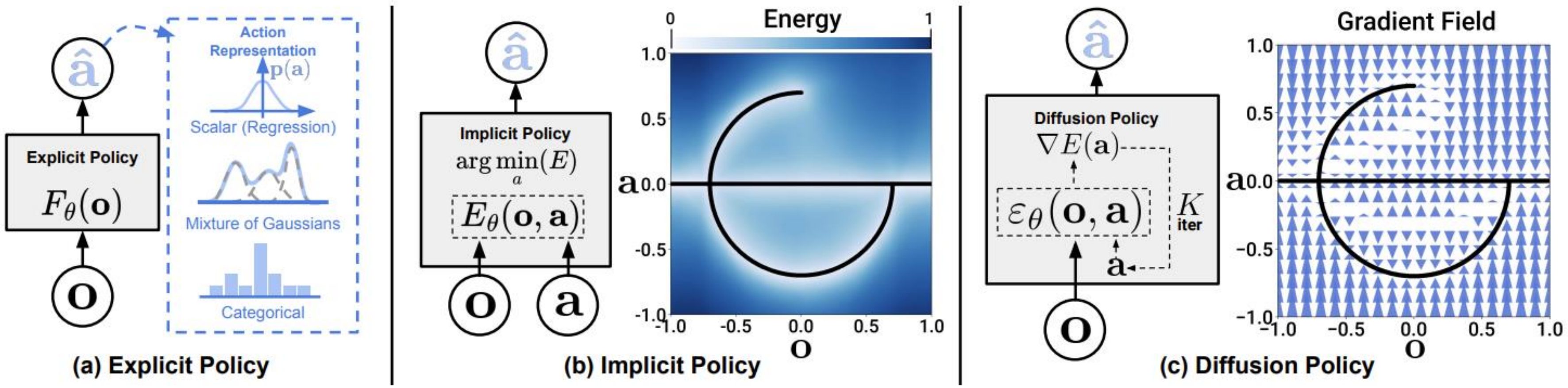
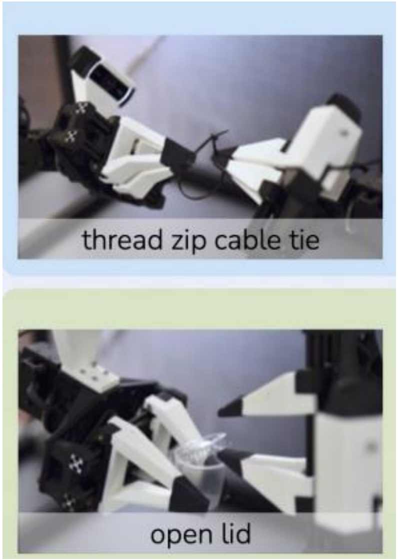
# Advanced Behavioral Cloning

