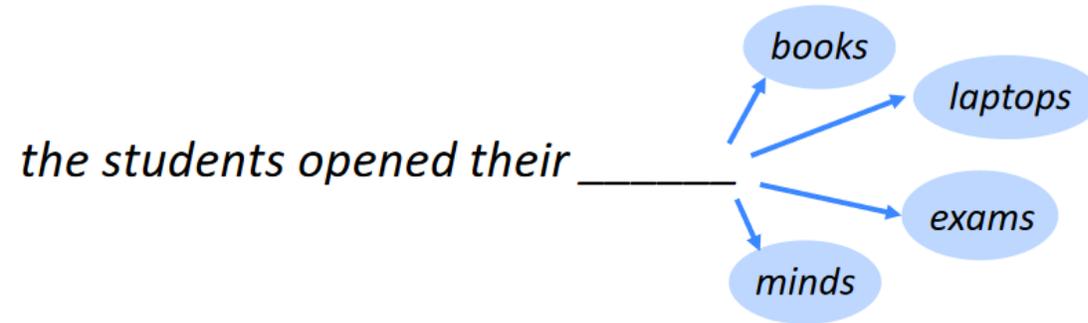
$$P(\mathbf{x}^{(t+1)} | \mathbf{x}^{(t)}, \dots, \mathbf{x}^{(1)})$$



Deep Learning for Sequences

- Language Models

$$P(\mathbf{x}^{(t+1)} | \mathbf{x}^{(t)}, \dots, \mathbf{x}^{(1)})$$



- Machine Translation



Deep Learning for Sequences

Key Challenges:

- Varying-sized input sequences

Deep Learning for Sequences

Key Challenges:

- Varying-sized input sequences
- Orders “may” be crucial for cognition

Aoccdrnig to a rscheearch at Cmabrigde Uinervtisy, it deosn't mttar in waht oredr the ltteers in a wrod are, the olny iprmoetnt tihng is taht the frist and lsat ltteer be at the rghit pclae. The rset can be a toatl mse and you can sitll raed it wouthit porbelm. Tihs is bcuseae the huamn mnid deos not raed ervey lteter by istlef, but the wrod as a wlohe.

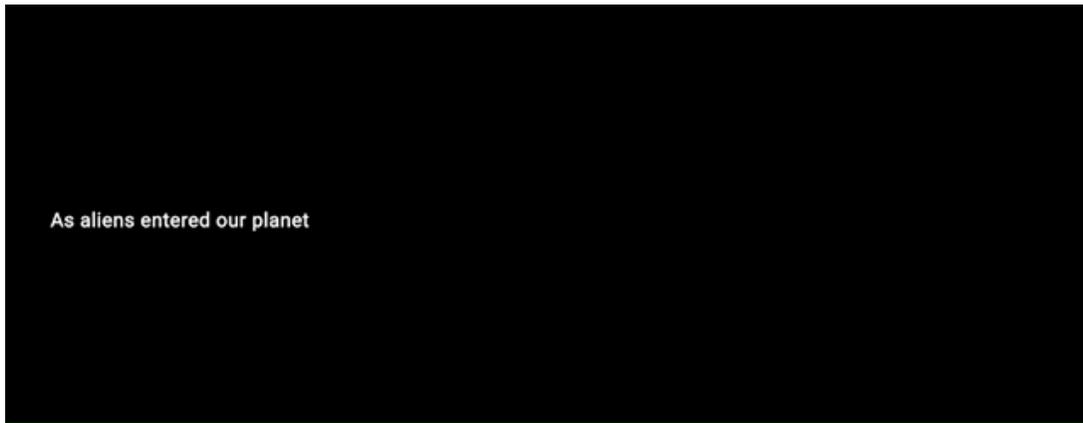
Deep Learning for Sequences

Key Challenges:

- Varying-sized input sequences
- Orders “may” be crucial for cognition

Aoccdrnig to a rscheearch at Cmabrigde Uinervtisy, it deosn't mttar in waht oredr the ltteers in a wrod are, the olny iprmoetnt tihg is taht the frist and lsat ltteer be at the rghit pclae. The rset can be a toatl mses and you can sitll raed it wouthit porbelm. Tihs is bcuseae the huamn mnid deos not raed ervey lteter by istlef, but the wrod as a wlohe.

- Complex statistical dependencies (e.g. long-range ones)



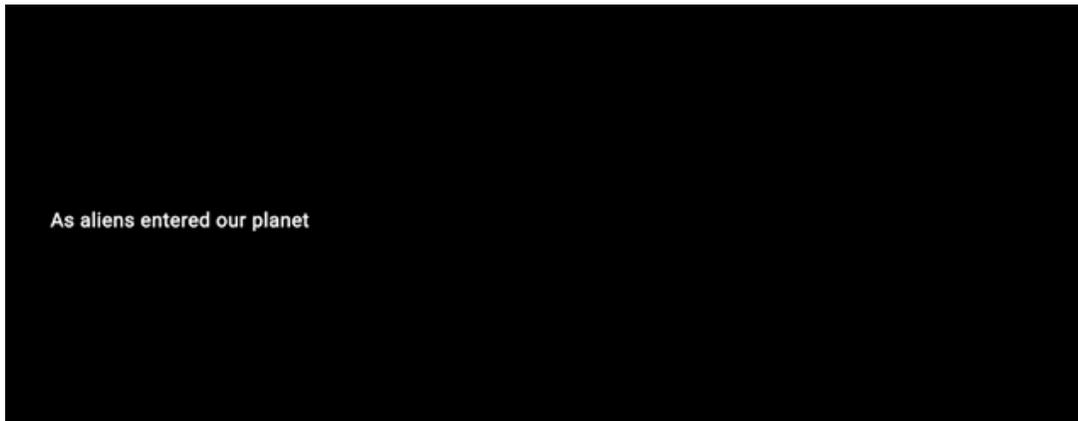
Deep Learning for Sequences

Key Challenges:

- Varying-sized input sequences
- Orders “may” be crucial for cognition

Aoccdrnig to a rscheearch at Cmabrigde Uinervtisy, it deosn't mttar in waht oredr the ltteers in a wrod are, the olny iprmoetnt tihng is taht the frist and lsat ltteer be at the rghit pclae. The rset can be a toatl mses and you can sitll raed it wouthit porbelm. Tihs is bcuseae the huamn mnid deos not raed ervey lteter by istlef, but the wrod as a wlohe.

- Complex statistical dependencies (e.g. long-range ones)

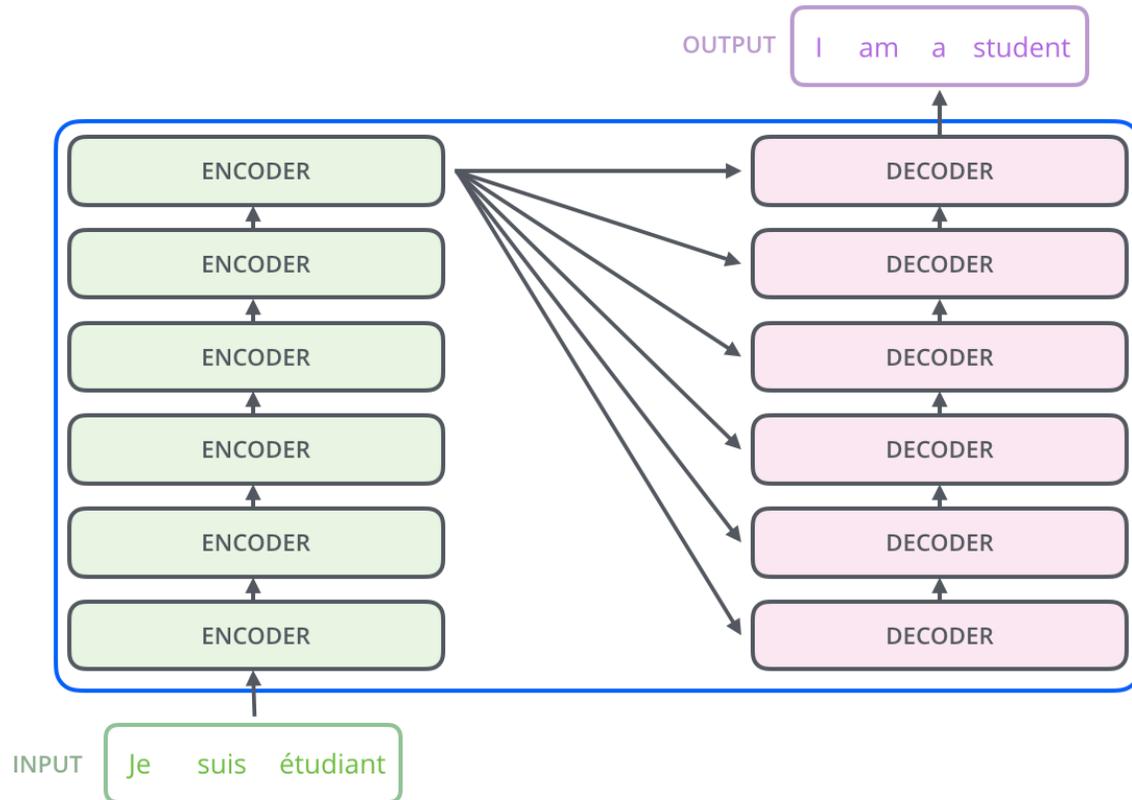


Transformer [5]

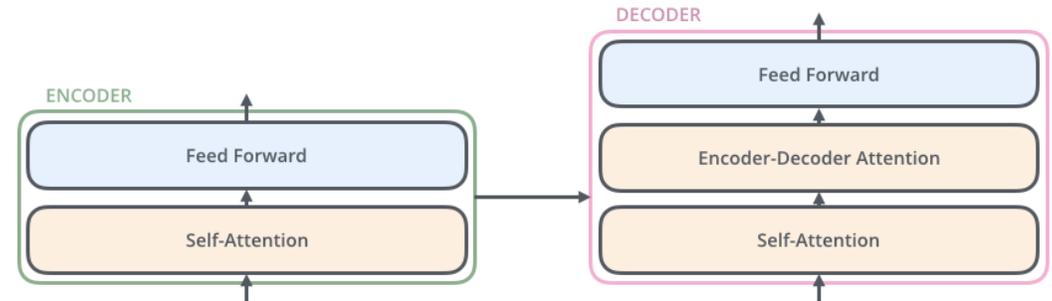
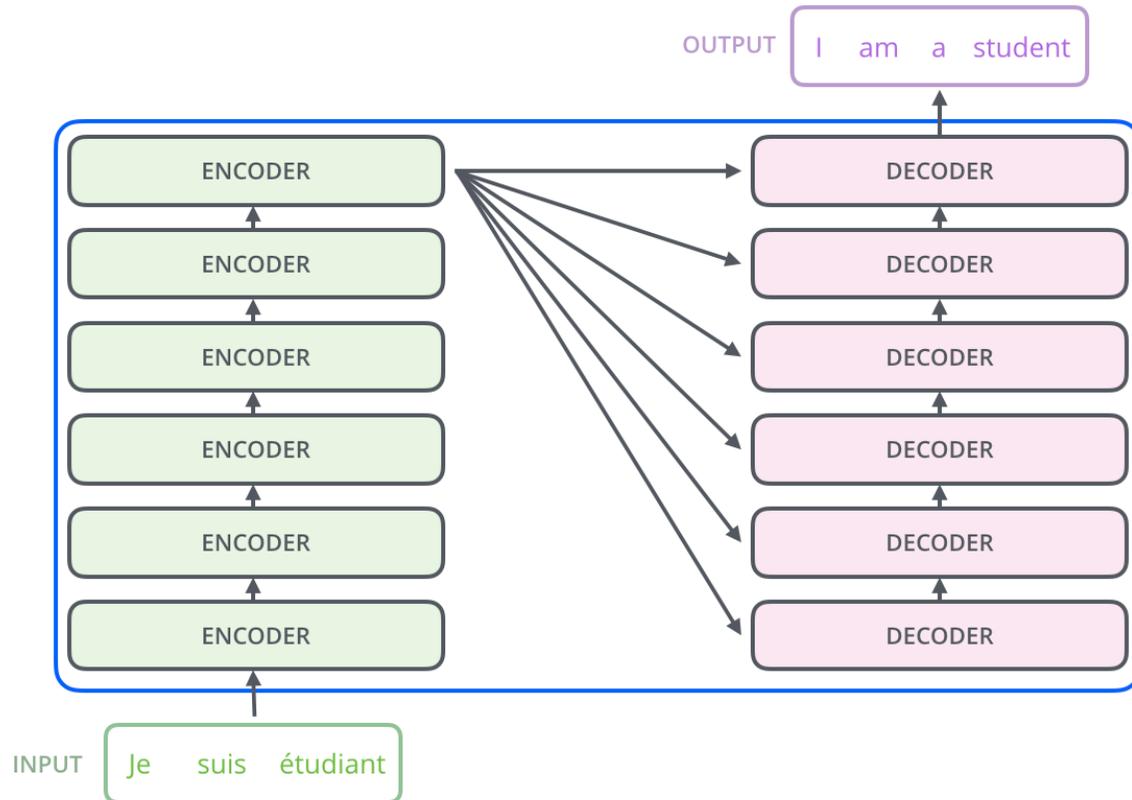
Outline

- Applications and Challenges of Sequence Modeling
- **Transformers**
 - Positional Encoding
 - Encoder
 - Multi-head Self-Attention
 - Decoder
- Limitations & Variants
 - Pre-norm vs. Post-norm
 - Vision Transformer
 - Swin Transformer

