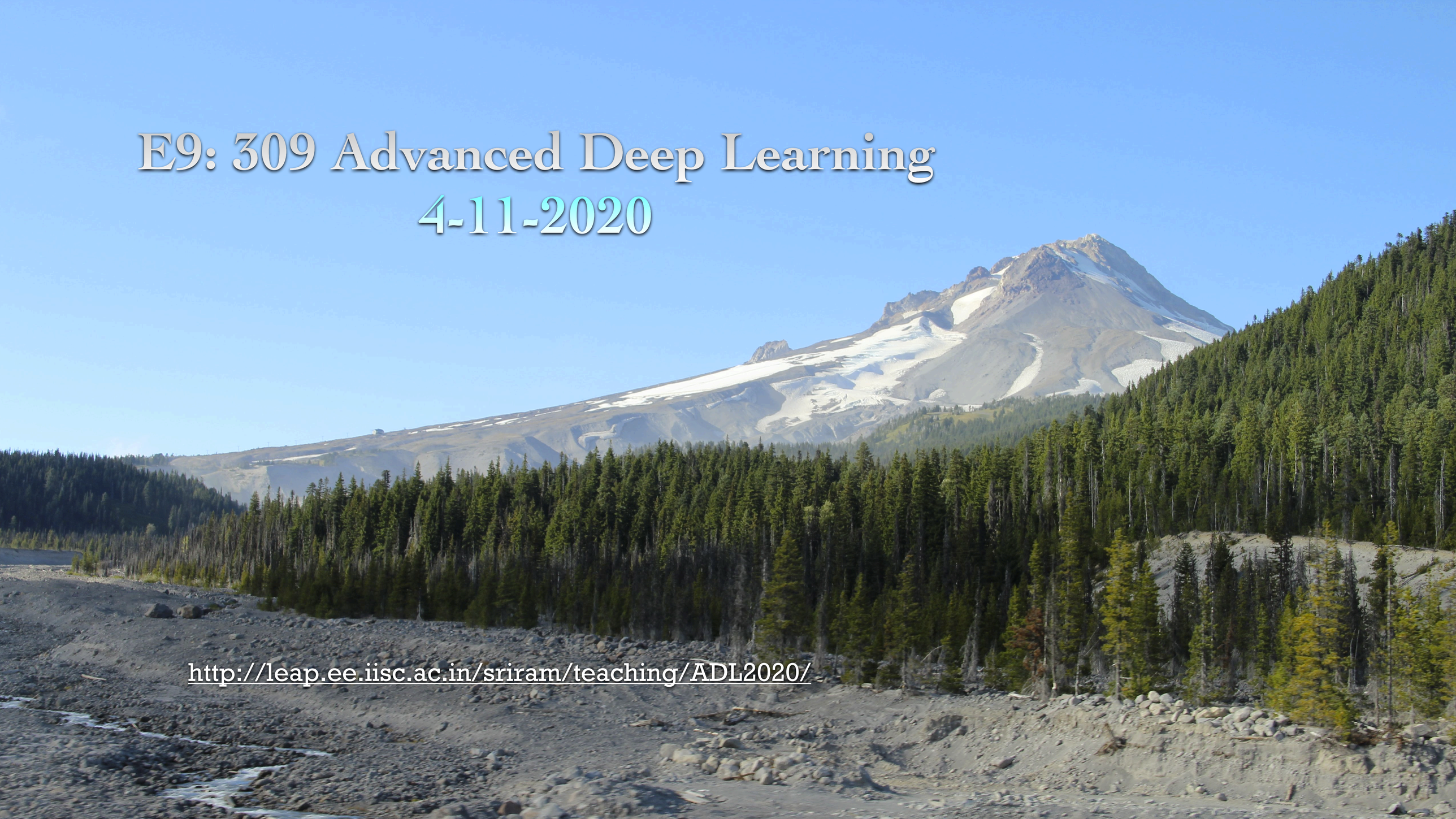


# E9: 309 Advanced Deep Learning

## 4-11-2020

http://leap.ee.iisc.ac.in/sriram/teaching/ADL2020/

# Housekeeping

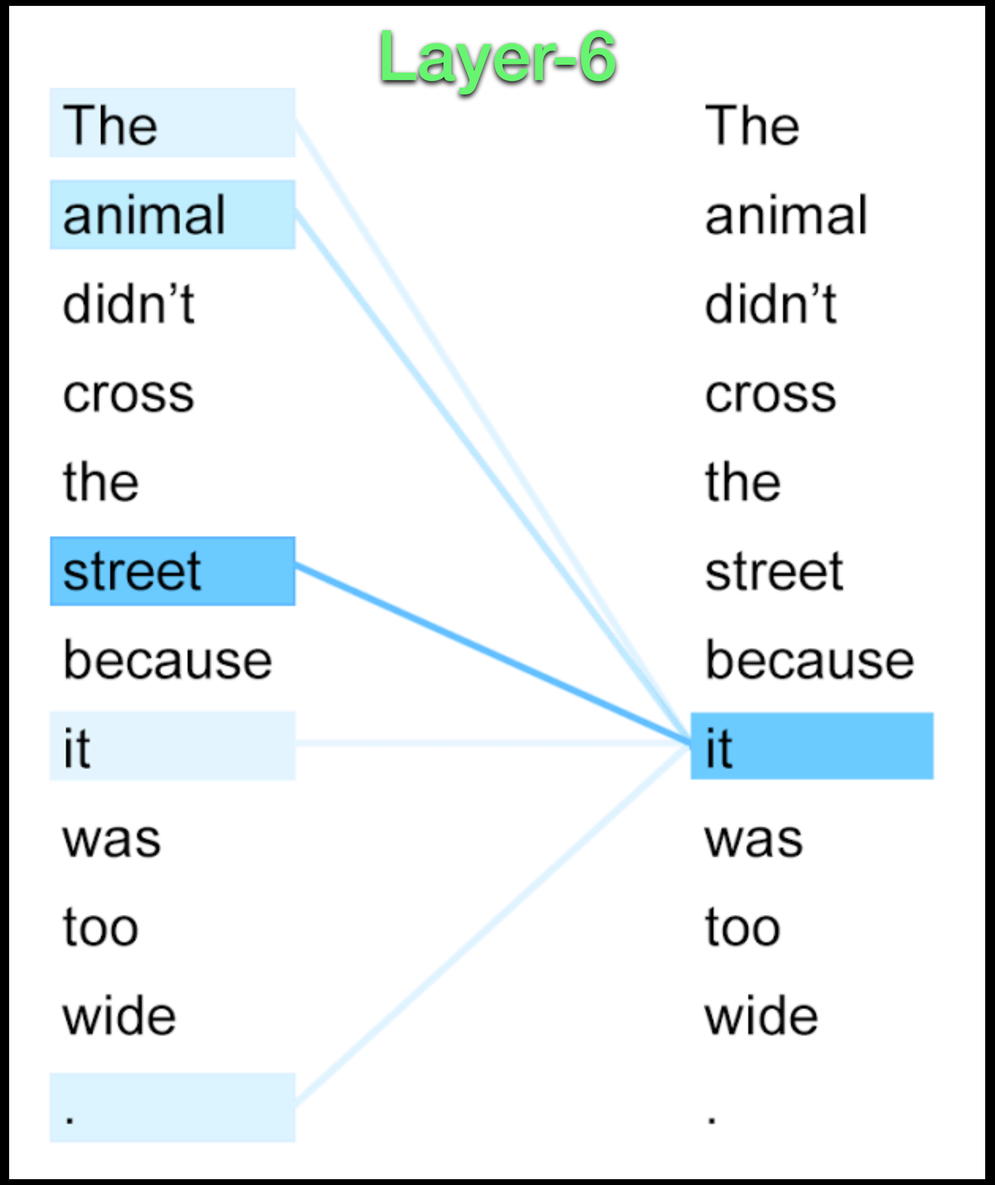
- 1st mini-project
  - ✓ Deadlines
    - ★ Abstract submission deadline (Nov 2nd, Monday)
      - ★ Using the google form given in the webpage
    - ★ Solo projects or 2-member projects
      - ★ Indicate roles of each member in 2-member project
      - ★ 200 page abstract of the work. If modifications are needed, we will review and let you know in 2-3 days.