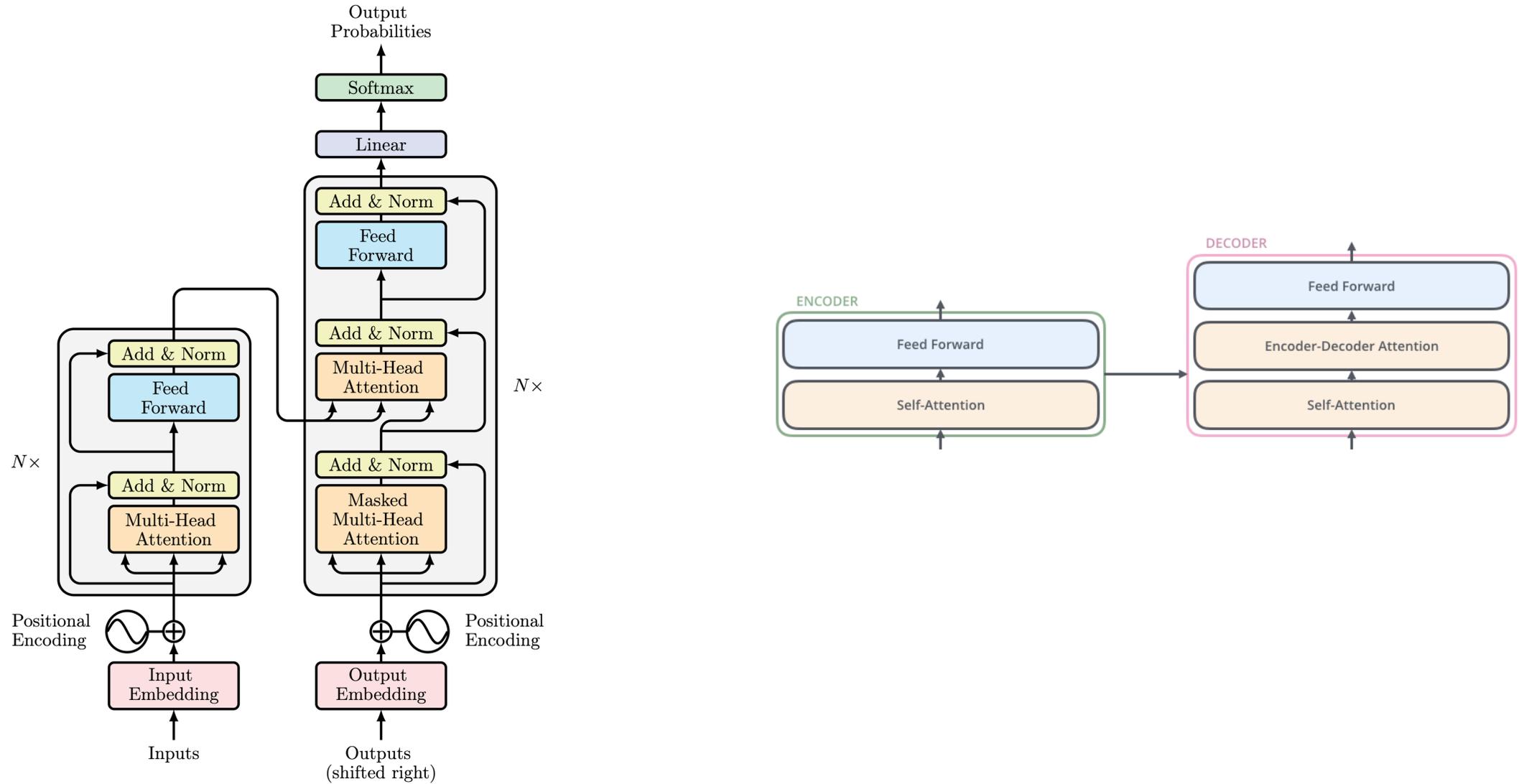
Transformers



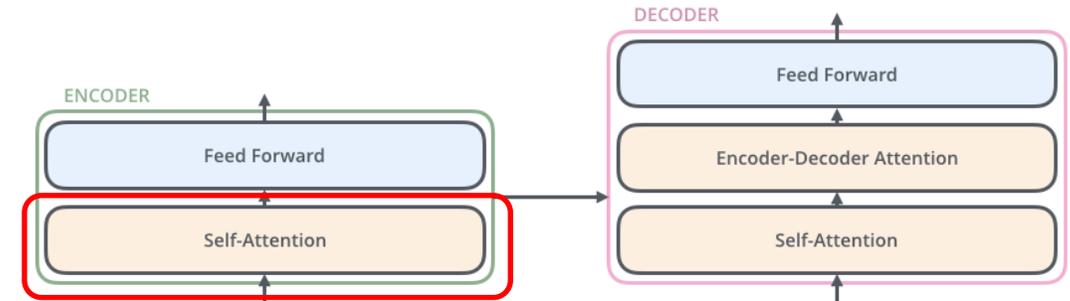
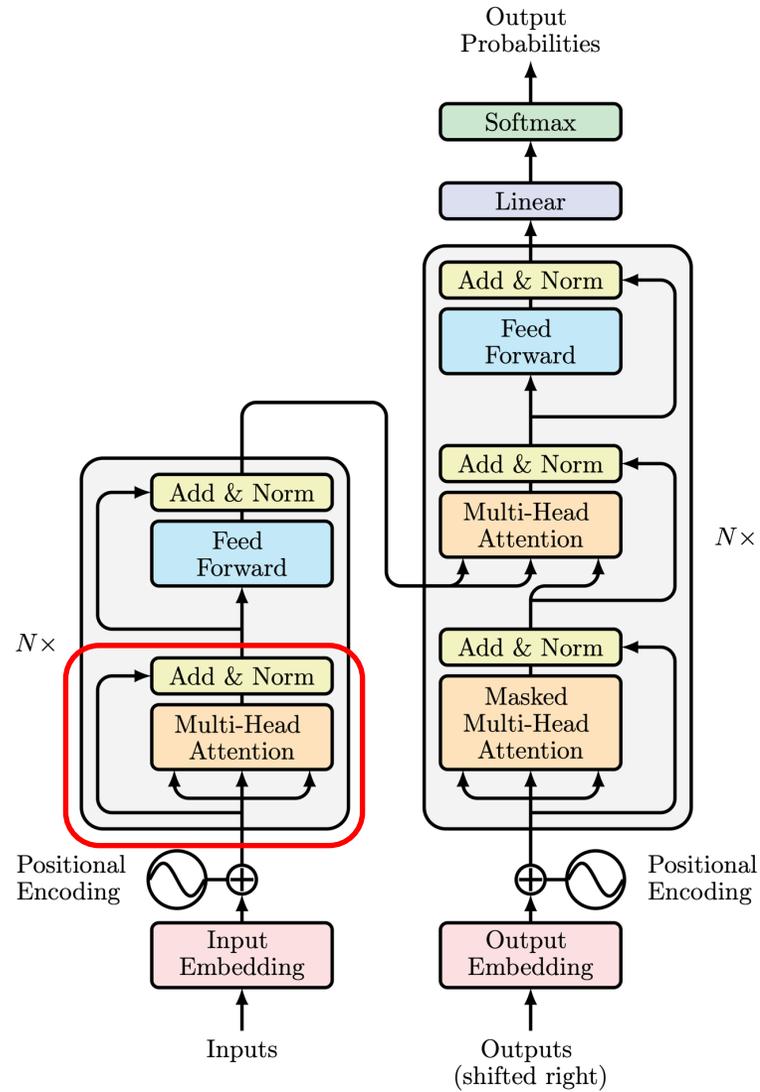
Transformers



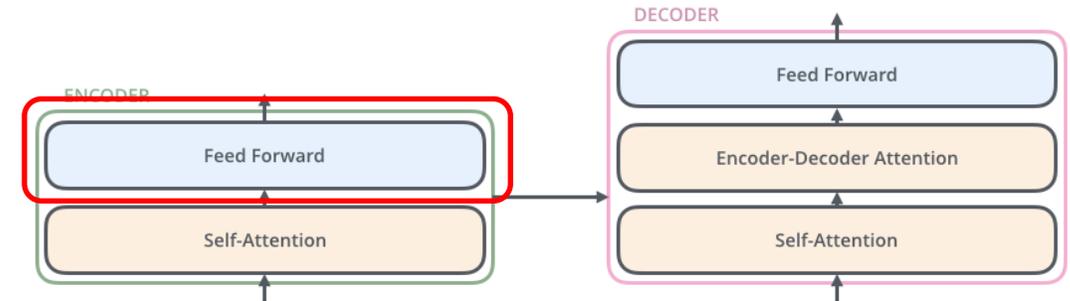
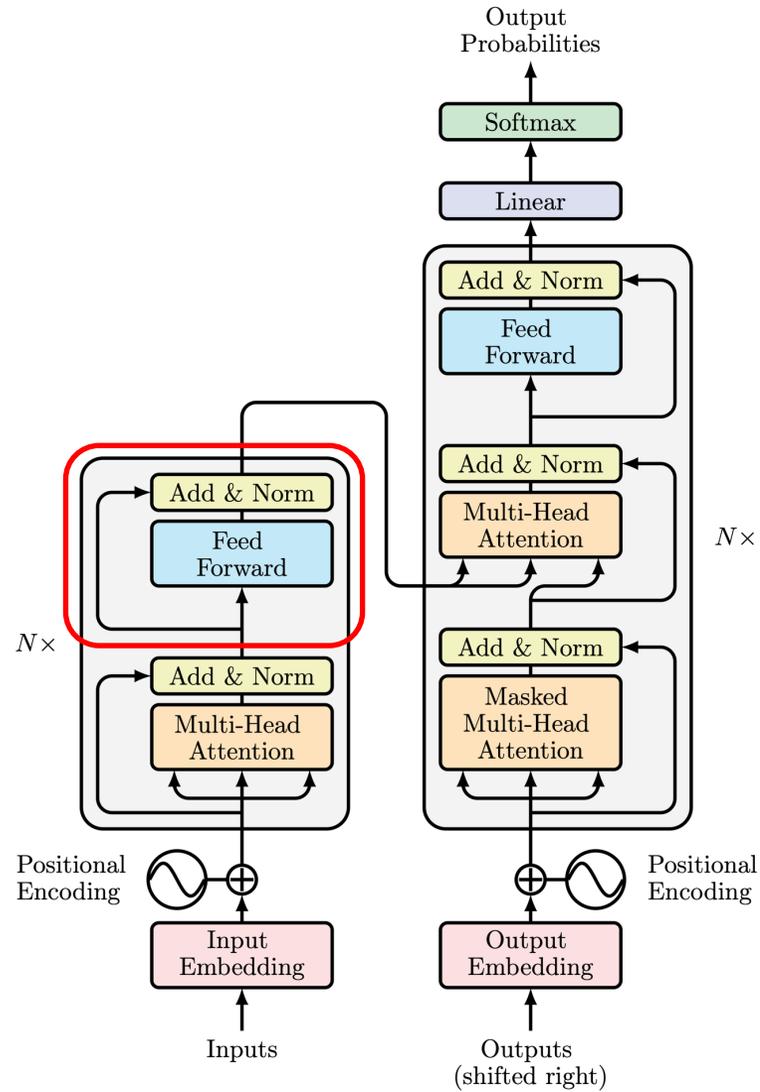
Transformers



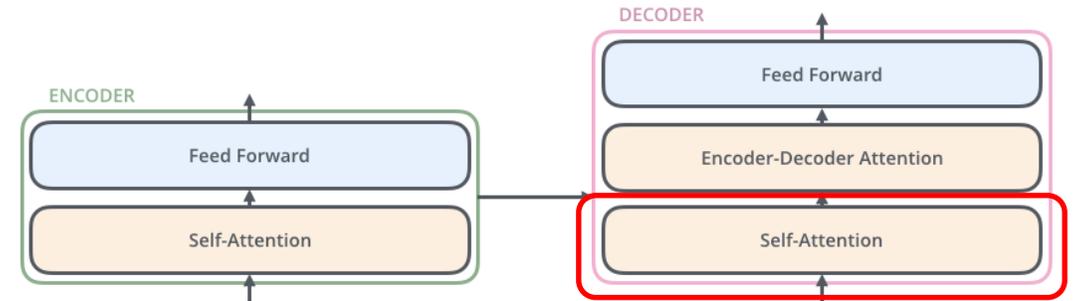
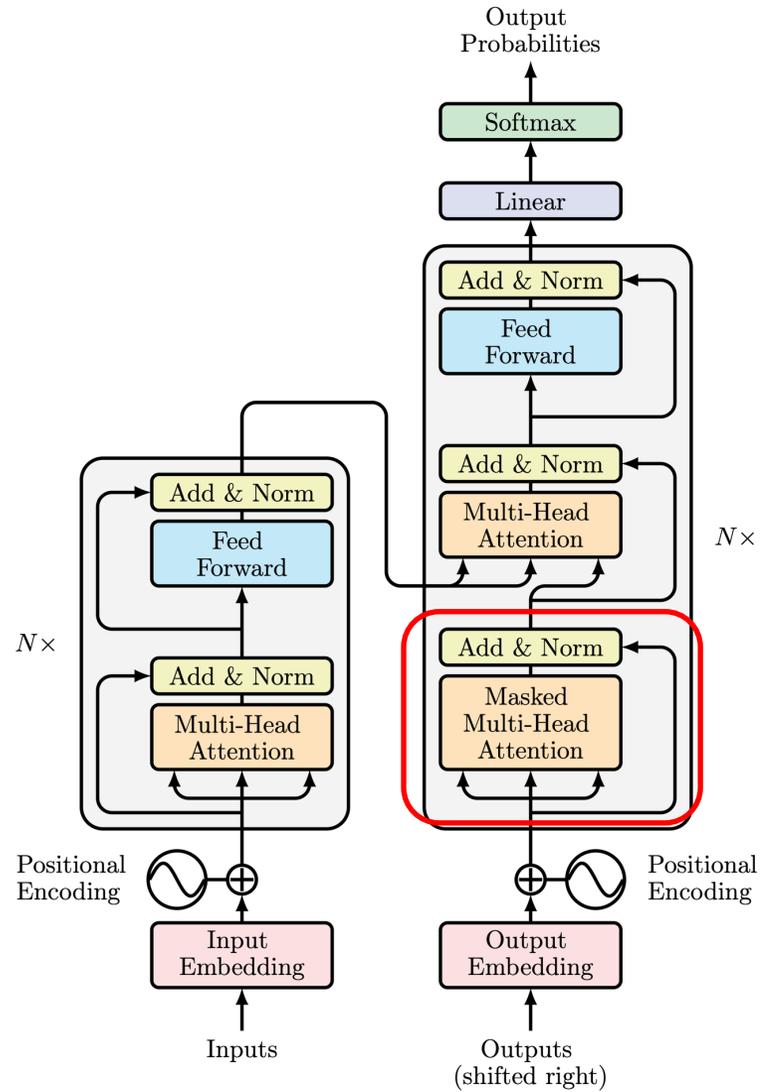
Transformers



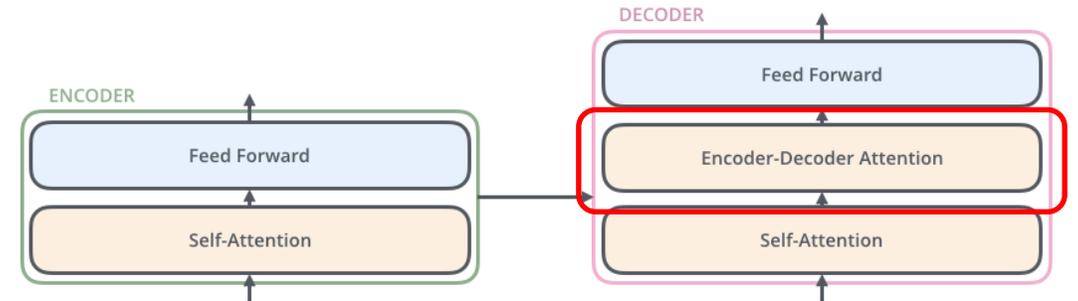
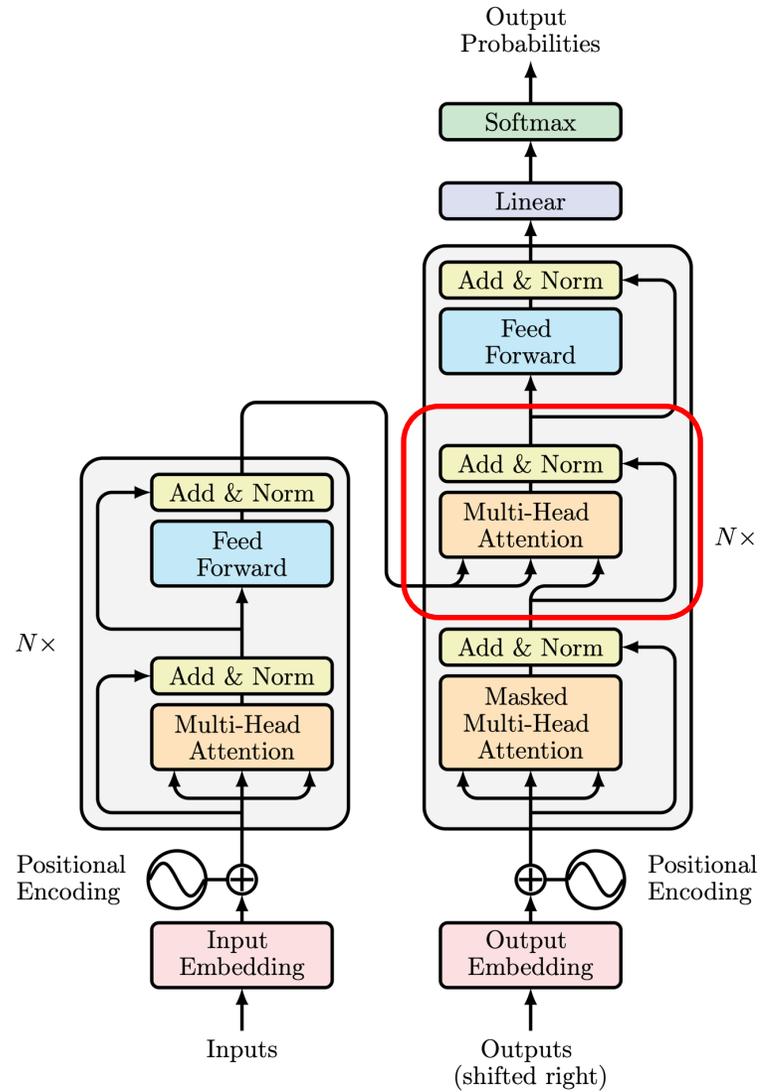
Transformers



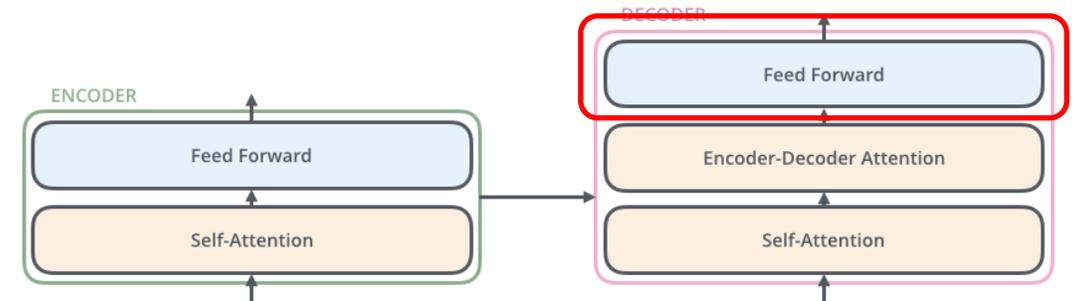
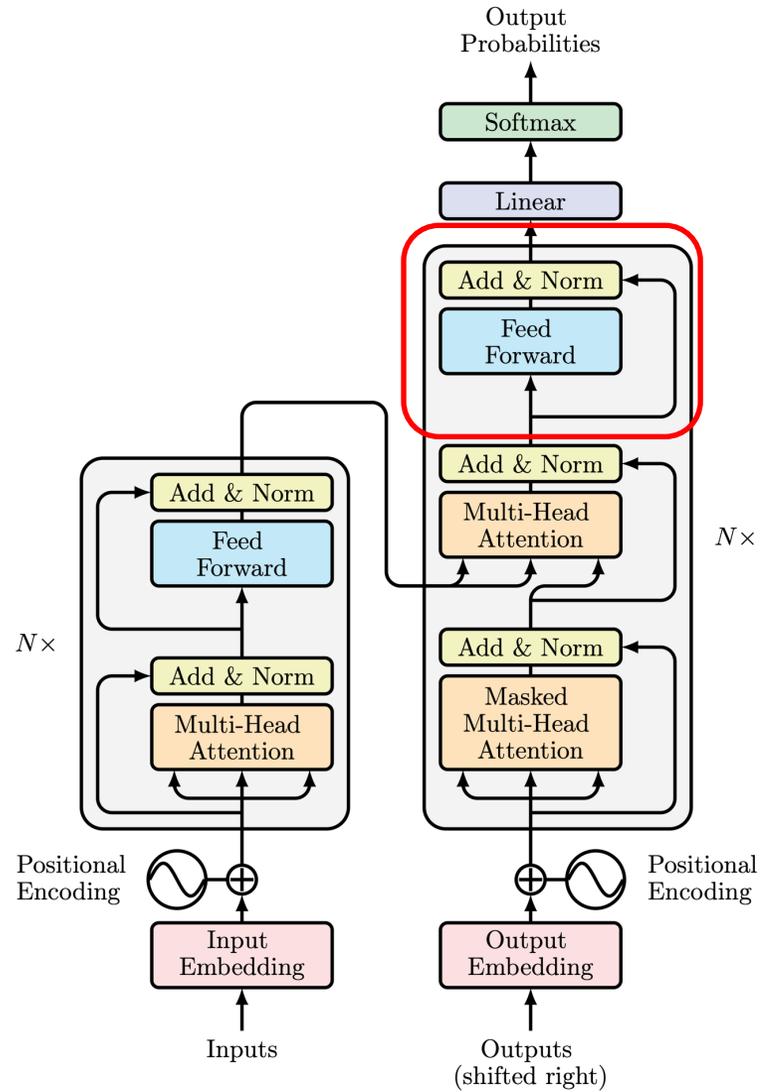
Transformers



Transformers



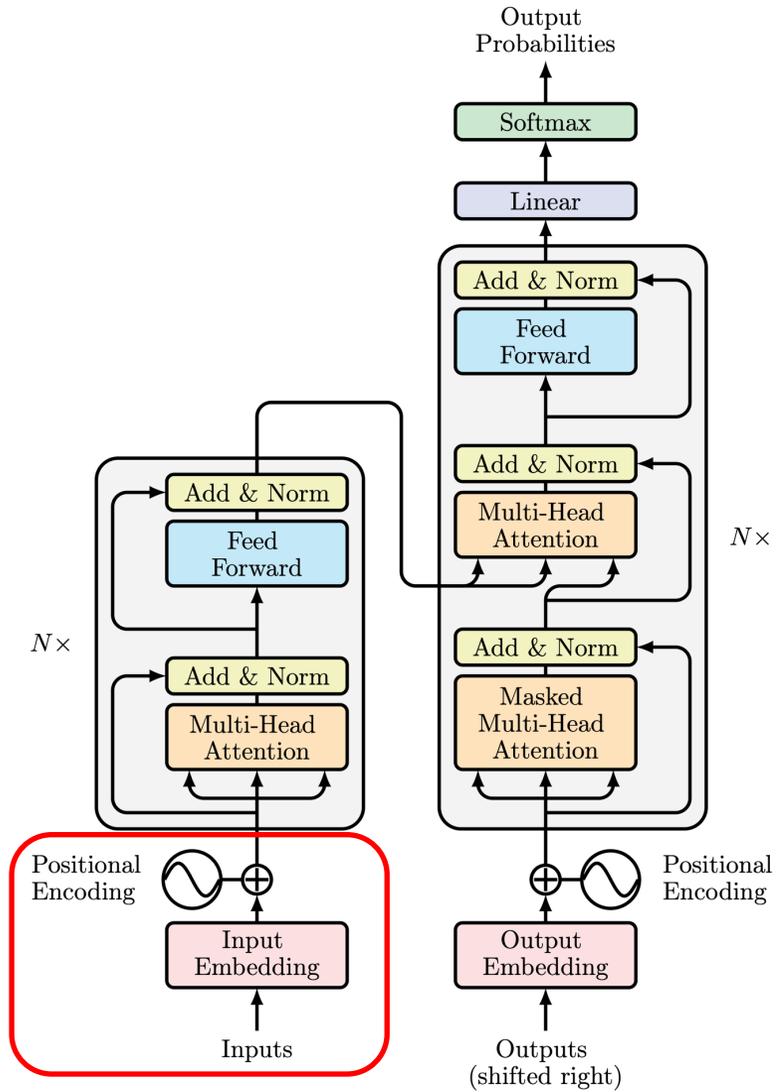
Transformers



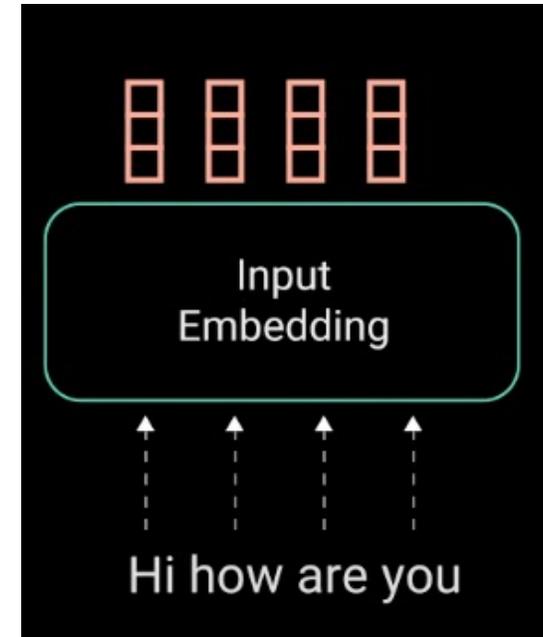
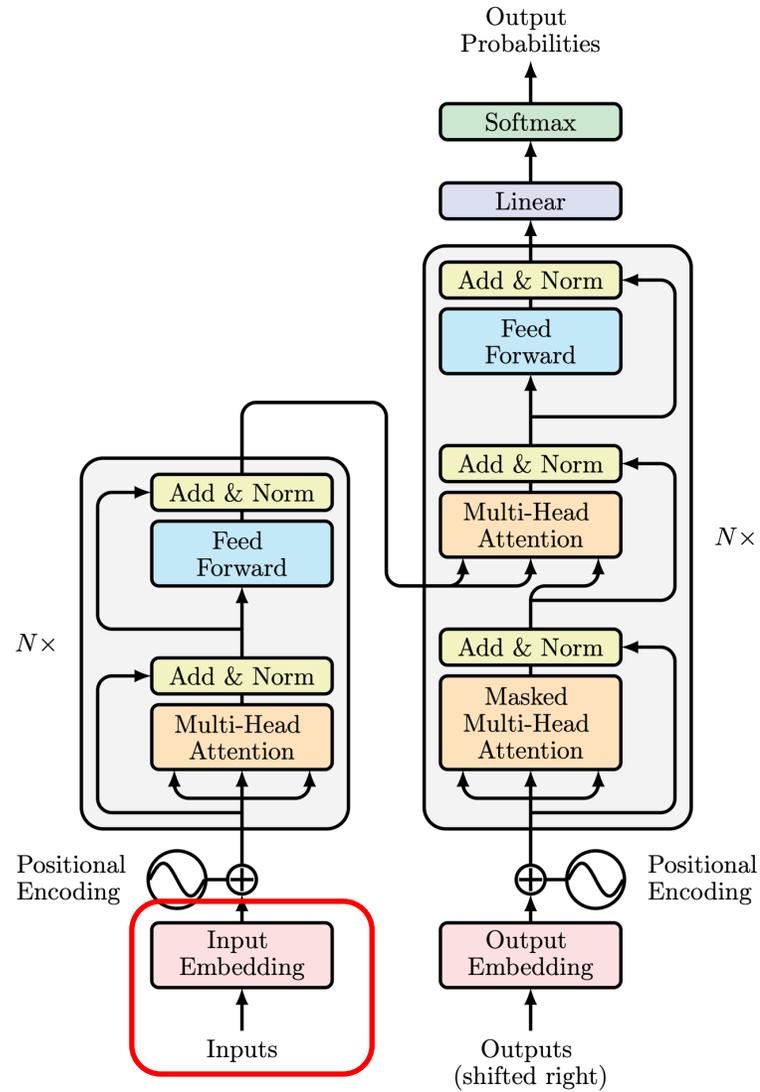
Outline

- Applications and Challenges of Sequence Modeling
- Transformers
 - **Positional Encoding**
 - Encoder
 - Multi-head Self-Attention
 - Decoder
- Limitations & Variants
 - Pre-norm vs. Post-norm
 - Vision Transformer
 - Swin Transformer

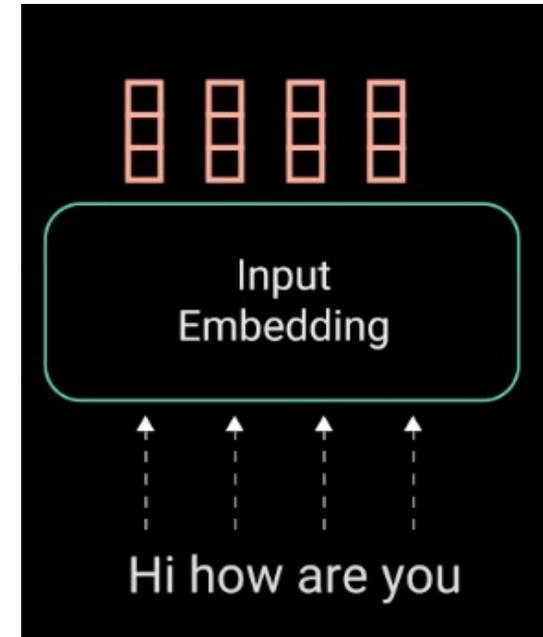
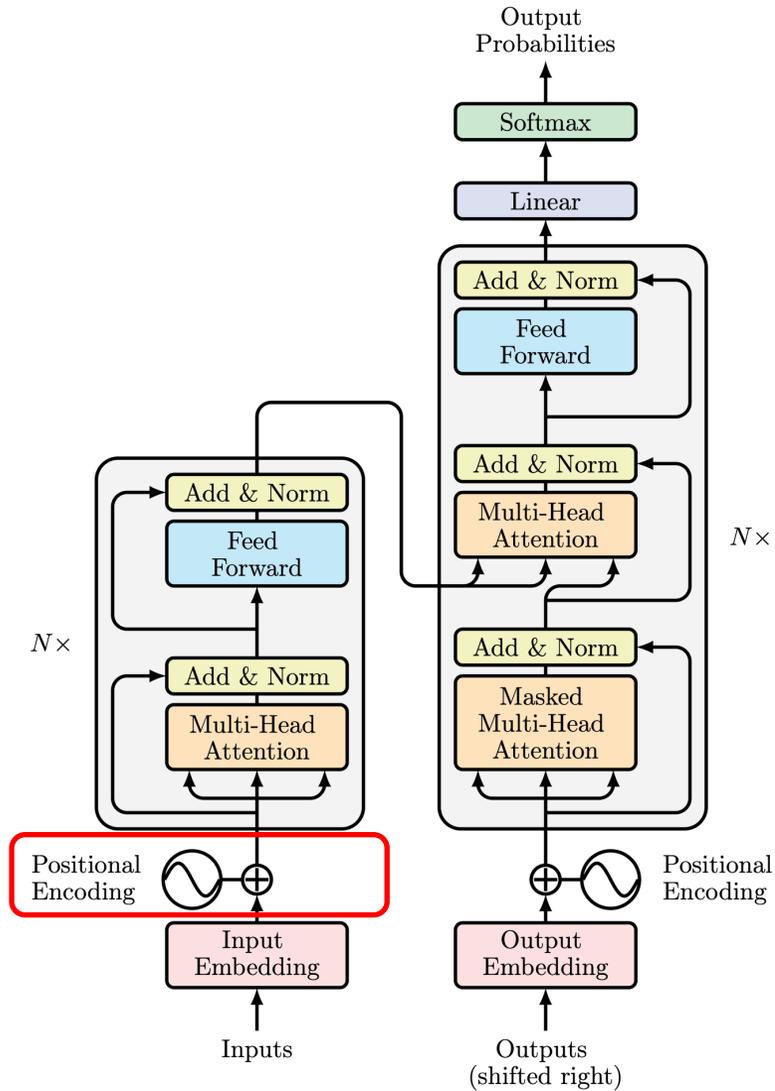
Input Encoding