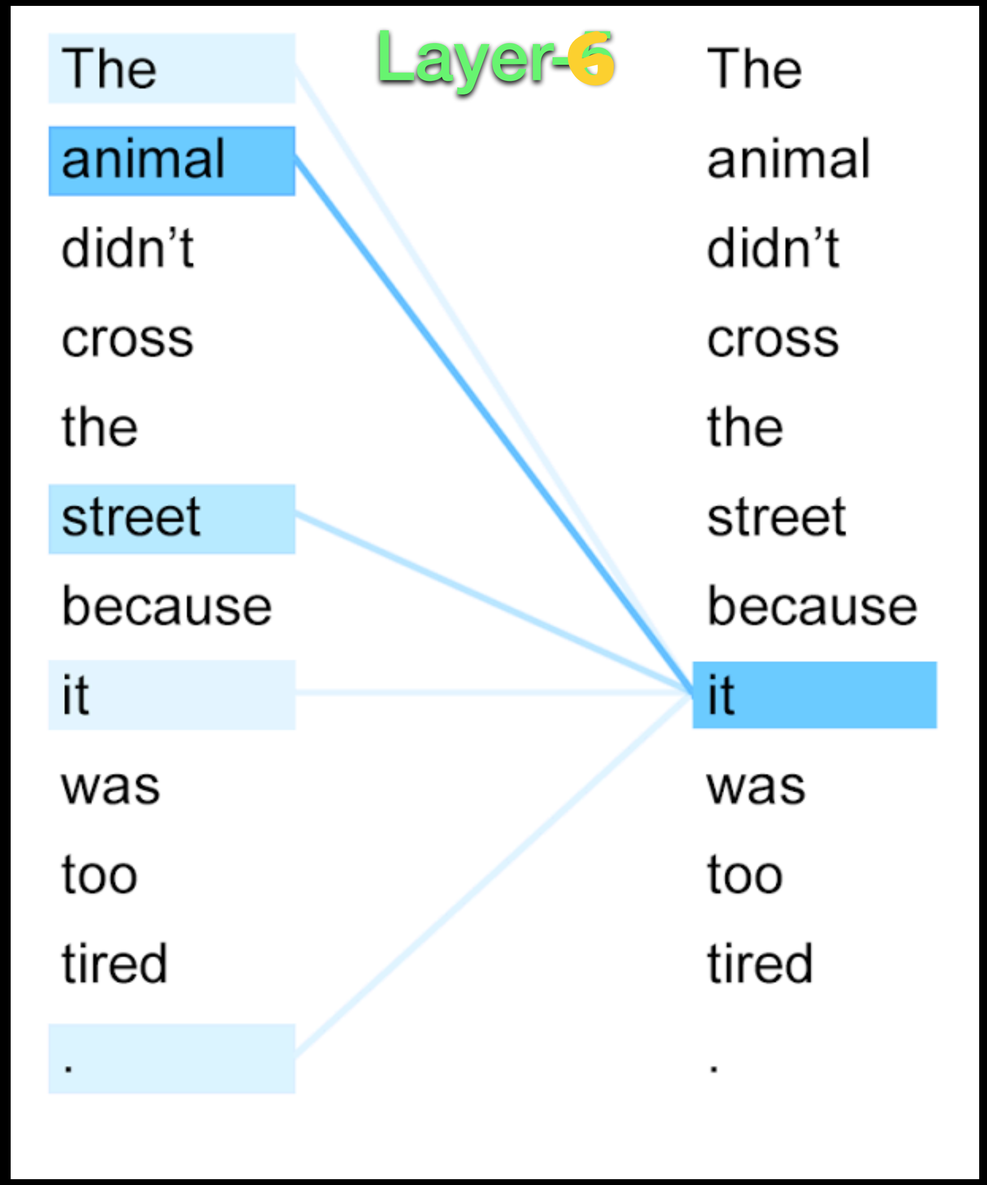
# Recap of previous class

# State of affairs

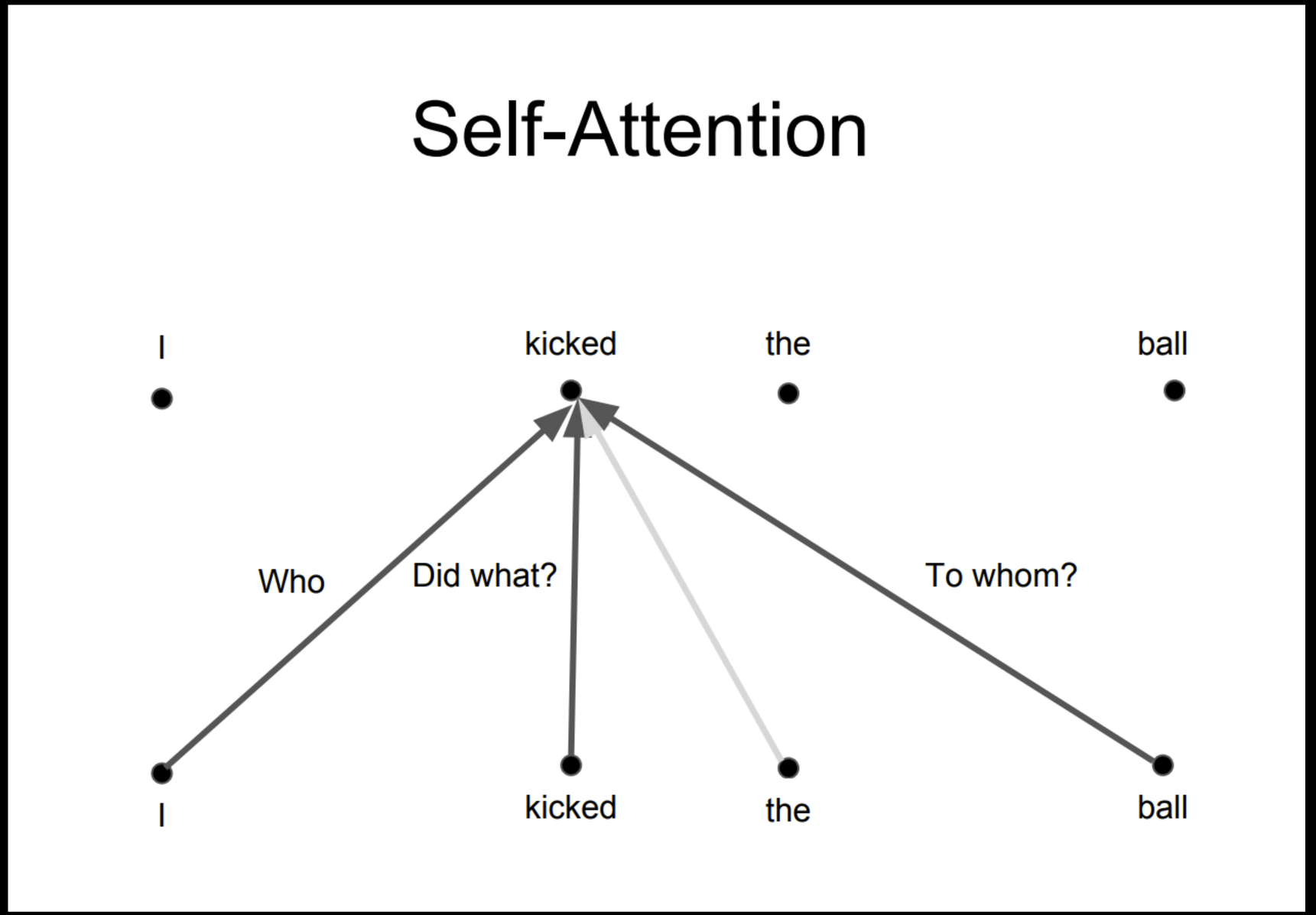
- Transformer models
  - ✓ Encoder-decoder model
    - ★ Repeated architecture in encoder and decoder stages
    - ★ Multi-head self-attention
    - ★ Position wise feed-forward.
    - ★ Skip connection with layer norm.