Input Embedding



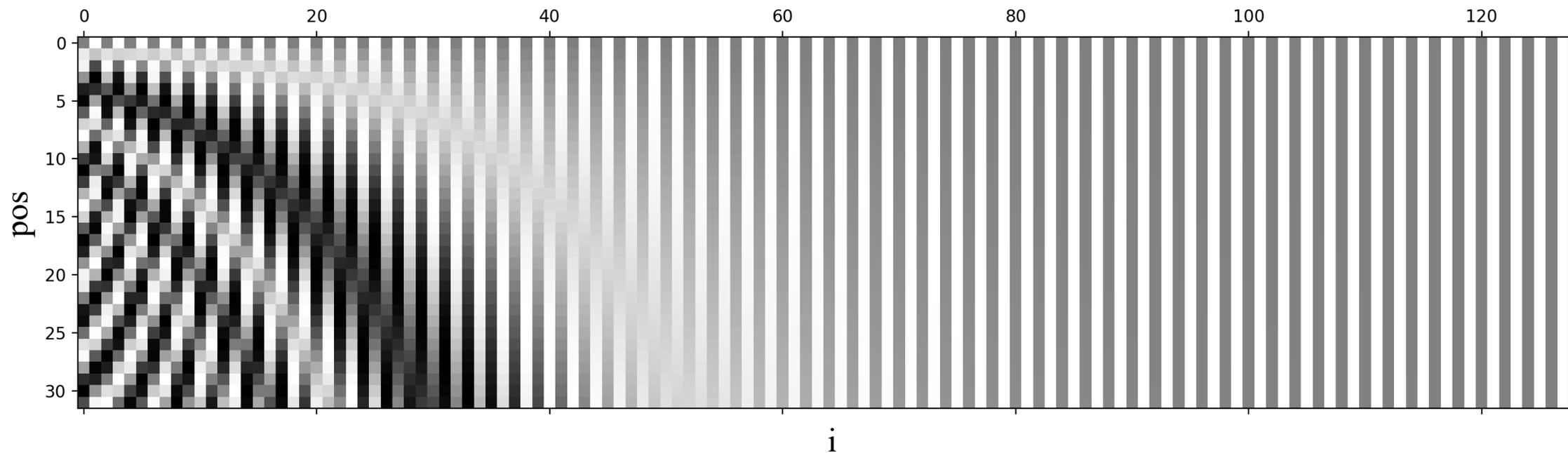
Positional Encoding



$$PE_{(pos,2i)} = \sin(pos/10000^{2i/d_{model}})$$

$$PE_{(pos,2i+1)} = \cos(pos/10000^{2i/d_{model}})$$

Positional Encoding



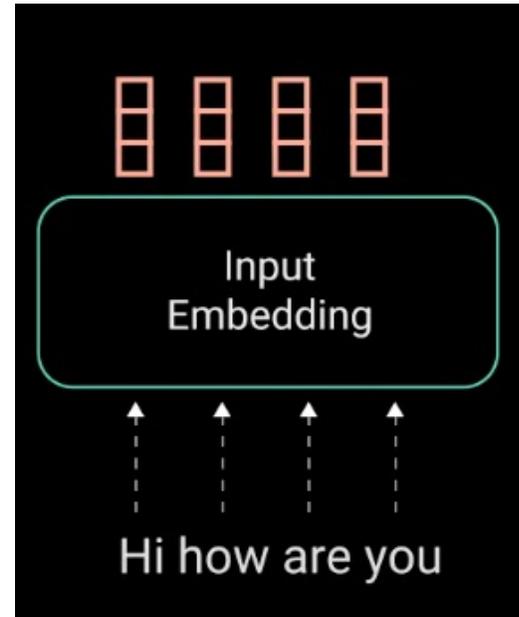
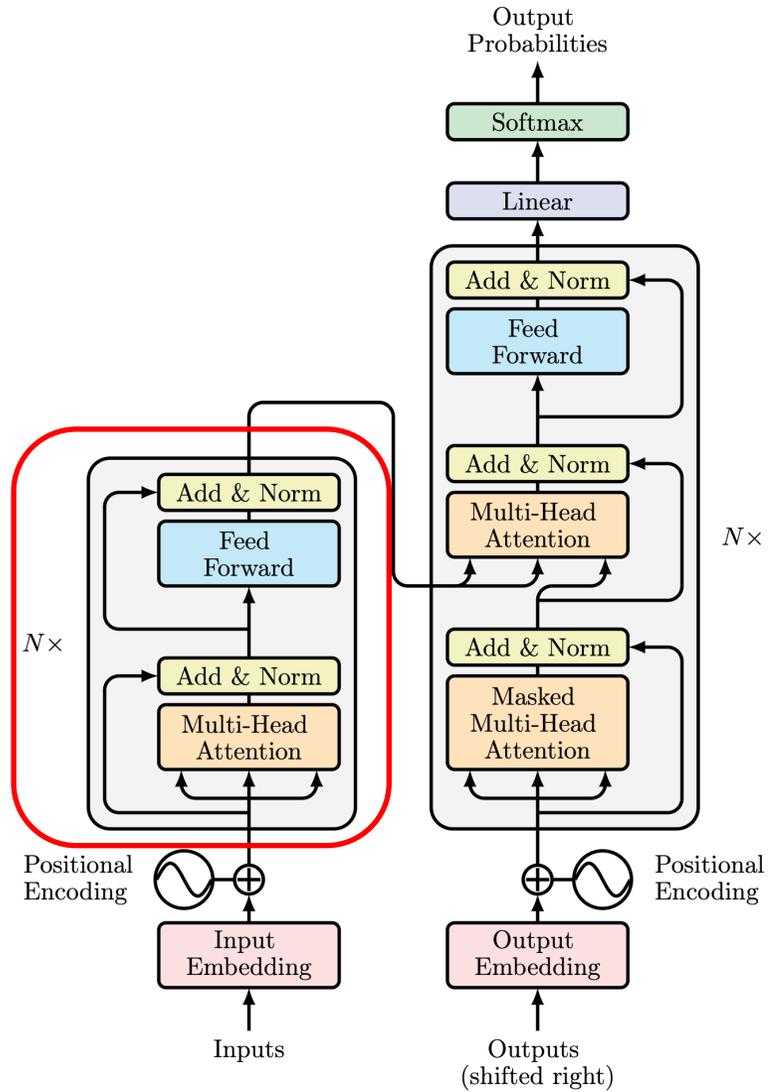
$$PE_{(pos,2i)} = \sin(pos/10000^{2i/d_{model}})$$

$$PE_{(pos,2i+1)} = \cos(pos/10000^{2i/d_{model}})$$

Outline

- Applications and Challenges of Sequence Modeling
- Transformers
 - Positional Encoding
 - **Encoder**
 - Multi-head Self-Attention
 - Decoder
- Limitations & Variants
 - Pre-norm vs. Post-norm
 - Vision Transformer
 - Swin Transformer

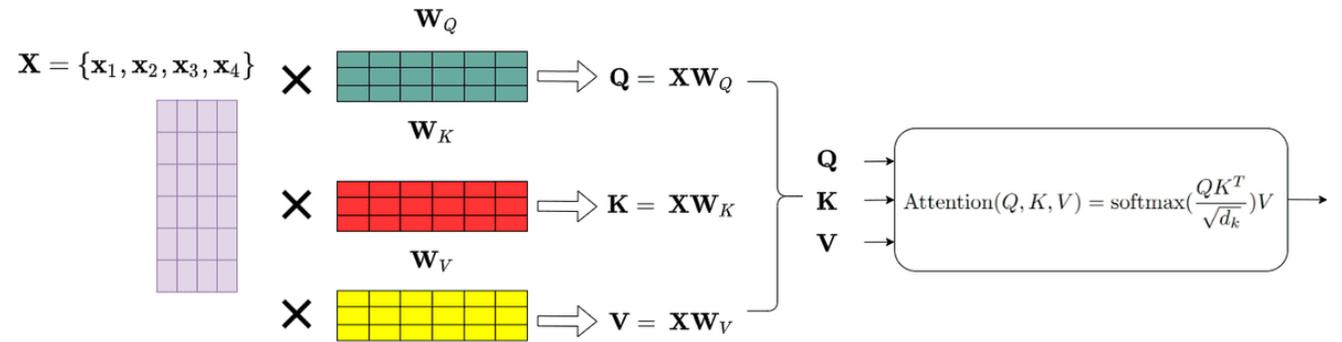
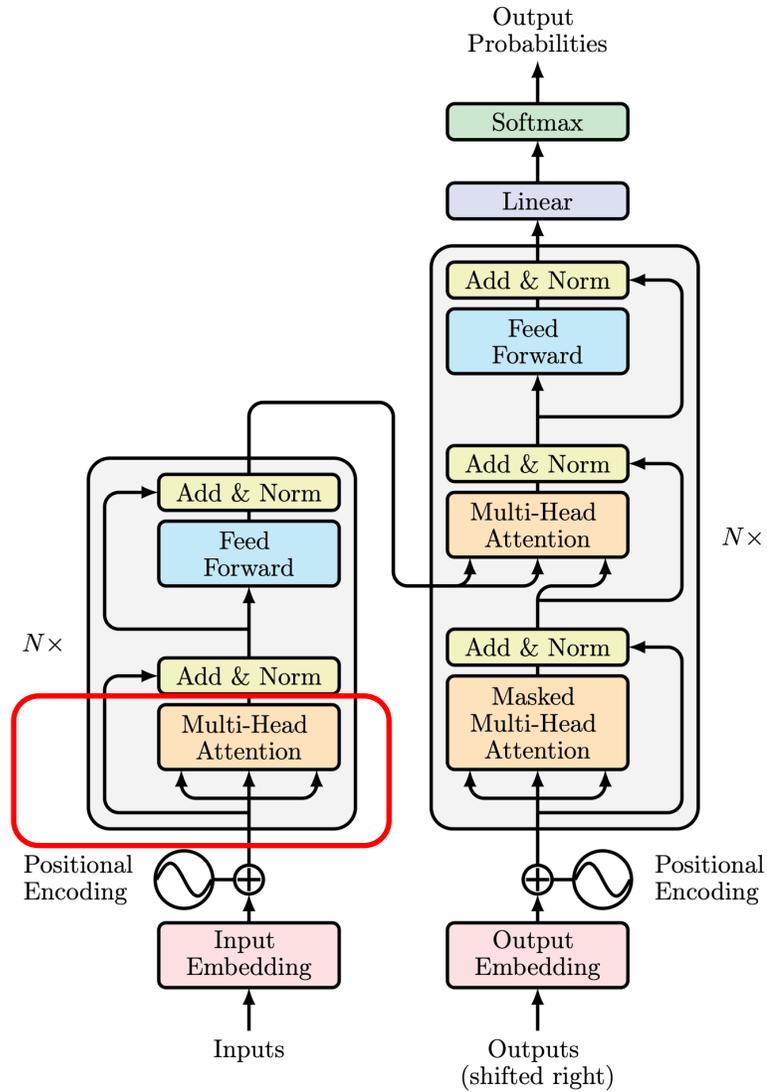
Encoder



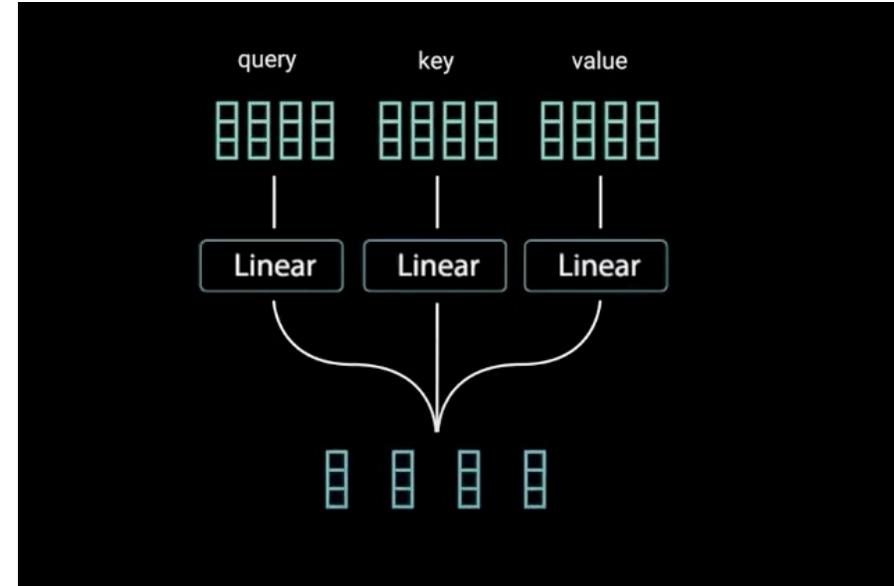
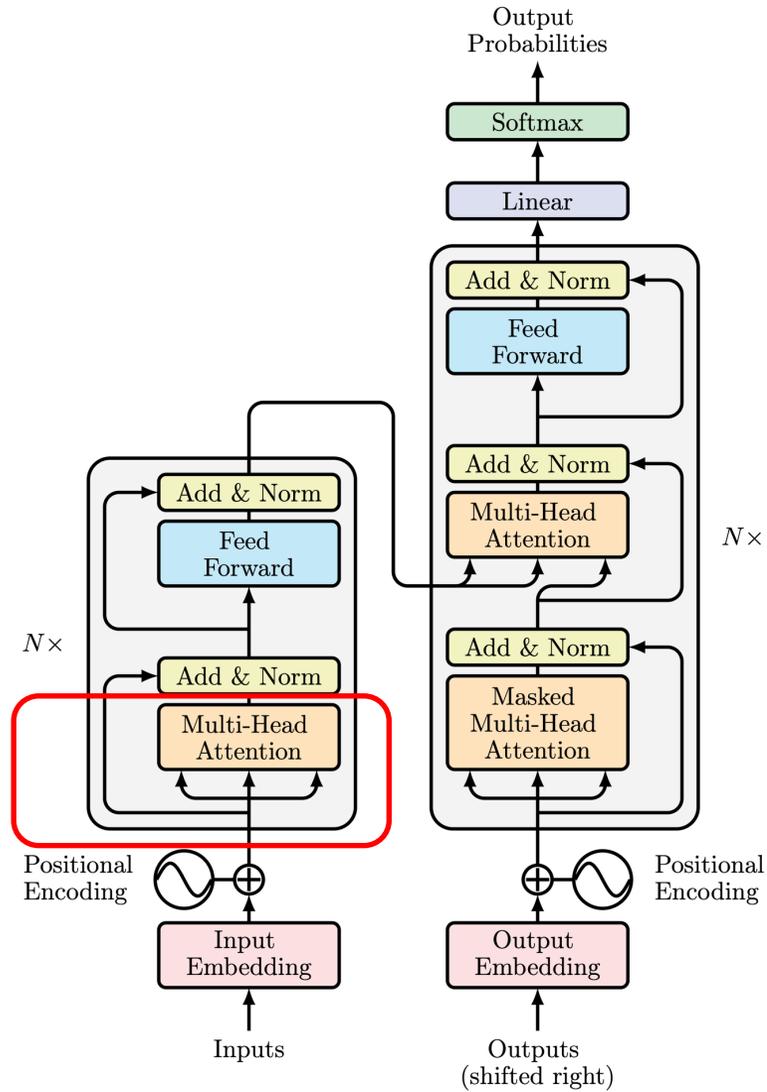
Outline

- Applications and Challenges of Sequence Modeling
- Transformers
 - Positional Encoding
 - Encoder
 - **Multi-head Self-Attention**
 - Decoder
- Limitations & Variants
 - Pre-norm vs. Post-norm
 - Vision Transformer
 - Swin Transformer

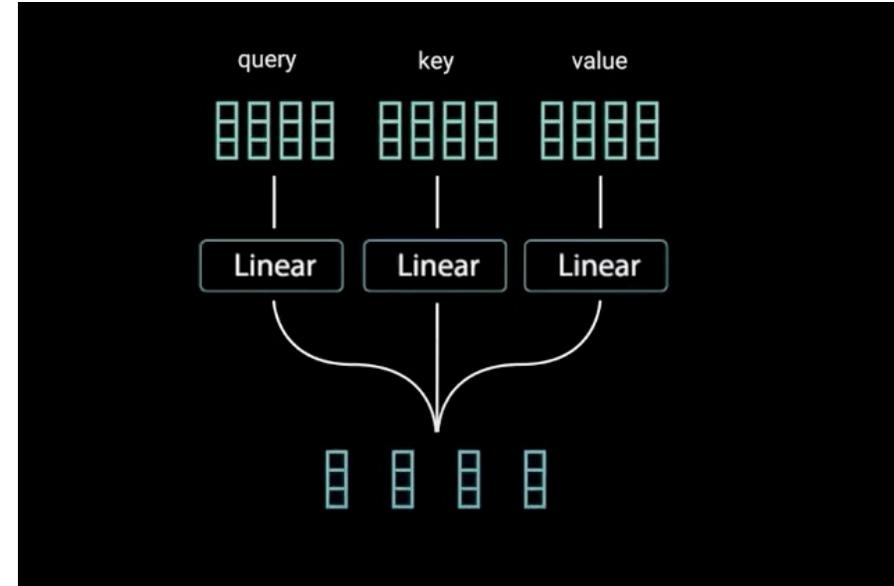
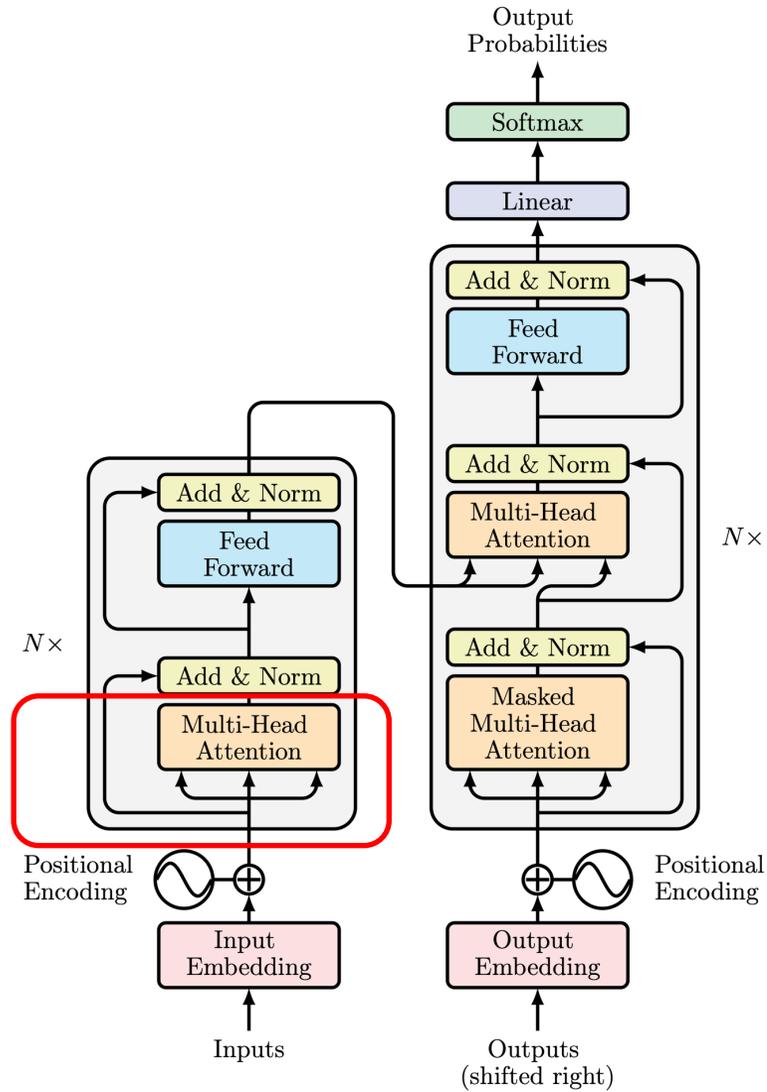
Multi-Head Attention