# Self-attention - need for depth



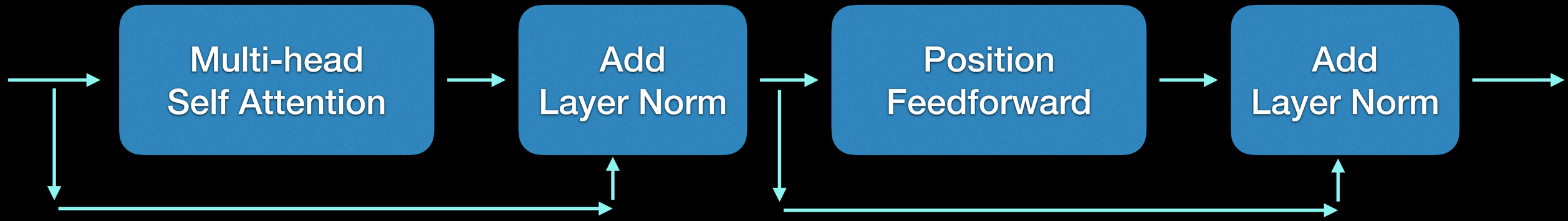



# Need for multi-head attention



# Single layer of encoder (typical implementation)

- Single encoder layer has typically self-attention skip connection, layer norm and feedforward layer



# Positional encoding

- No recurrence or position awareness yet in the model

Binary format -
position can encode the rate of change of bits across time

In floating format - one can use sines and cosines

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 0: | 0 | 0 | 0 | 0 | 8: | 1 | 0 | 0 | 0 |
| 1: | 0 | 0 | 0 | 1 | 9: | 1 | 0 | 0 | 1 |
| 2: | 0 | 0 | 1 | 0 | 10: | 1 | 0 | 1 | 0 |
| 3: | 0 | 0 | 1 | 1 | 11: | 1 | 0 | 1 | 1 |
| 4: | 0 | 1 | 0 | 0 | 12: | 1 | 1 | 0 | 0 |
| 5: | 0 | 1 | 0 | 1 | 13: | 1 | 1 | 0 | 1 |
| 6: | 0 | 1 | 1 | 0 | 14: | 1 | 1 | 1 | 0 |
| 7: | 0 | 1 | 1 | 1 | 15: | 1 | 1 | 1 | 1 |

# Positional encoding

- An example used in the first paper

$$\mathbf{p}(t) \in \mathcal{R}^D$$

$$p_i(t) = \begin{cases} sin(\omega_k t), & \text{if } i = 2k \\ cos(\omega_k t), & \text{if } i = 2k+1 \end{cases}$$

$$\omega_k = \frac{1}{10000^{\frac{2k}{D}}}$$

$i = 1 \ldots D$

$k \in \{1 \ldots \frac{D}{2}\}$

$t = 1 \ldots T$

model

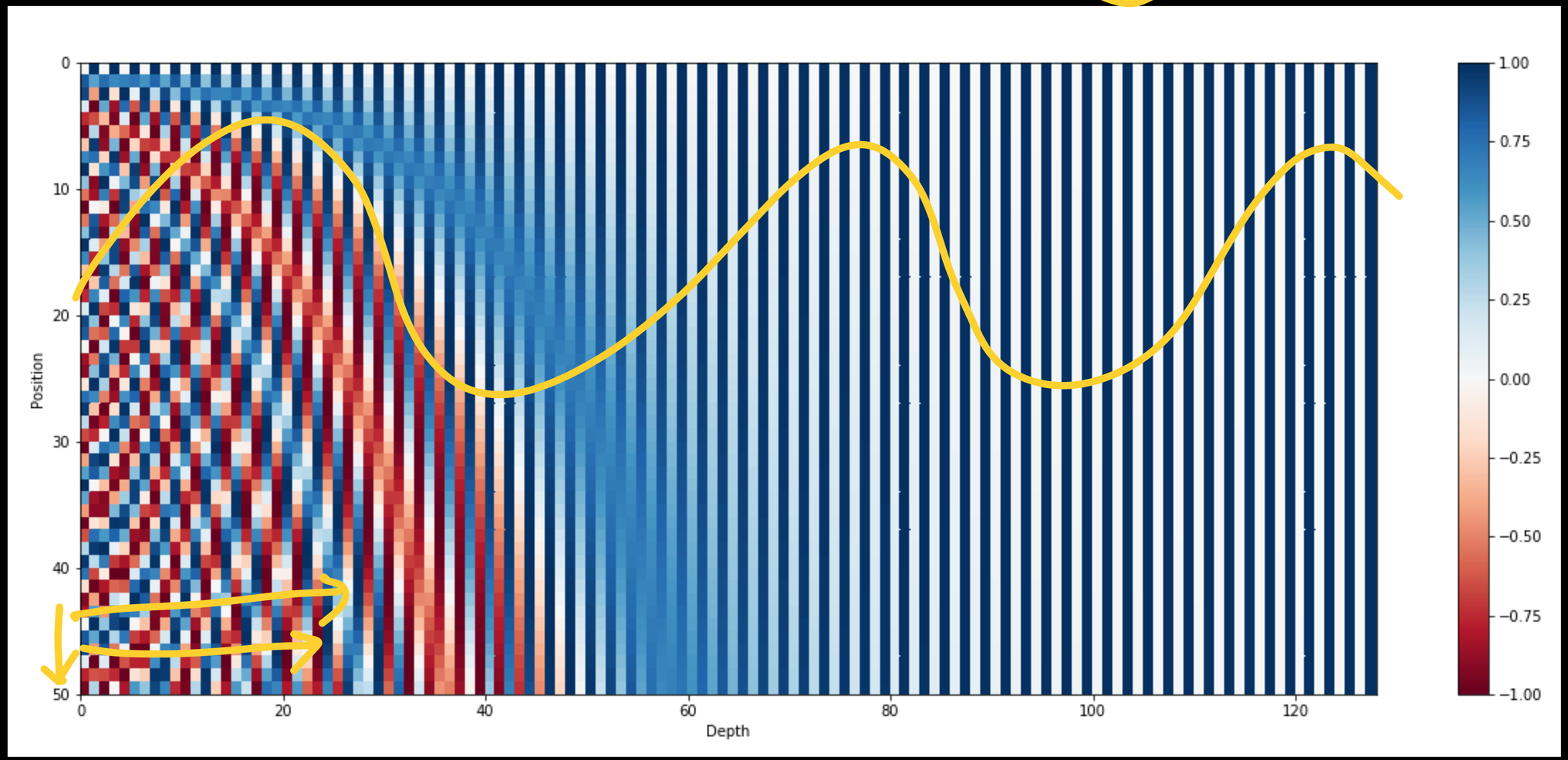
$$\mathbf{x}(t) = \mathbf{x}(t) + \mathbf{p}(t)$$

$x(1)\ x(2) \quad \ldots \quad x(T)$

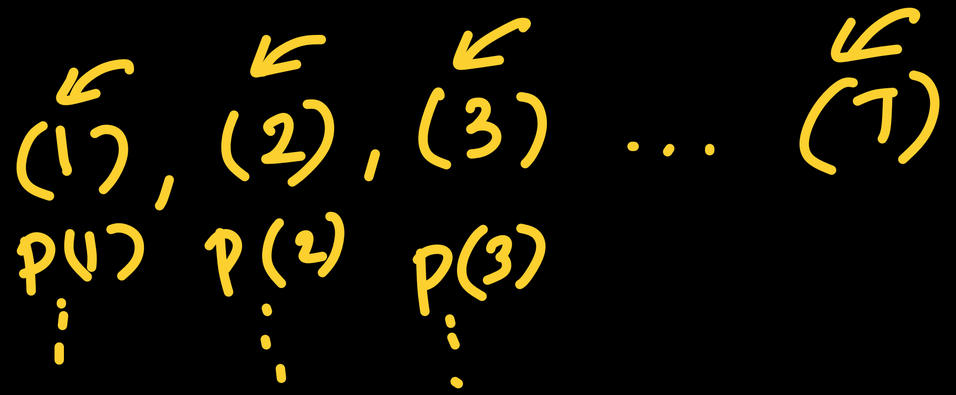
$X = \{x(1) \ldots x(T)\} \in \mathbb{R}^{T \times D}$

# Positional encoding

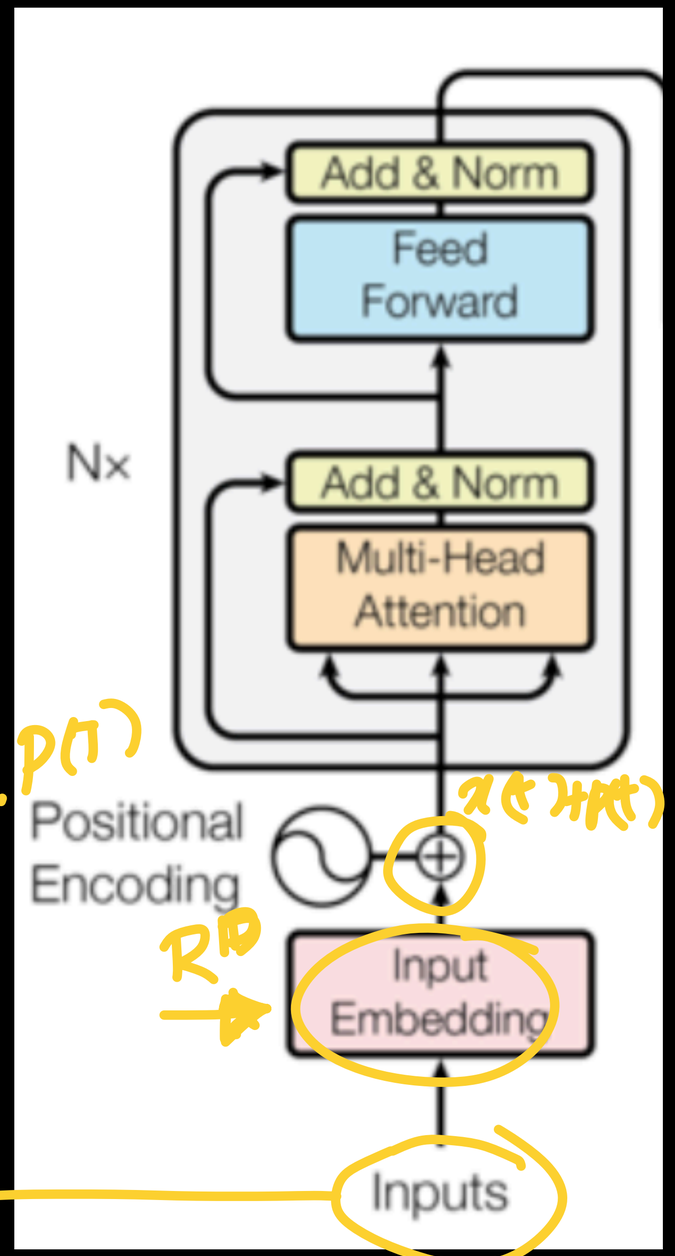

- An example used in the first paper $\mathbf{p}(t) \in \mathcal{R}^D$ [T=50, D=128]



# Transformer encoder - overview

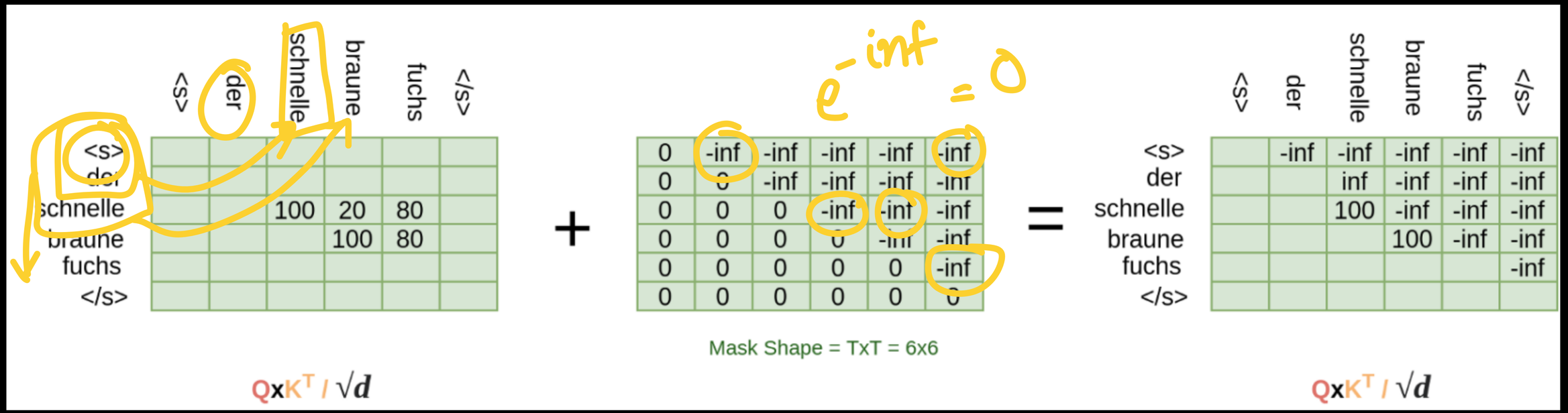



Reading Assignment - “Attention is All You Need”
https://arxiv.org/pdf/1706.03762.pdf



# Transformer decoder

- Masked self-attention layer -
  - ✓ Mask makes the output dependecies causal
    - ★ Only the past is used to encode the attention.