Multi-Head Attention

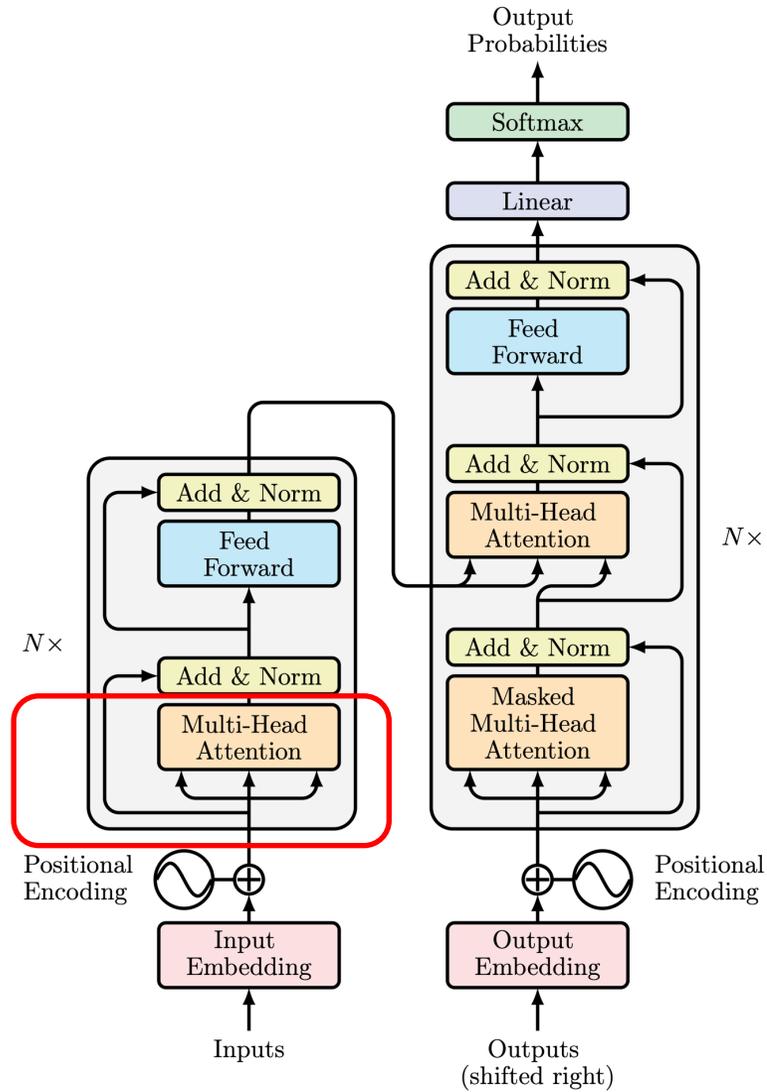


Multi-Head Attention

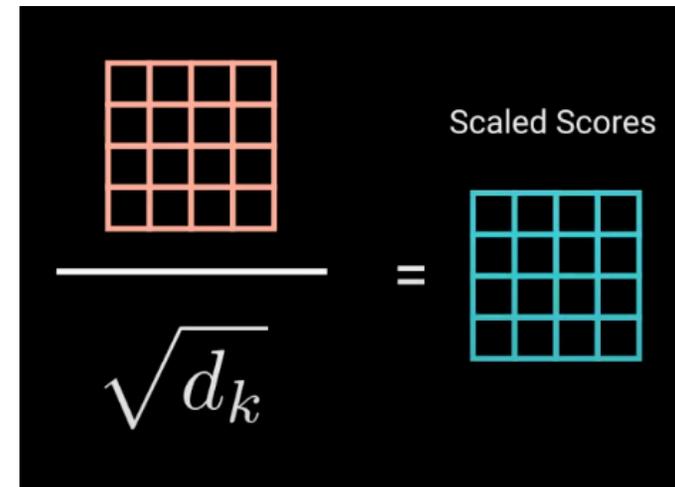


| | Hi | how | are | you |
|-----|----|-----|-----|-----|
| Hi | 98 | 27 | 10 | 12 |
| how | 27 | 89 | 31 | 67 |
| are | 10 | 31 | 91 | 54 |
| you | 12 | 67 | 54 | 92 |

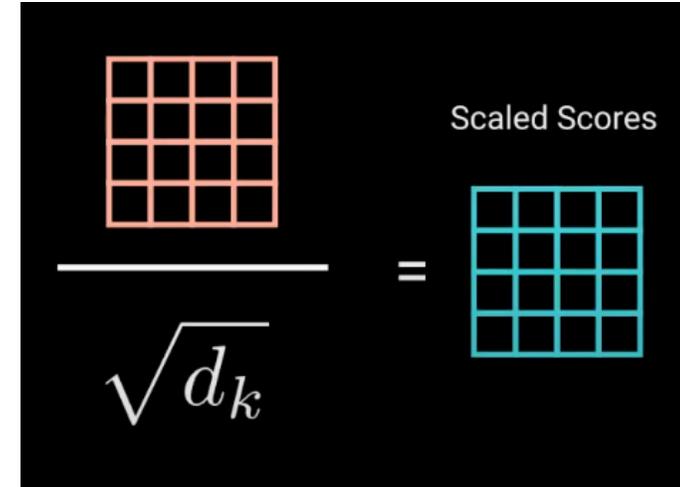
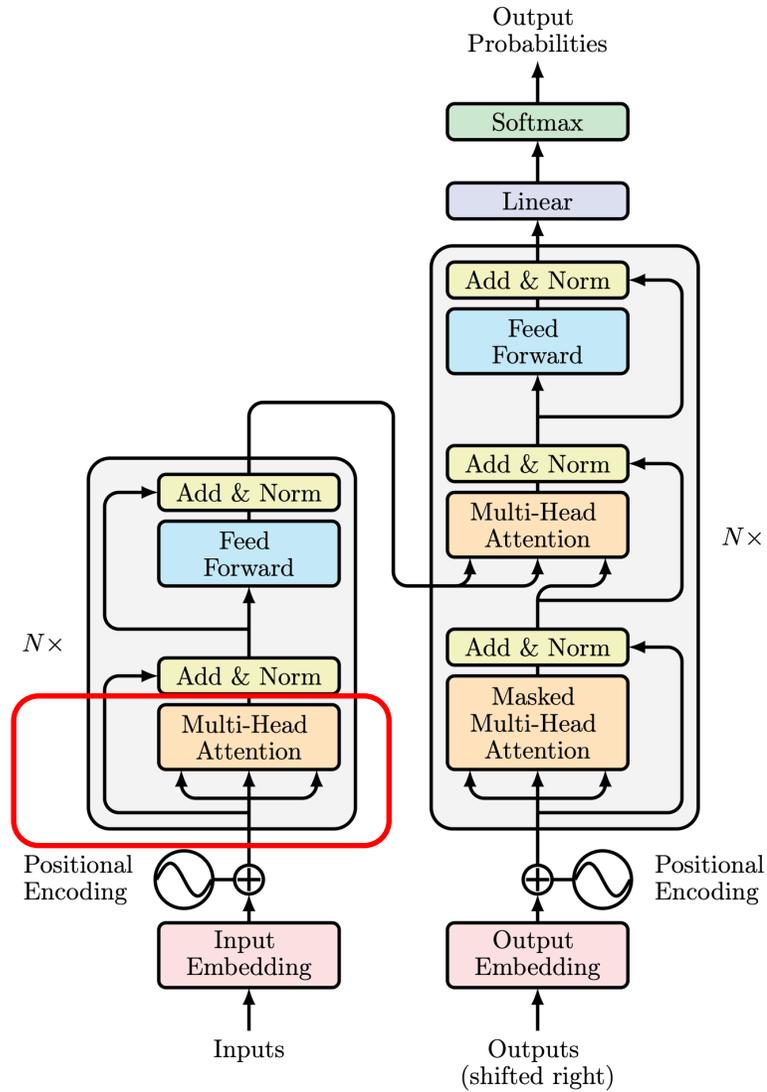
Multi-Head Attention



| | | | | |
|-----|----|-----|-----|-----|
| | Hi | how | are | you |
| Hi | 98 | 27 | 10 | 12 |
| how | 27 | 89 | 31 | 67 |
| are | 10 | 31 | 91 | 54 |
| you | 12 | 67 | 54 | 92 |



Multi-Head Attention

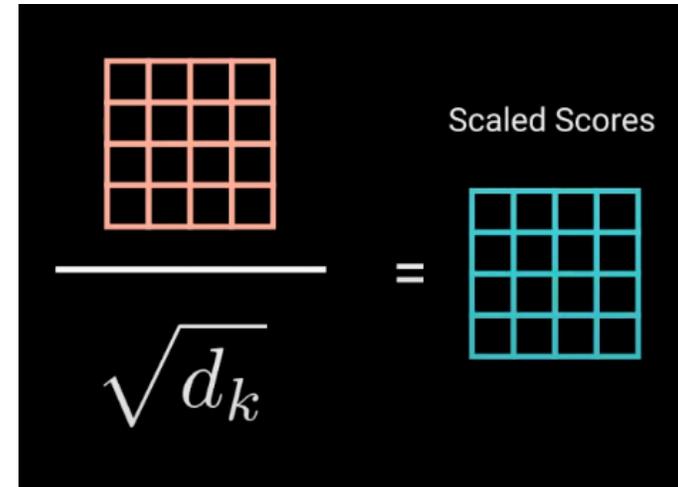
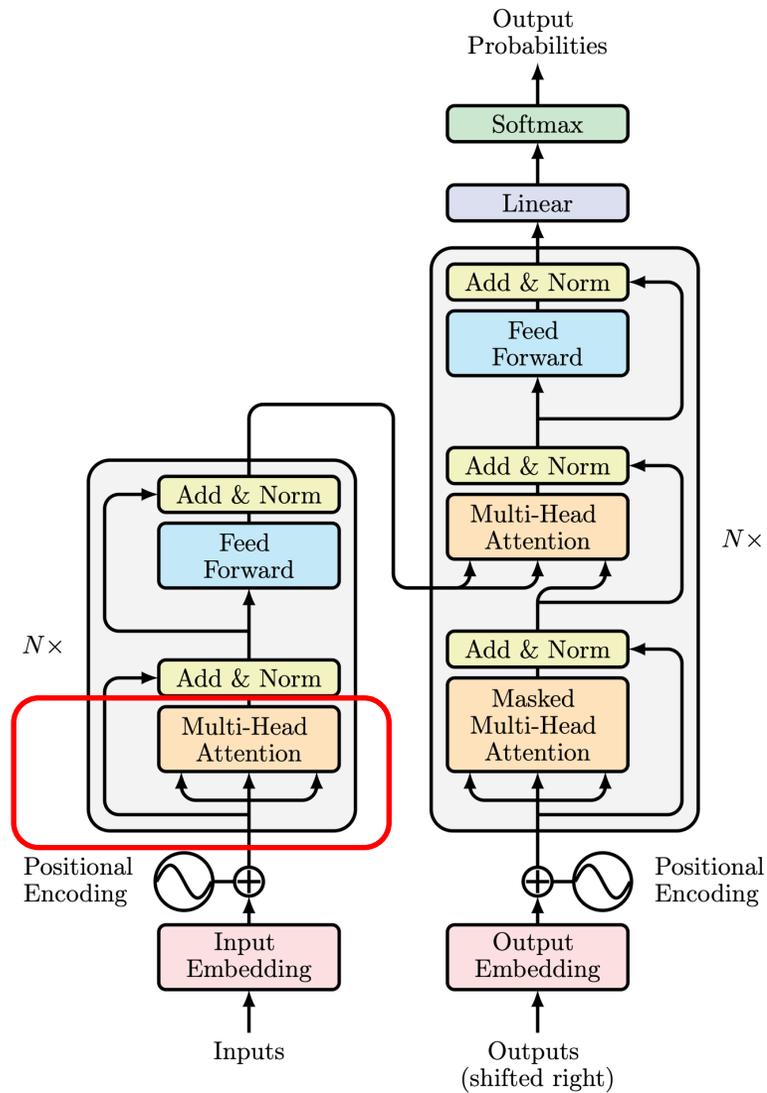


$\text{Softmax}(\begin{bmatrix} \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot \end{bmatrix}) =$

| | Hi | how | are | you |
|-----|-----|-----|-----|-----|
| Hi | 0.7 | 0.1 | 0.1 | 0.1 |
| how | 0.1 | 0.6 | 0.2 | 0.1 |
| are | 0.1 | 0.3 | 0.6 | 0 |
| you | 0.1 | 0.3 | 0.3 | 0.3 |

$$\text{softmax}(x)_i = \frac{\exp(x_i)}{\sum_j \exp(x_j)}$$

Multi-Head Attention



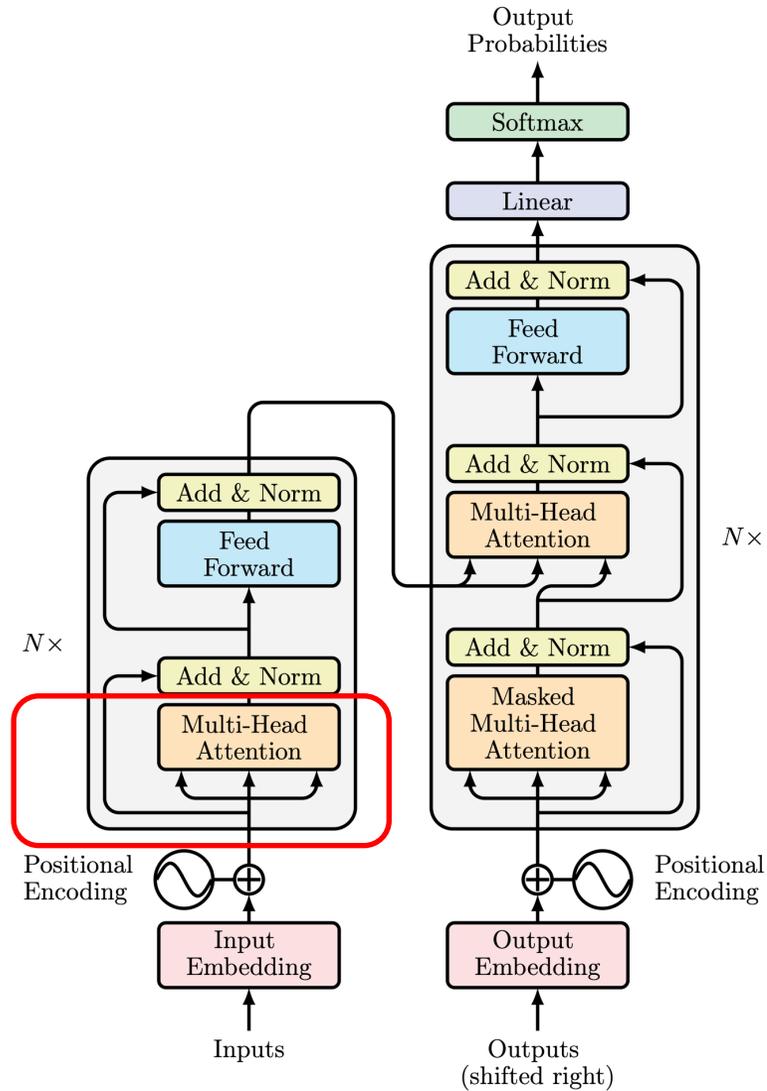
Why square root?