Source - https://lionbridge.ai/articles/what-are-transformer-models-in-machine-learning/

# Transformer decoder

- Masked self-attention layer -
  - ✓ Mask makes the output dependencies causal
    - ★ Only the past is used to encode the attention.

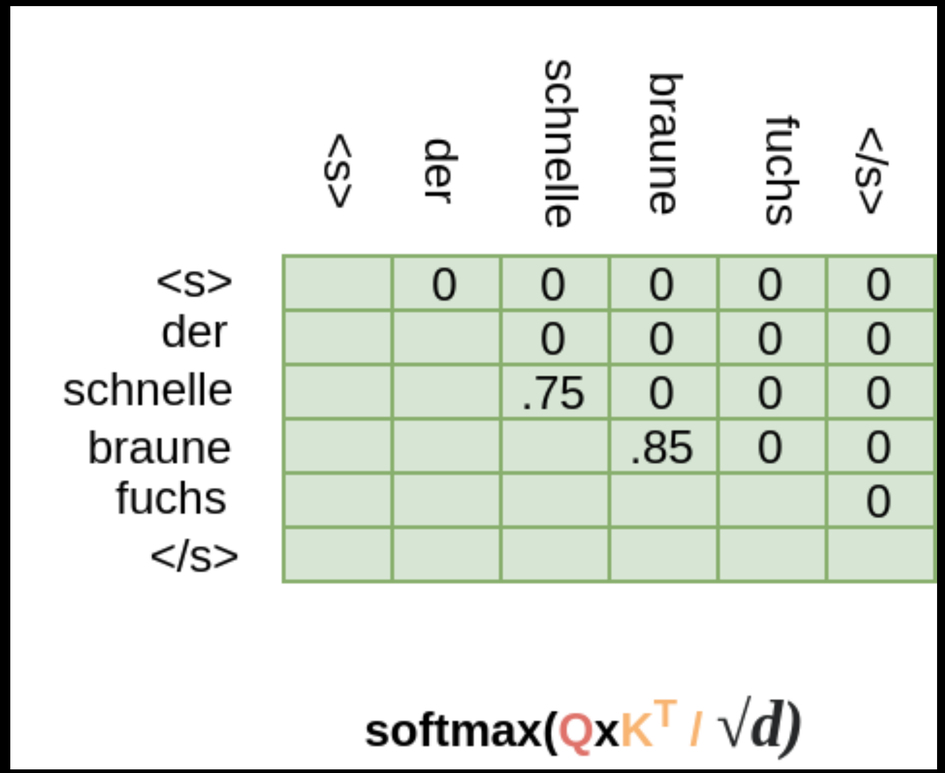
$$Softmax\left\{\frac{\mathbf{QK}^T}{\sqrt{d}}\right\}\mathbf{V} \longrightarrow Softmax\left\{Mask + \frac{\mathbf{QK}^T}{\sqrt{d}}\right\}\mathbf{V}$$

$A_n^?$

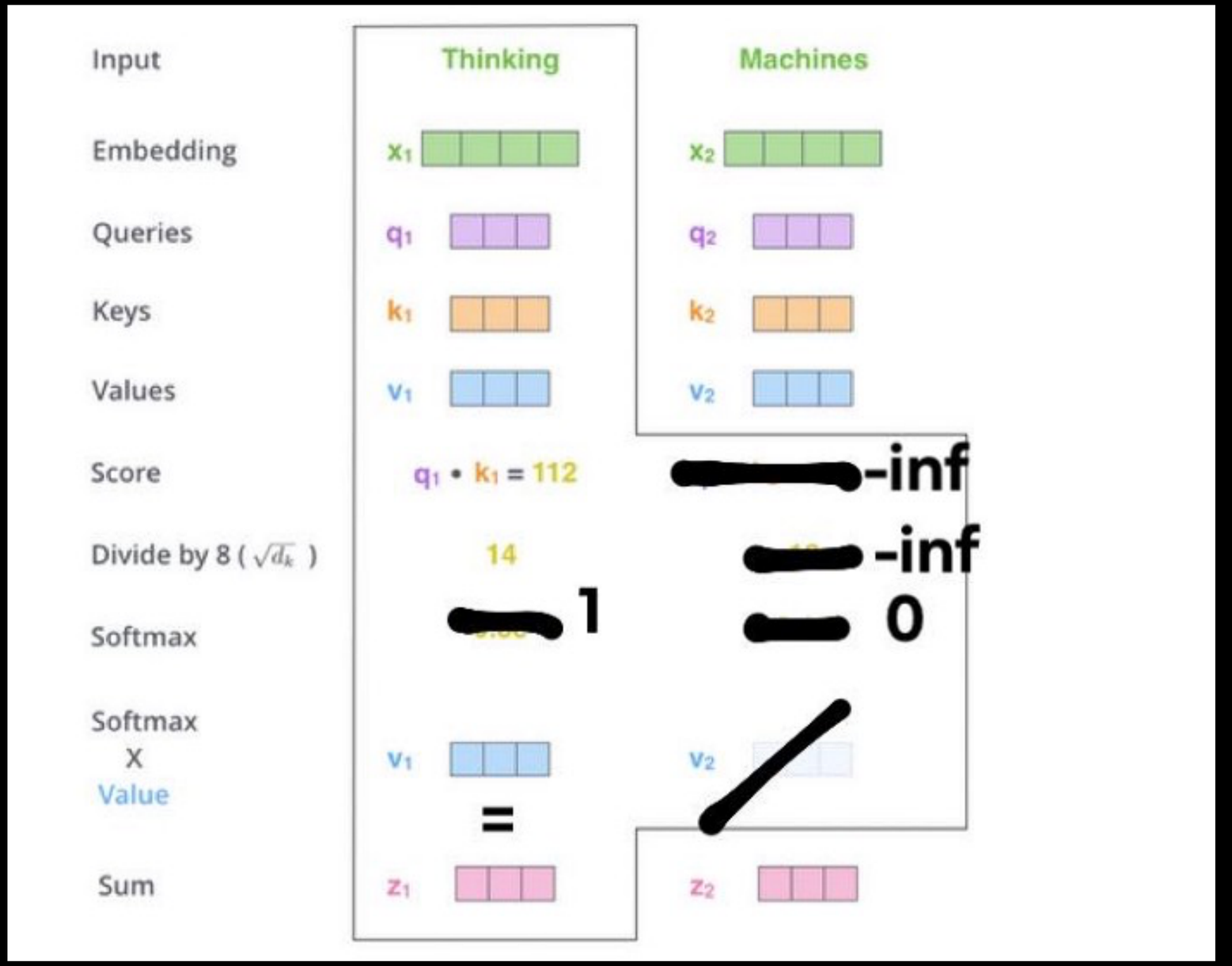
    - ★ Make the attention matrix to be lower triangular

# Transformer decoder

- Masked self-attention layer -
  - ✓ Mask makes the output dependecies causal
    - ★ Only the past is used to encode the attention.