$\text{Softmax}(\begin{bmatrix} \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot \end{bmatrix}) =$

| | Hi | how | are | you |
|-----|-----|-----|-----|-----|
| Hi | 0.7 | 0.1 | 0.1 | 0.1 |
| how | 0.1 | 0.6 | 0.2 | 0.1 |
| are | 0.1 | 0.3 | 0.6 | 0 |
| you | 0.1 | 0.3 | 0.3 | 0.3 |

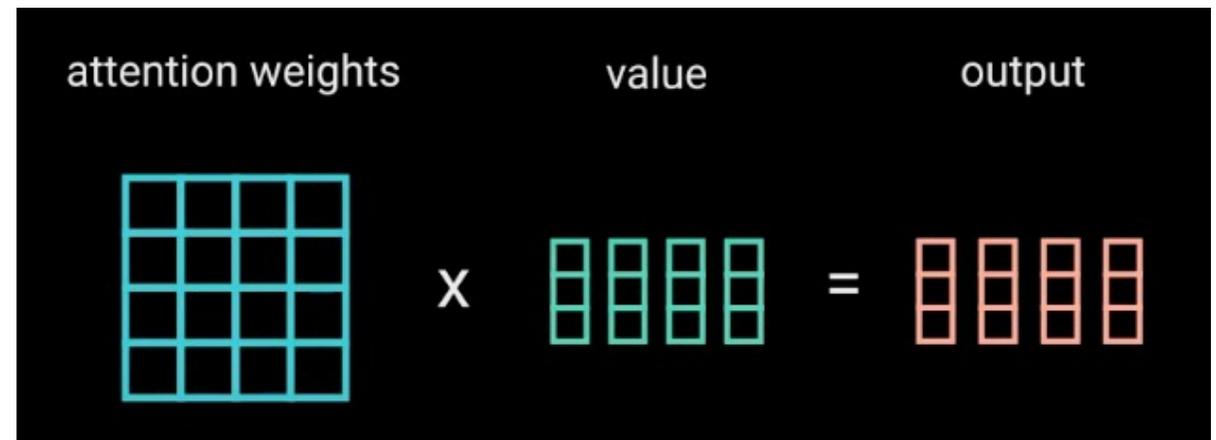
$$\text{softmax}(x)_i = \frac{\exp(x_i)}{\sum_j \exp(x_j)}$$

Multi-Head Attention

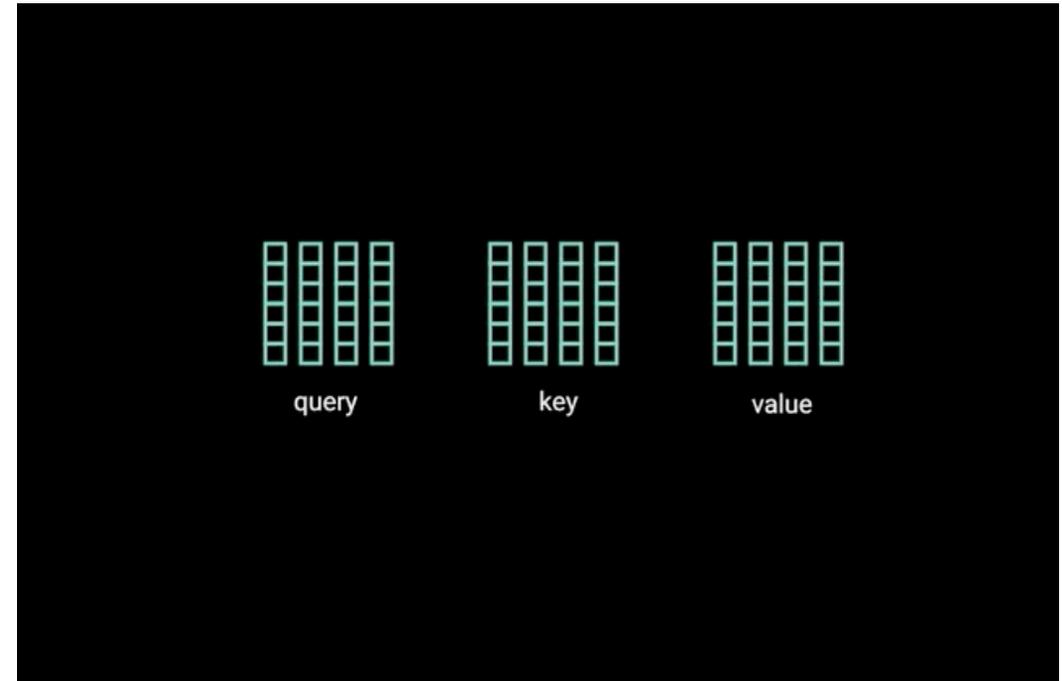
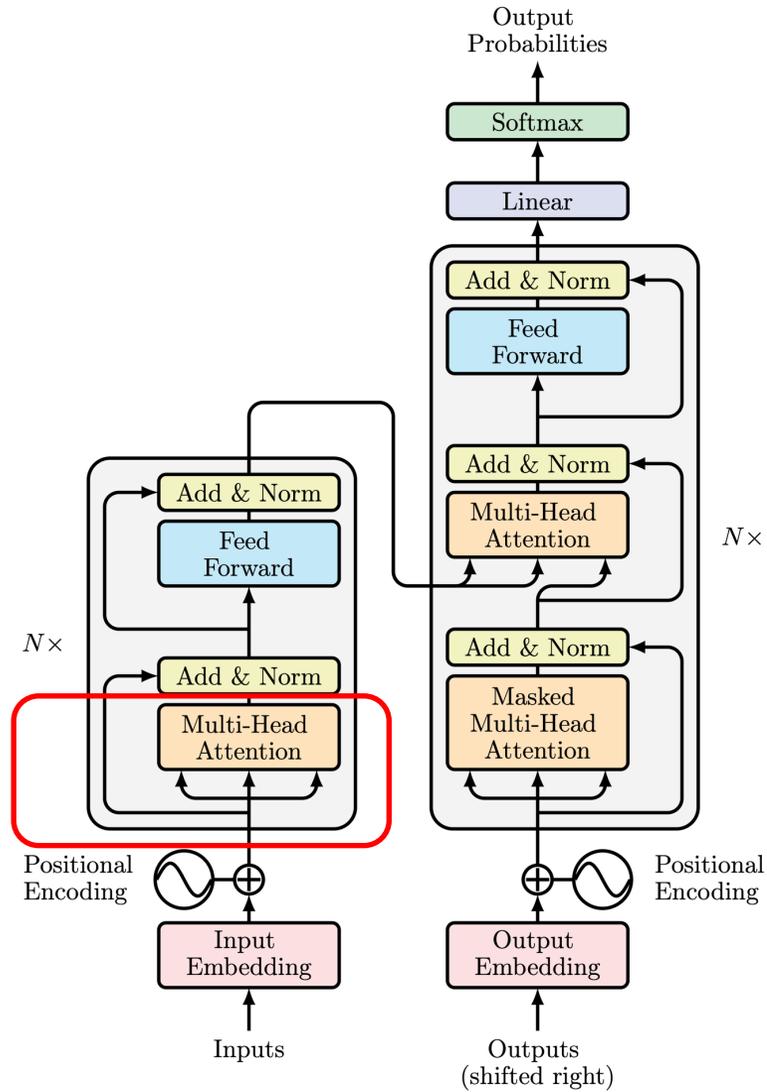


$\text{Softmax}(\begin{bmatrix} & & & & \\ & & & & \\ & & & & \\ & & & & \\ & & & & \end{bmatrix}) =$

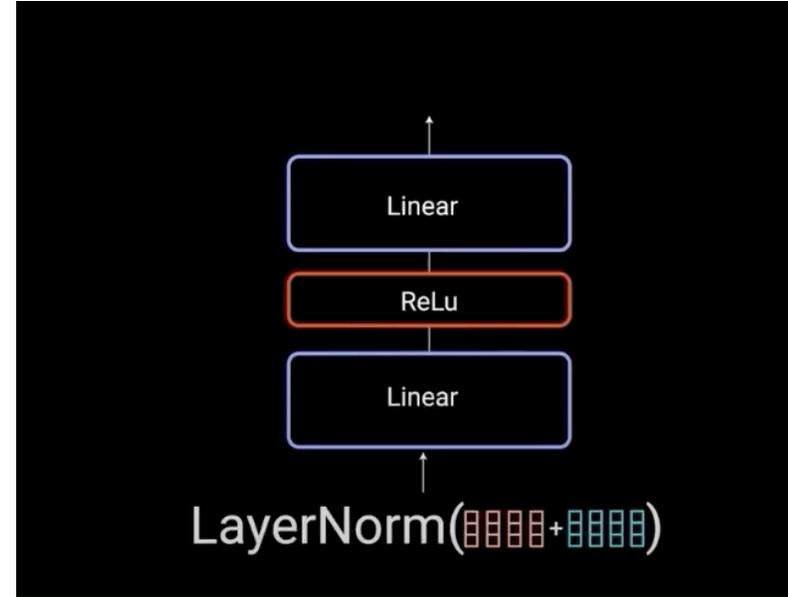
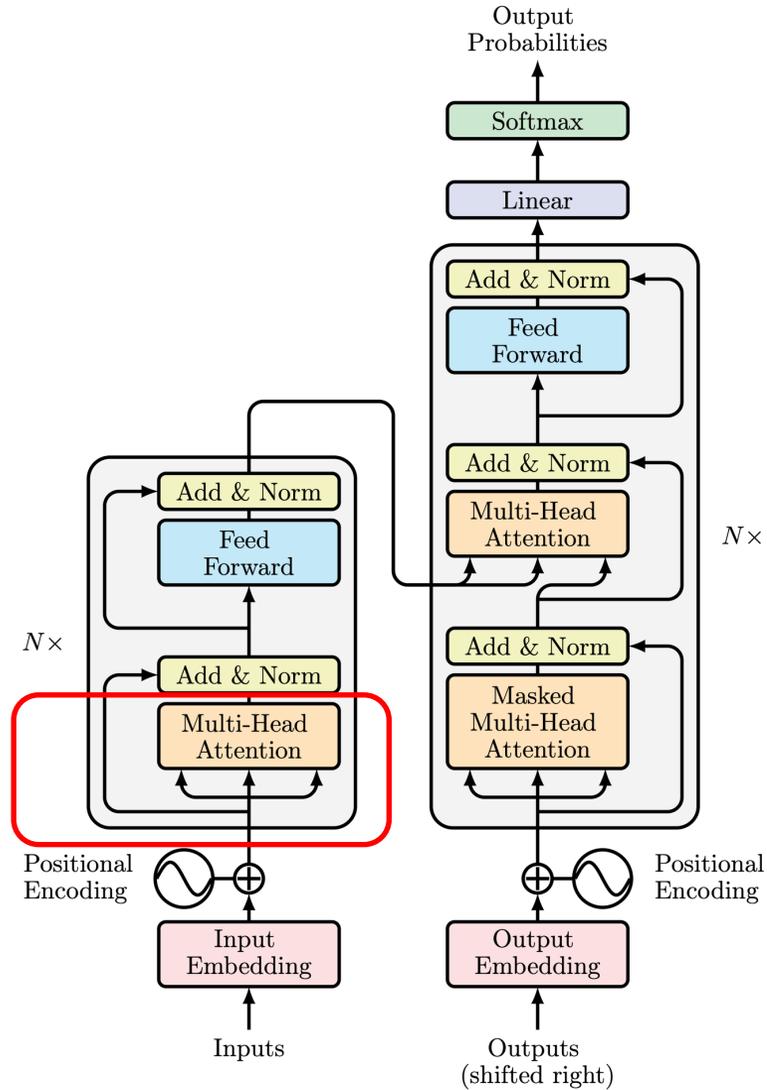
| | Hi | how | are | you |
|-----|-----|-----|-----|-----|
| Hi | 0.7 | 0.1 | 0.1 | 0.1 |
| how | 0.1 | 0.6 | 0.2 | 0.1 |
| are | 0.1 | 0.3 | 0.6 | 0 |
| you | 0.1 | 0.3 | 0.3 | 0.3 |

$$\text{softmax}(x)_i = \frac{\exp(x_i)}{\sum_j \exp(x_j)}$$


Multi-Head Attention



Layer Norm & Residual Connection



$$\mu_i = \frac{1}{K} \sum_{k=1}^K x_{i,k}$$

$$\sigma_i^2 = \frac{1}{K} \sum_{k=1}^K (x_{i,k} - \mu_i)^2$$

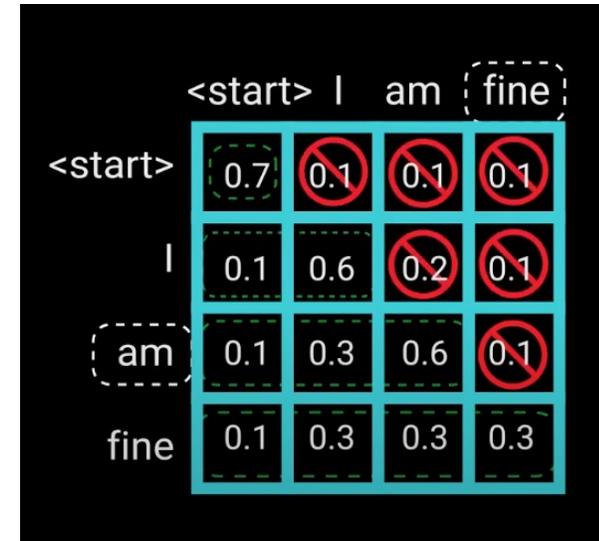
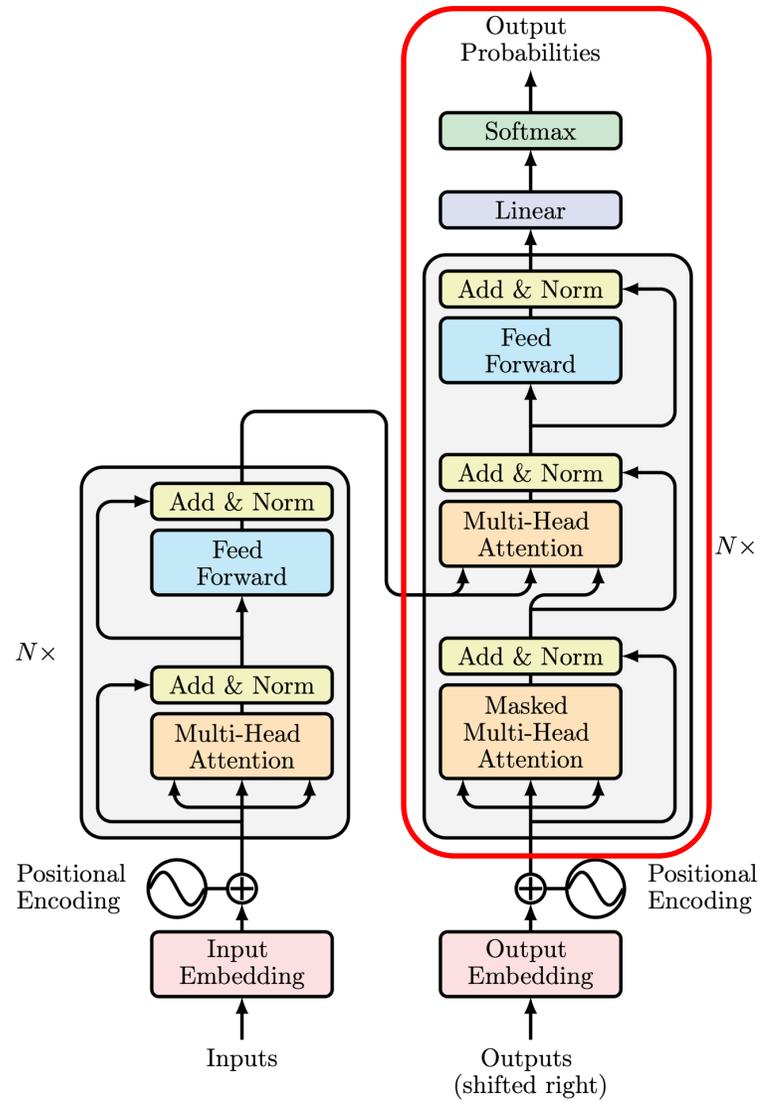
$$\hat{x}_{i,k} = \frac{x_{i,k} - \mu_i}{\sqrt{\sigma_i^2 + \epsilon}}$$