# Transformer decoder



# Encoder-decoder attention

$x = \{x(1) \ldots x(T)\}$ $T \times D$

$\hat{y} = \{\hat{y}(1) \ldots \hat{y}(S)\}$ $S \times D$.

- Use the Key and Value matrices from the last layer of the encoder

$$\mathbf{Q}_h^p = \mathbf{D}^{p-1}\mathbf{W}_h^{p,Q} + \mathbf{1}(\mathbf{b}_h^{p,Q})^T \in \mathcal{R}^{S \times d}$$

$$\mathbf{K}_h^p = \mathbf{E}^L\mathbf{W}_h^{p,K} + \mathbf{1}(\mathbf{b}_h^{p,K})^T \in \mathcal{R}^{T \times d}$$

$$\mathbf{V}_h^p = \mathbf{E}^L\mathbf{W}_h^{p,V} + \mathbf{1}(\mathbf{b}_h^{p,V})^T \in \mathcal{R}^{T \times d}$$

$p = 1 \ldots P$ ↑ decoder layer

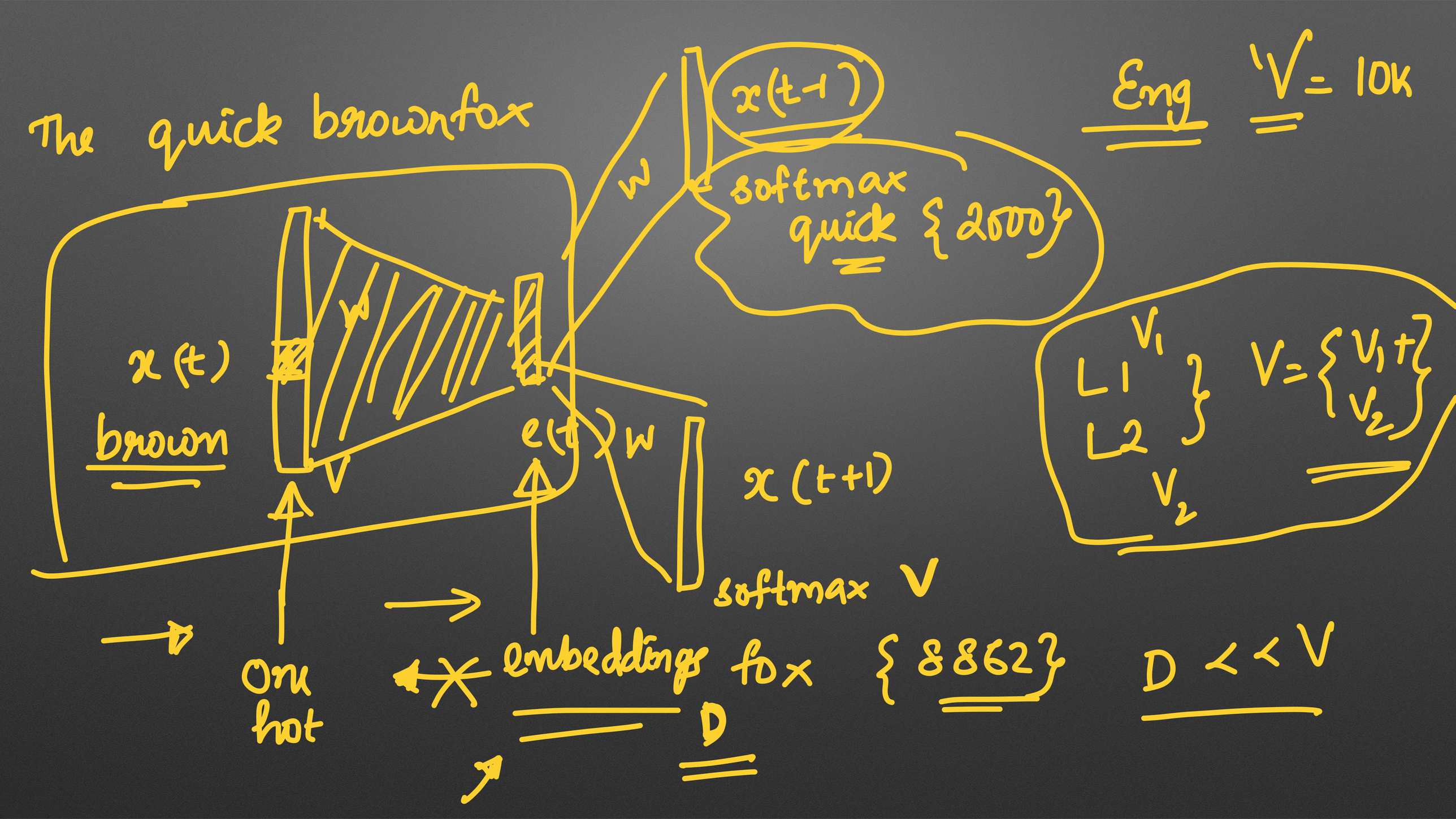
$$\mathbf{D}_h^p = softmax\left(\frac{\mathbf{Q}_h^p(\mathbf{K}_h^p)^T}{\sqrt{d}}\right)\mathbf{V}_h^p \in \mathcal{R}^{S \times d}$$

$S \times T$ .

$(S \times d)(d \times d) \rightarrow S \times d$

$S \times D$

$h = \{1..H\}$ heads $d = \frac{D}{H}$


The quick brownfox
x(t-1)
W
softmax
quick {2000}
Eng
V = 10K
x(t)
brown
W
e(t)
W
x(t+1)
L1
L2
V1
V2
V = {V1 t}
V2
softmax V
One hot
embeddings
fox {8862}
D
D << V