$$y_i = \gamma \hat{x}_i + \beta \equiv \text{LN}_{\gamma, \beta}(x_i)$$

Outline

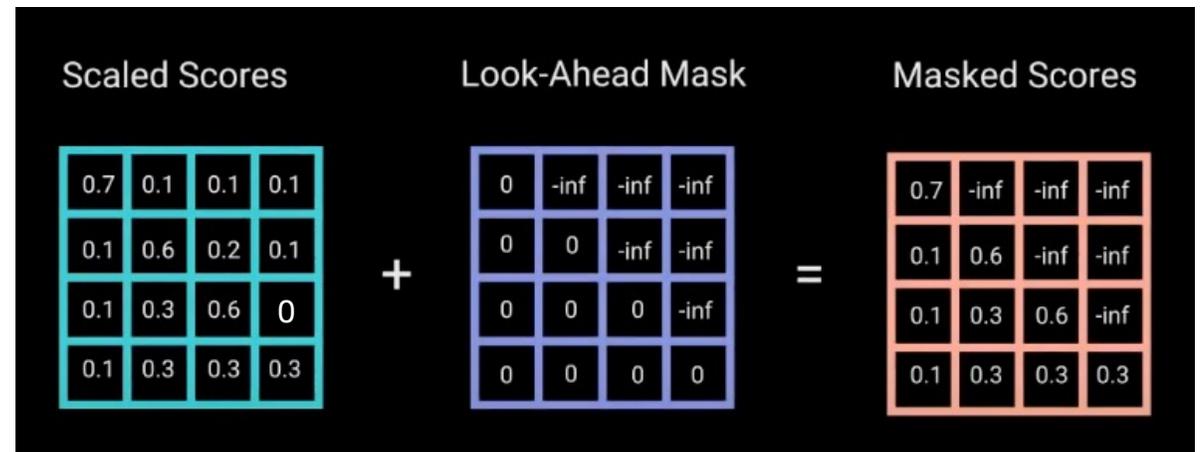
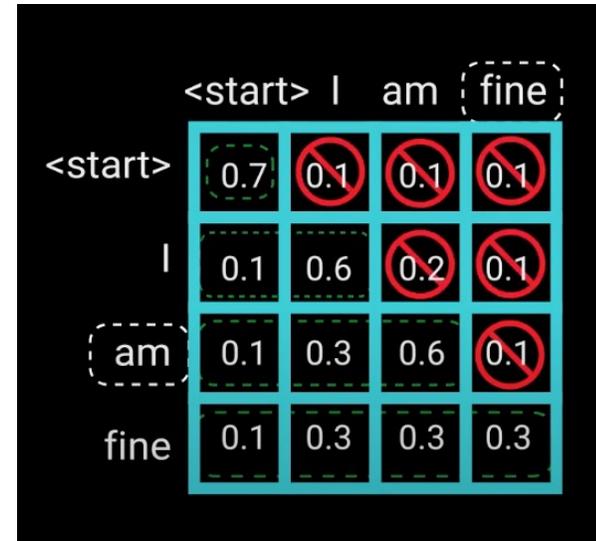
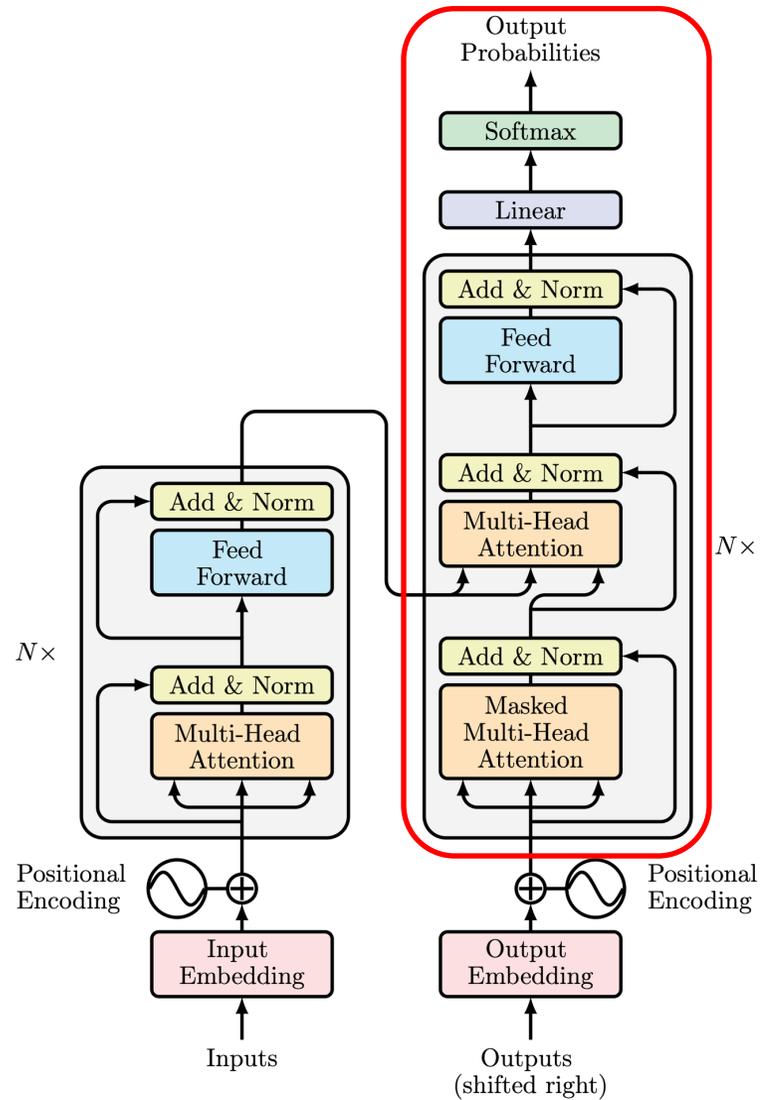
- Applications and Challenges of Sequence Modeling
- Transformers
 - Positional Encoding
 - Encoder
 - Multi-head Self-Attention
 - **Decoder**
- Limitations & Variants
 - Pre-norm vs. Post-norm
 - Vision Transformer
 - Swin Transformer

Masked Multi-Head Attention

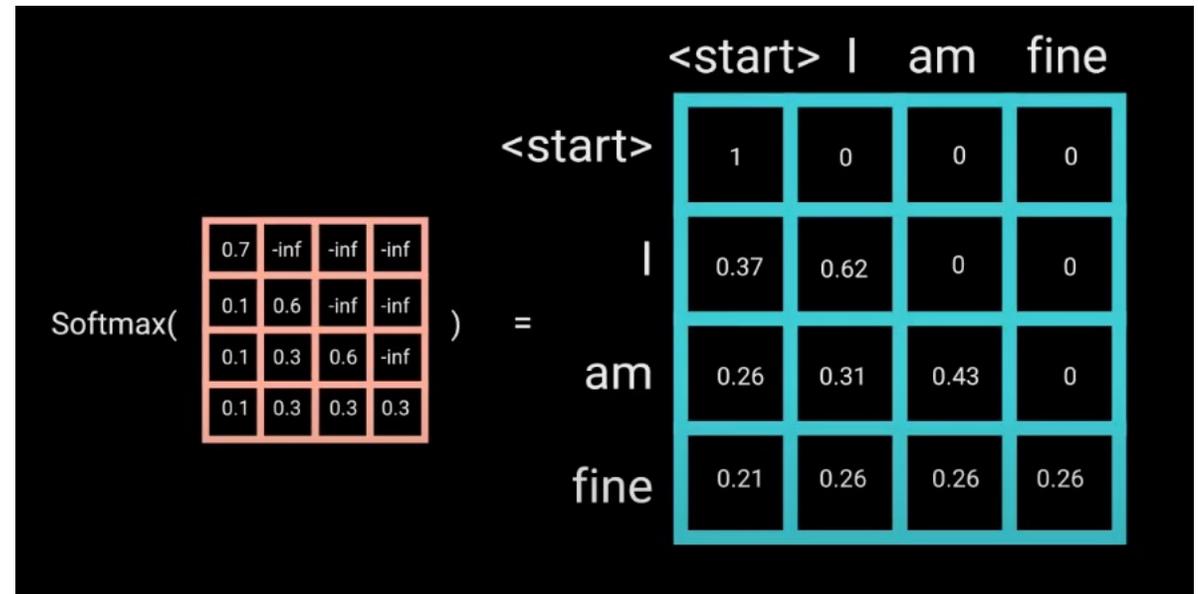
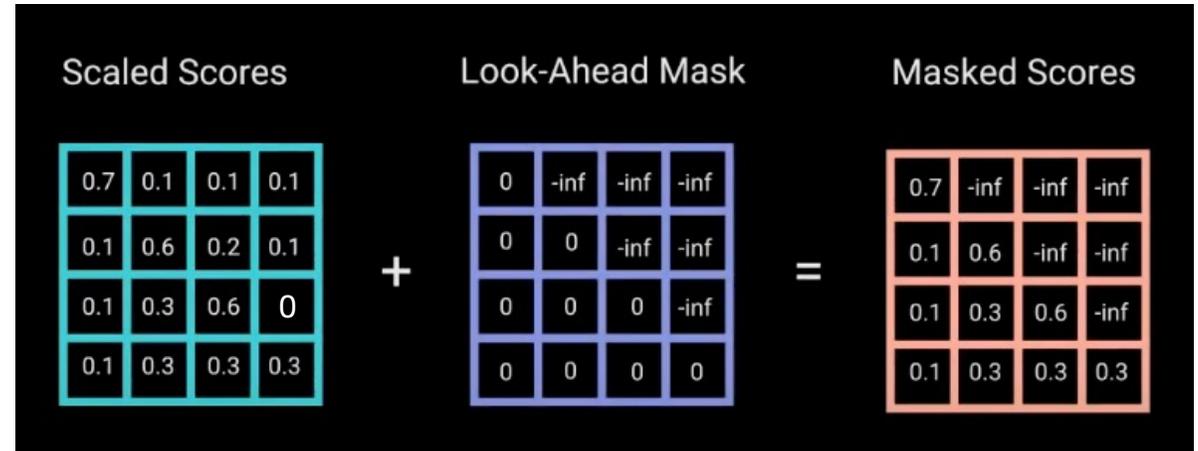
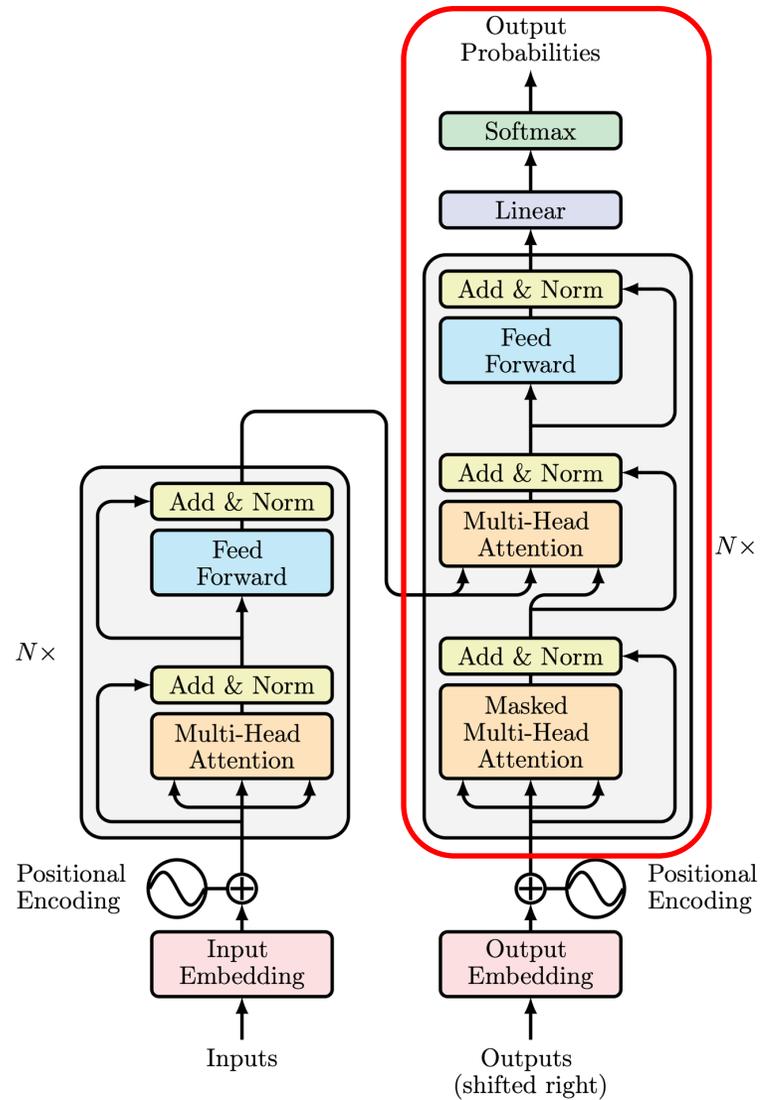


Prevent attending from future!

Masked Multi-Head Attention



Masked Multi-Head Attention



Hugging Face Demos

<https://transformer.huggingface.co/>



Write With Transformer

Get a modern neural network to
auto-complete your thoughts.

This web app, built by the Hugging Face team, is the official demo of the `🤗/transformers` repository's text generation capabilities.



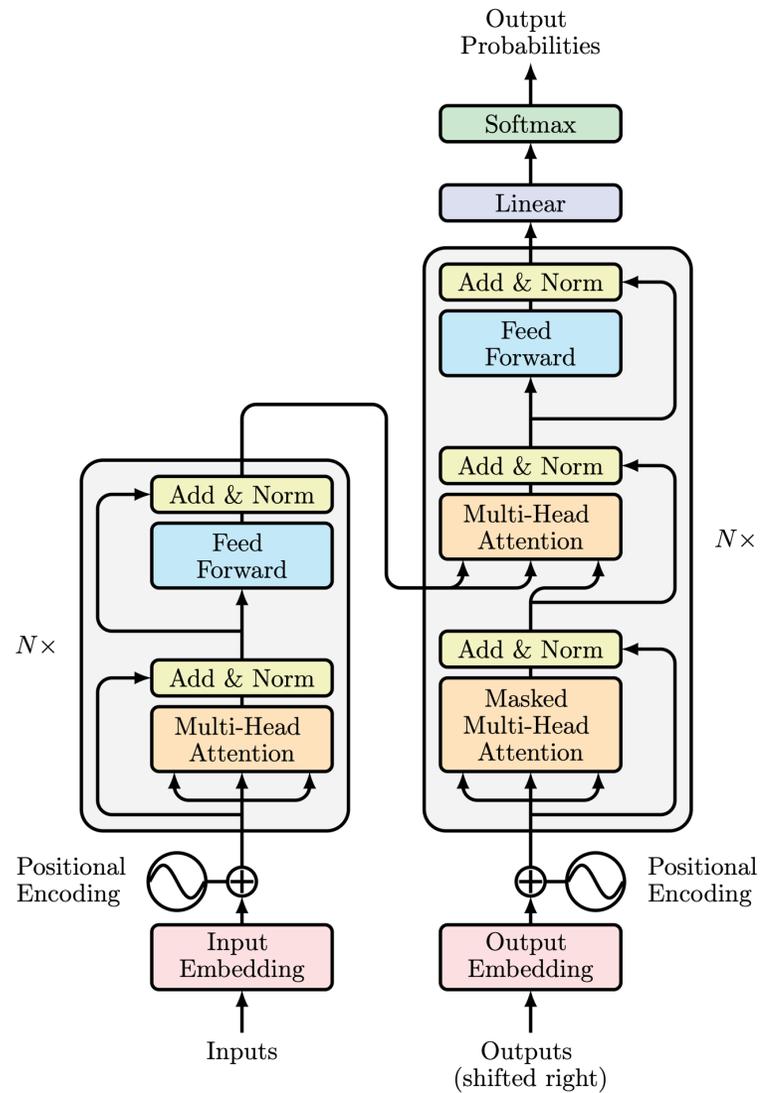
Star

57,016

Outline

- Applications and Challenges of Sequence Modeling
- Transformers
 - Positional Encoding
 - Encoder
 - Multi-head Self-Attention
 - Decoder
- **Limitations & Variants**
 - Pre-norm vs. Post-norm
 - Vision Transformer
 - Swin Transformer

Limitations



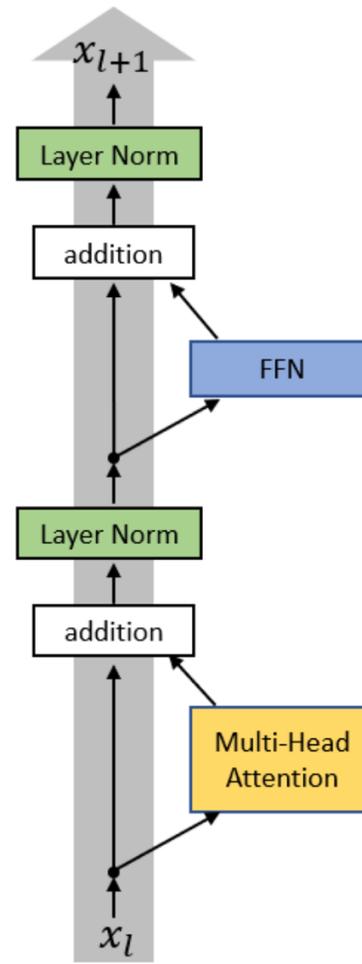
- $O(L^2)$ time/memory cost for self-attention
- How can we incorporate prior knowledge into attention rather than having a fully connected attention?
 - Encourage sparse attention
 - Inject known graph structures
 -

Outline

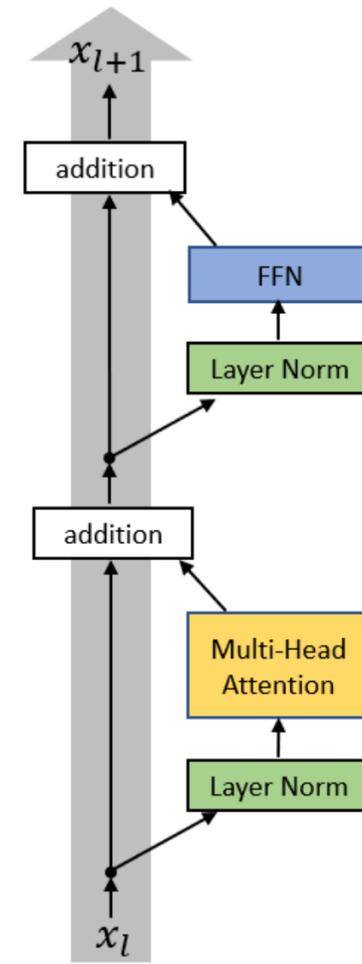
- Applications and Challenges of Sequence Modeling
- Transformers
 - Positional Encoding
 - Encoder
 - Multi-head Self-Attention
 - Decoder
- Limitations & Variants
 - **Pre-norm vs. Post-norm**
 - Vision Transformer
 - Swin Transformer

Pre-Norm vs. Post-Norm

Where to place the Layer Normalization?



Post-Norm

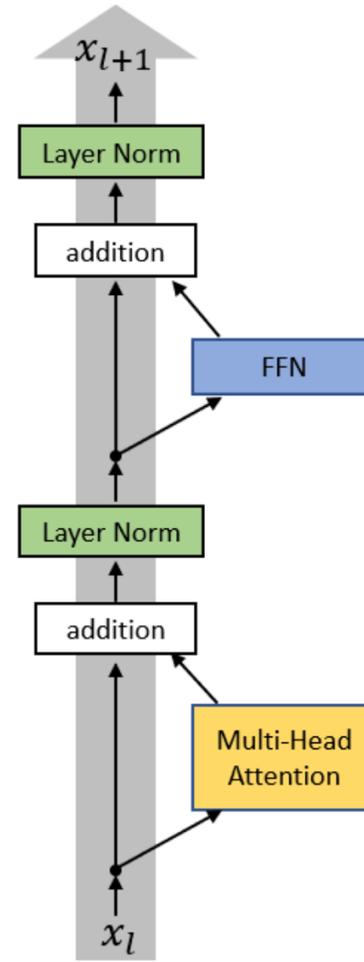


Pre-Norm

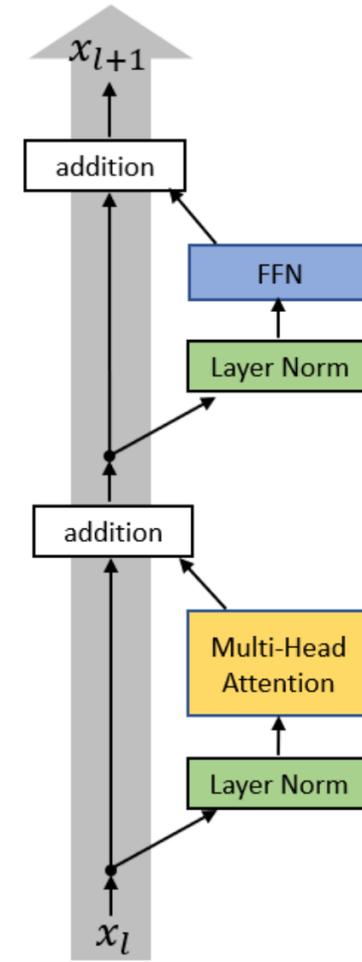
Pre-Norm vs. Post-Norm

Where to place the Layer Normalization?

- Gradient norm in the Post-Norm Transformer is large for parameters near the output and will be likely to decay as the layer gets closer to input



Post-Norm

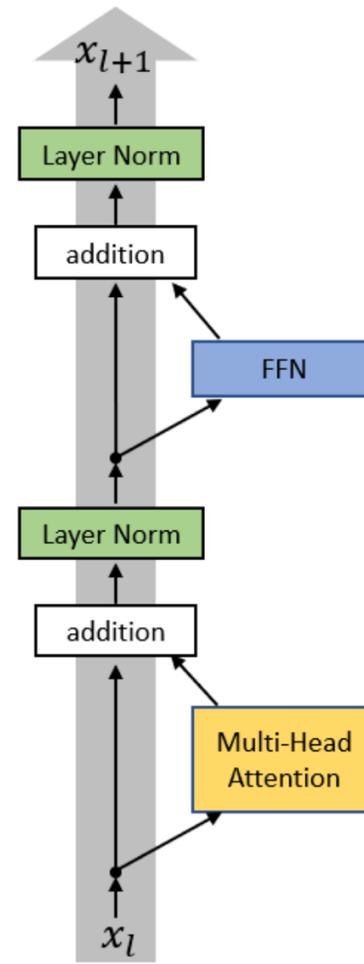


Pre-Norm

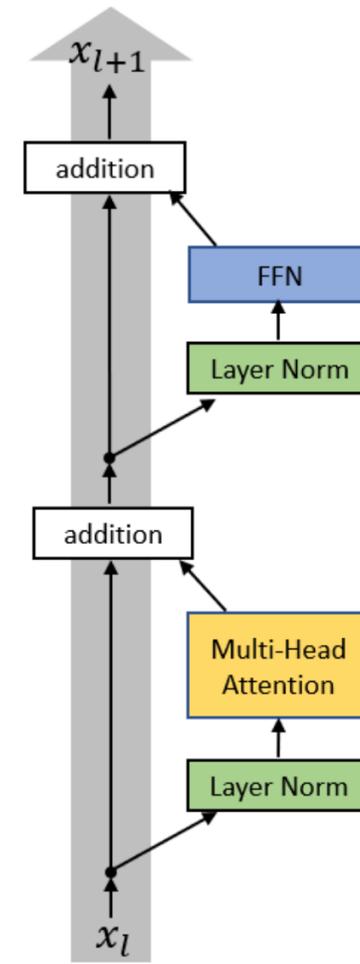
Pre-Norm vs. Post-Norm

Where to place the Layer Normalization?

- Gradient norm in the Post-Norm Transformer is large for parameters near the output and will be likely to decay as the layer gets closer to input
- Training the Pre-Norm Transformer does not rely on the learning rate warm-up stage and can be trained much faster than the Post-Norm



Post-Norm



Pre-Norm

Outline

- Applications and Challenges of Sequence Modeling
- Transformers
 - Positional Encoding
 - Encoder
 - Multi-head Self-Attention
 - Decoder
- Limitations & Variants
 - Pre-norm vs. Post-norm
 - **Vision Transformer**
 - Swin Transformer

Extensions: Vision Transformer



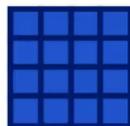
Outline

- Applications and Challenges of Sequence Modeling
- Transformers
 - Positional Encoding
 - Encoder
 - Multi-head Self-Attention
 - Decoder
- Limitations & Variants
 - Pre-norm vs. Post-norm
 - Vision Transformer
 - **Swin Transformer**

Extensions: Swin Transformer

Standard MSA

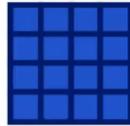
Attention for each patch is computed against all patches, resulting in quadratic complexity



Extensions: Swin Transformer

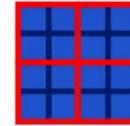
Standard MSA

Attention for each patch is computed against all patches, resulting in quadratic complexity



Window-based MSA

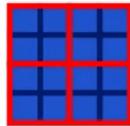
Attention for each patch is only computed within its own window (drawn in red). Window size is 2x2 in this example.



Extensions: Swin Transformer

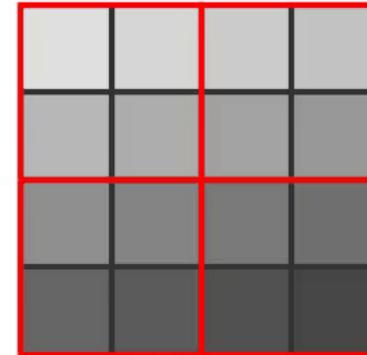
Window-based MSA

Attention for each patch is only computed within its own window (drawn in red).
Window size is 2x2 in this example.



Shifted Window MSA

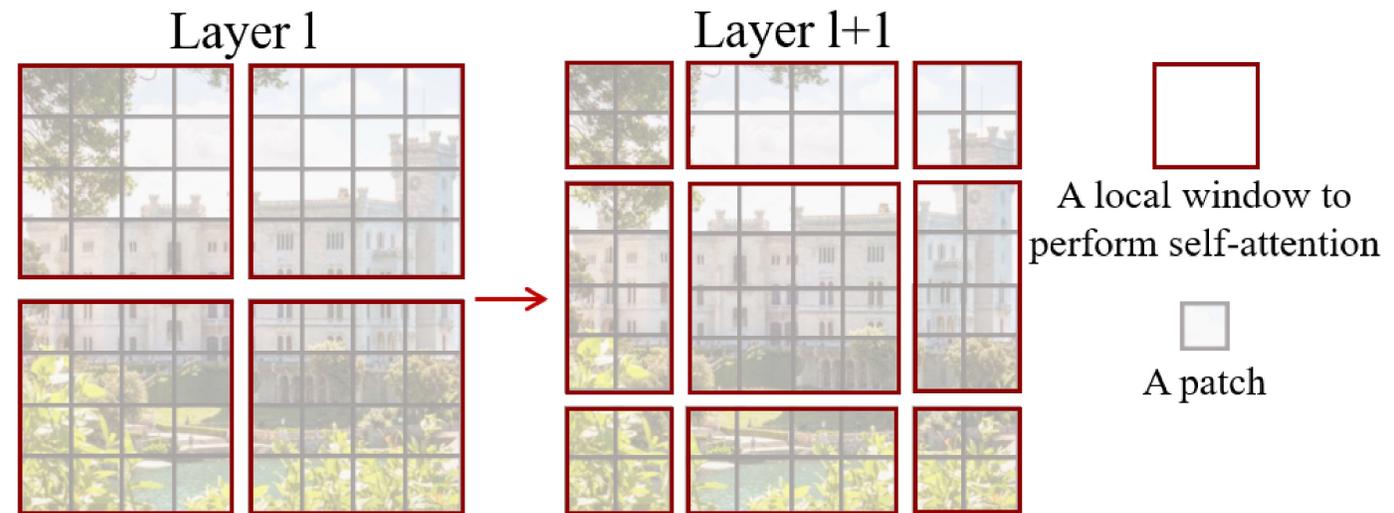
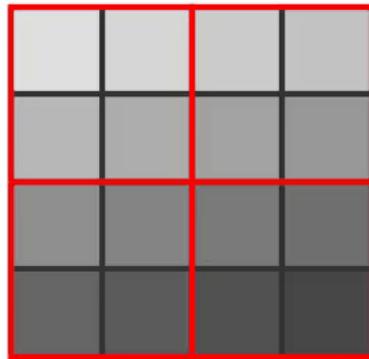
Step 1: Shift window by a factor of $M/2$, where M = window size
Step 2: For efficient batch computation, move patches into empty slots to create a complete window.
This is known as 'cyclic shift' in the paper.



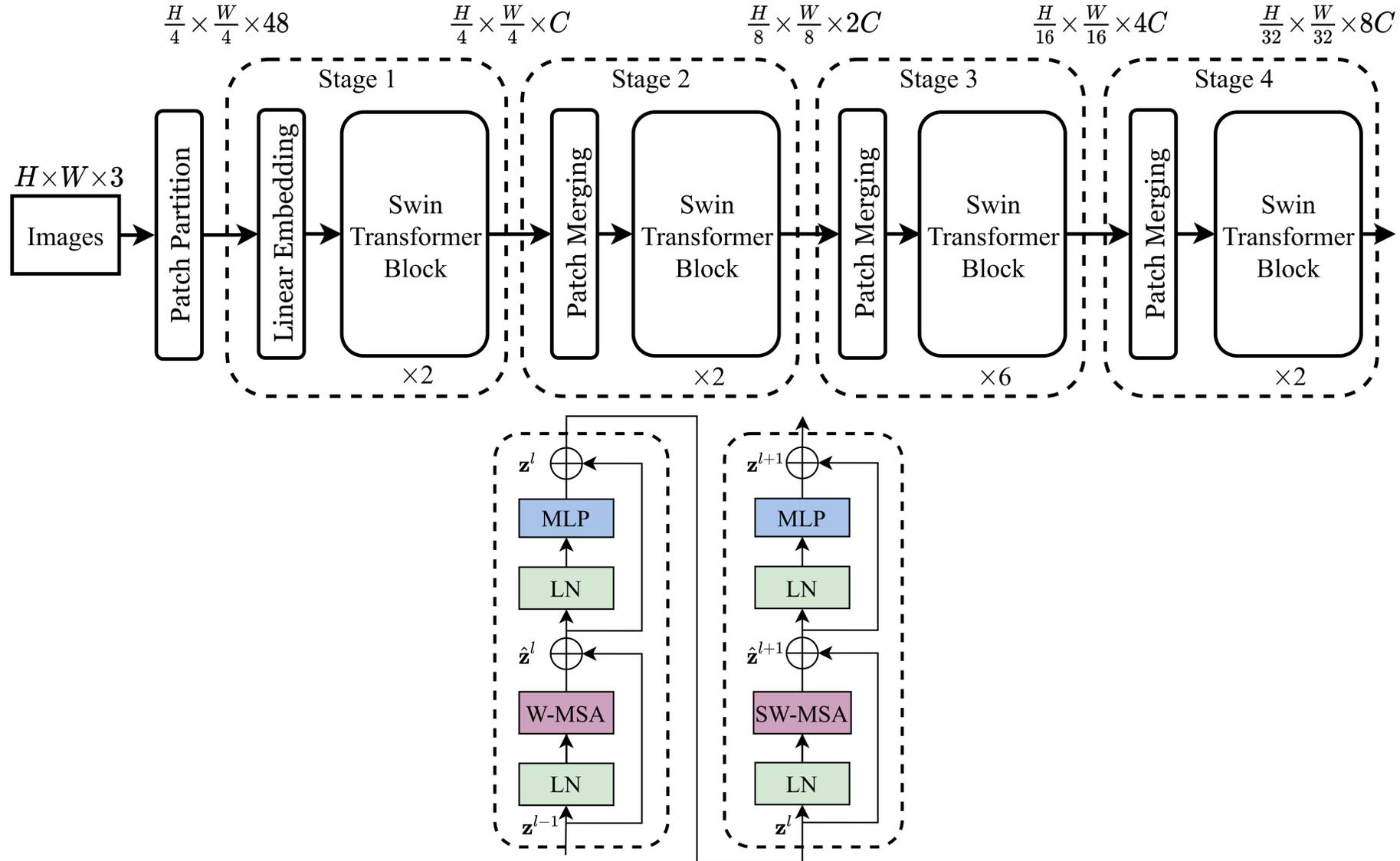
Extensions: Swin Transformer

Shifted Window MSA

Step 1: Shift window by a factor of $M/2$, where M = window size
Step 2: For efficient batch computation, move patches into empty slots to create a complete window.
This is known as 'cyclic shift' in the paper.



Extensions: Swin Transformer



References

- [1] <http://web.stanford.edu/class/cs224n/>
- [2] <https://jalammar.github.io/illustrated-transformer/>
- [3] <https://www.mrc-cbu.cam.ac.uk/people/matt.davis/cmabridge>
- [4] <https://towardsdatascience.com/illustrated-guide-to-transformers-step-by-step-explanation-f74876522bc0>
- [5] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, Ł. and Polosukhin, I., 2017. Attention is all you need. In Advances in neural information processing systems (pp. 5998-6008).
- [6] <https://jalammar.github.io/illustrated-transformer/>
- [7] <https://towardsdatascience.com/illustrated-guide-to-transformers-step-by-step-explanation-f74876522bc0>
- [8] <https://lilianweng.github.io/lil-log/2020/04/07/the-transformer-family.html>
- [9] <https://theaisummer.com/transformer/>
- [10] <https://transformer.huggingface.co/>
- [11] Xiong, R., Yang, Y., He, D., Zheng, K., Zheng, S., Xing, C., Zhang, H., Lan, Y., Wang, L. and Liu, T., 2020, November. On layer normalization in the transformer architecture. In International Conference on Machine Learning (pp. 10524-10533). PMLR.
- [12] Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S. and Uszkoreit, J., 2020. An image is worth 16x16 words: Transformers for image recognition at scale. arXiv preprint arXiv:2010.11929.

References

[13] <https://github.com/lucidrains/vit-pytorch>

[14] <https://towardsdatascience.com/a-comprehensive-guide-to-swin-transformer-64965f89d14c>

[15] Liu, Z., Lin, Y., Cao, Y., Hu, H., Wei, Y., Zhang, Z., Lin, S. and Guo, B., 2021. Swin transformer: Hierarchical vision transformer using shifted windows. In Proceedings of the IEEE/CVF International Conference on Computer Vision (pp. 10012-10022).

Questions?