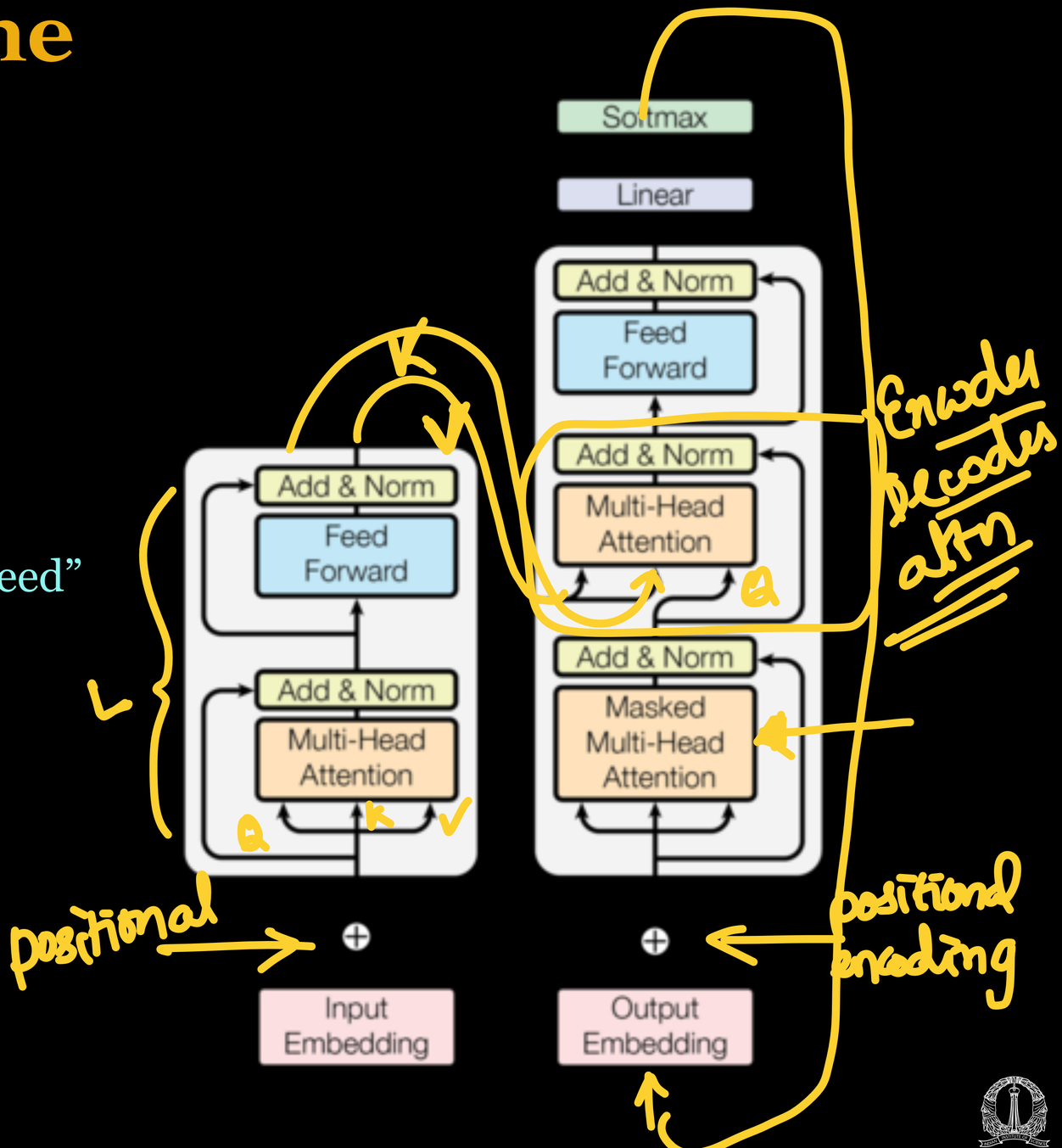
# Transformer - decoder

- Decoder Layer Output

- $$[\mathbf{d}^p(1)...\mathbf{d}^p(S)] = ReLU\left(\mathbf{D}^p_{ff}\mathbf{W}^p_{of} + \mathbf{1}(\mathbf{b}^p_{of})^T\right) \in \mathcal{R}^{S\times D}$$

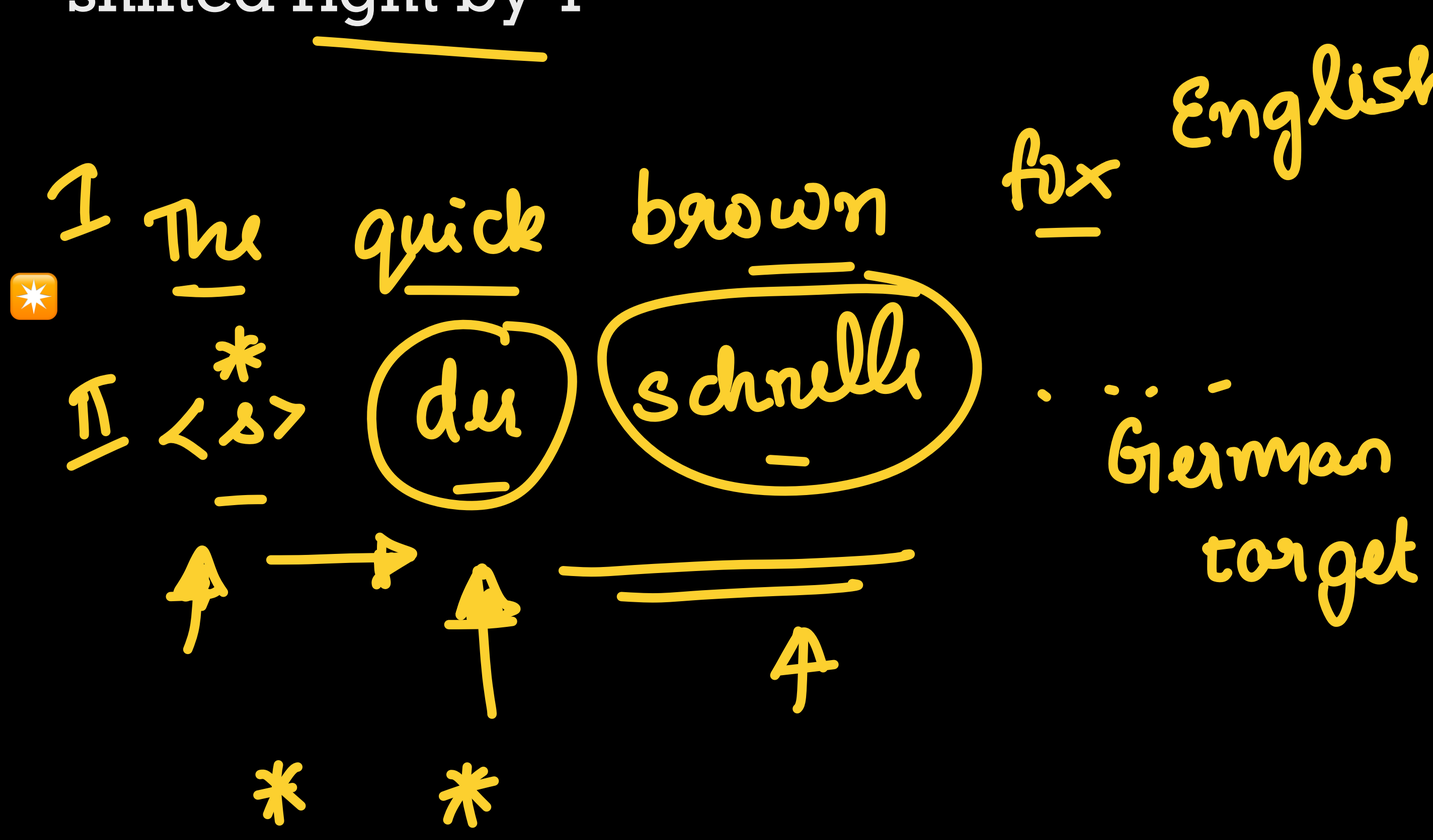
# Transformer - full pipeline

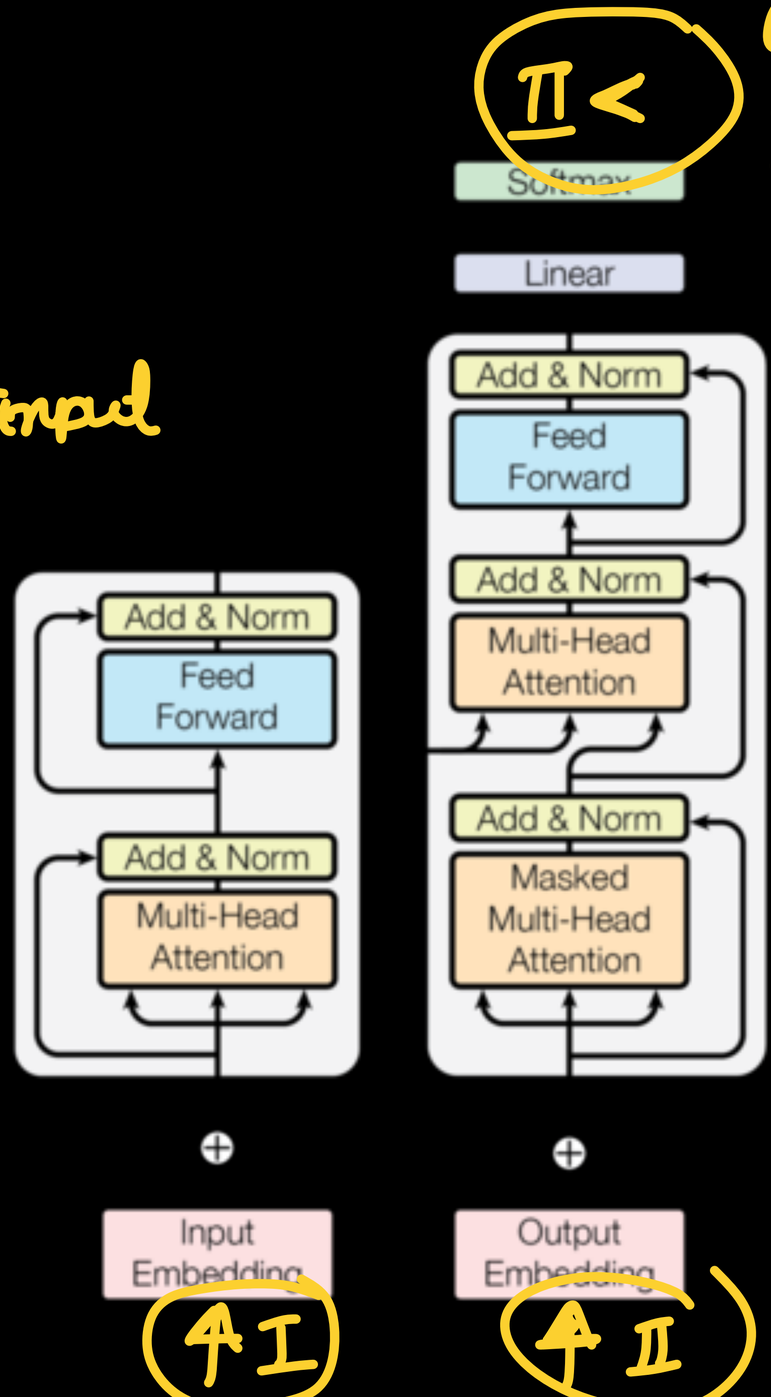
Reading Assignment - “Attention is All You Need”
https://arxiv.org/pdf/1706.03762.pdf



# Transformer - training

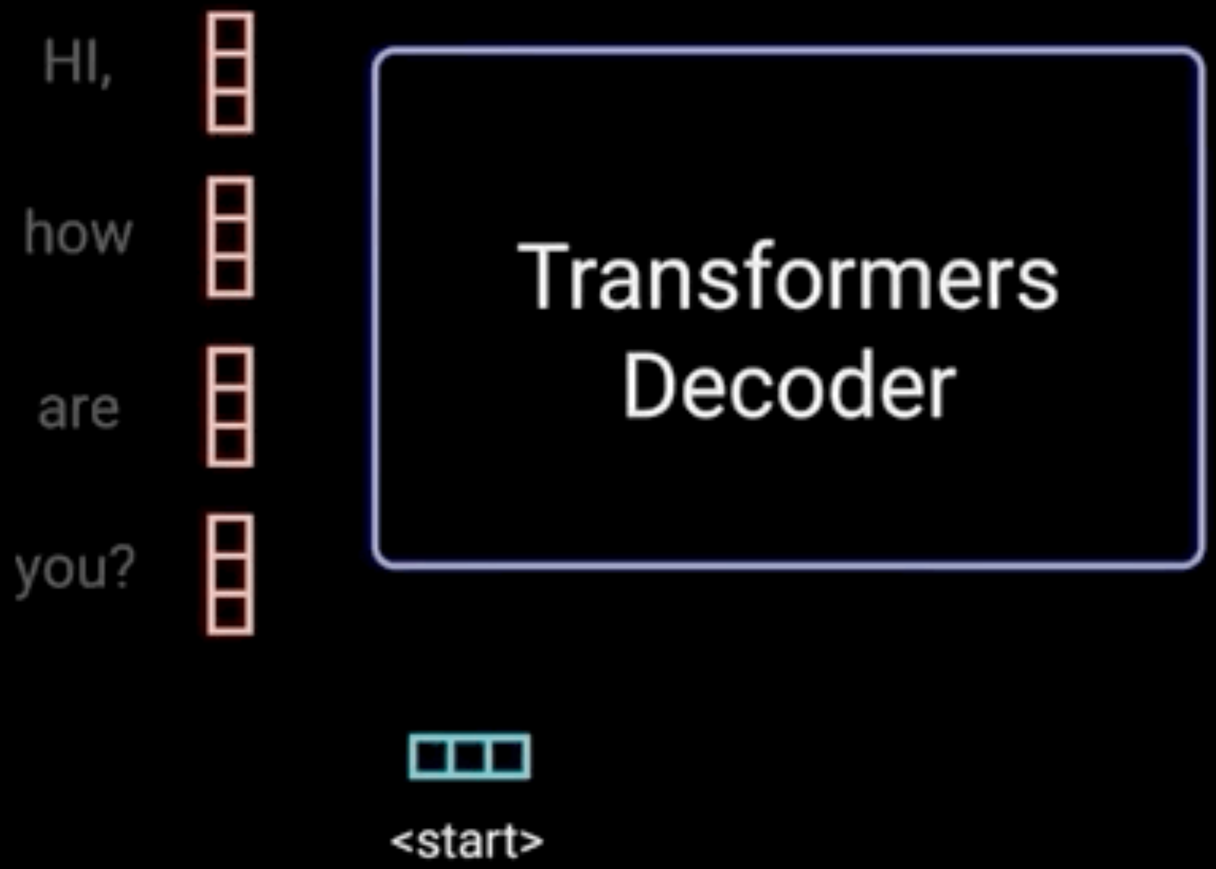
- Input sequence and the output sequence shifted right by 1



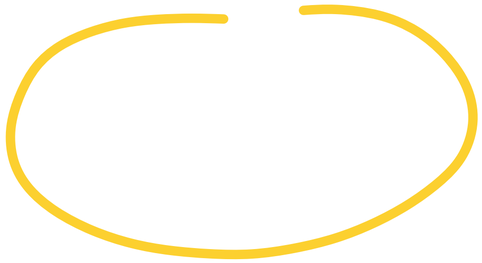



# Transformer decoder

- Inference



# Greedy decoding





Source - https://lionbridge.ai/articles/what-are-transformer-models-in-machine-learning/

# Topics thus far …

**Visual and Time Series Modeling:** Semantic Models, Recurrent neural models and LSTM models, Encoder-decoder models, Attention models.

**Representation Learning, Causality And Explainability:** t-SNE visualization, Hierarchical Representation, semantic embeddings, gradient and perturbation analysis, Topics in Explainable learning, Structural causal models.

**Unsupervised Learning:** Restricted Boltzmann Machines, Variational Autoencoders, Generative Adversarial Networks.

**New Architectures:** Capsule networks, End-to-end models, Transformer Networks.

**Applications:** Applications in NLP, Speech, Image/Video domains in all modules.