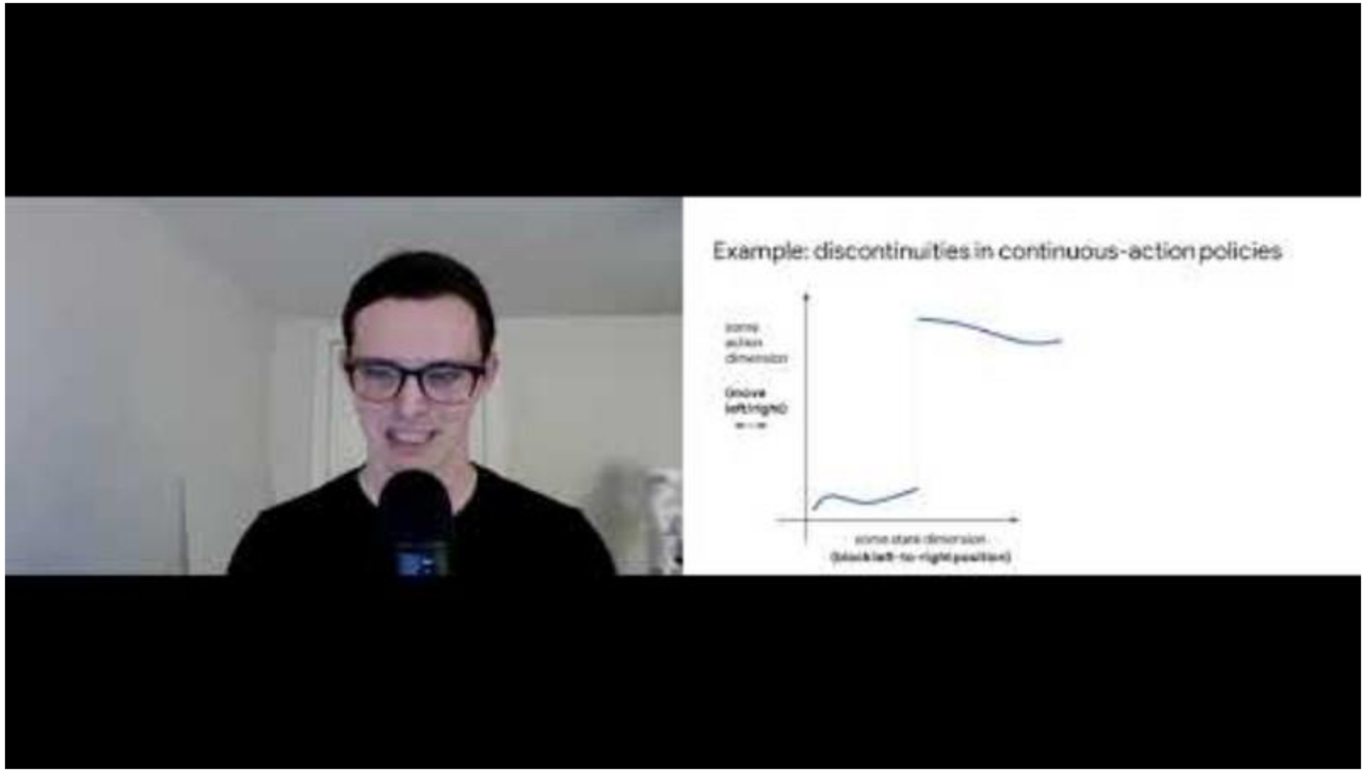



Instructor: Daniel Brown

# Implicit Behavioral Cloning

- Paper: https://arxiv.org/abs/2109.00137
- Video: https://www.youtube.com/watch?v=QslGqRUSRzs

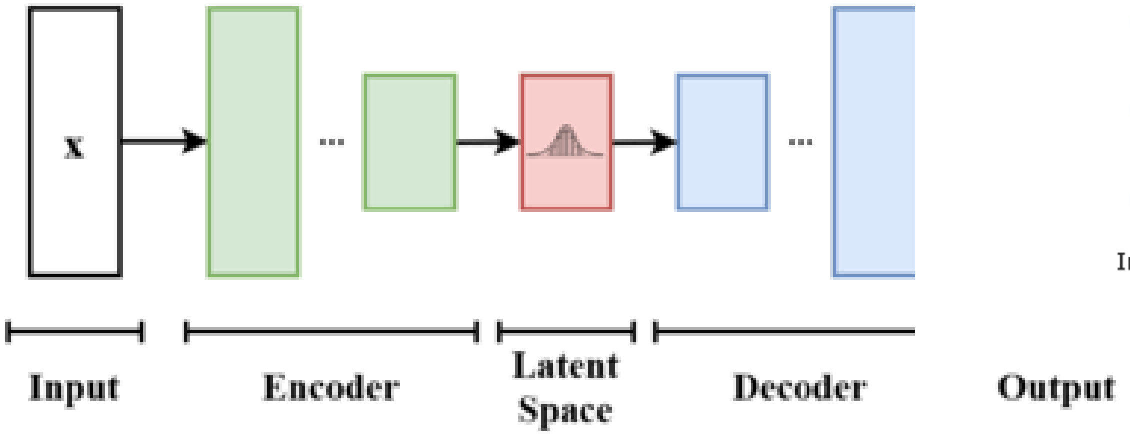


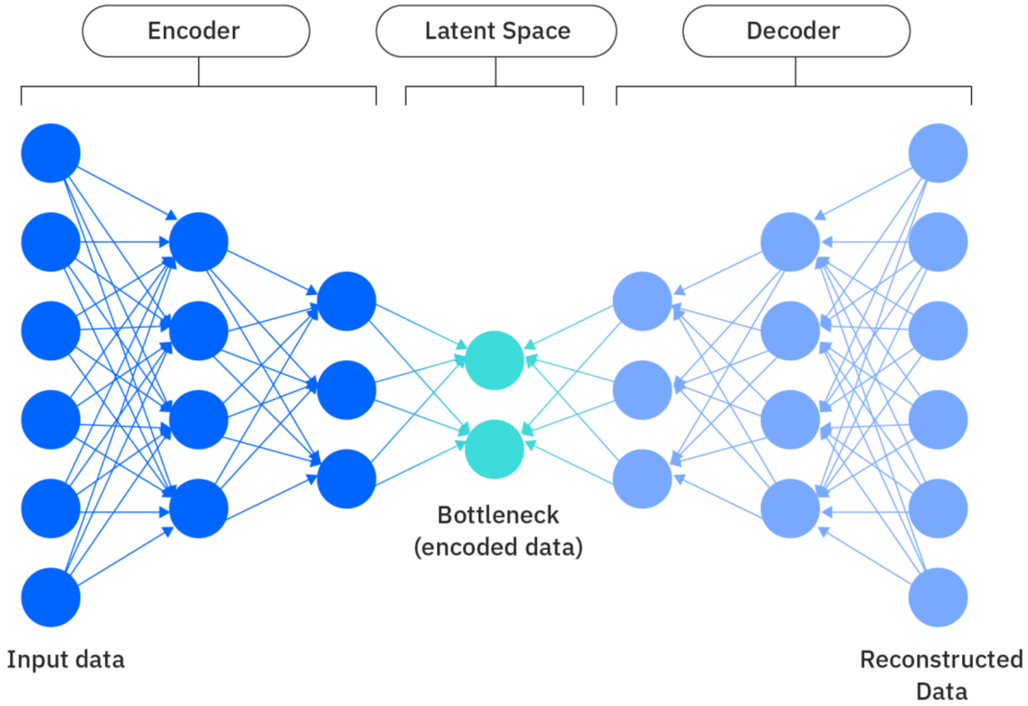
# Action Chunking with Transformers (ACT)

- Paper: https://arxiv.org/pdf/2304.13705
- Videos: https://tonyzhaozh.github.io/aloha/

# Variational Autoencoders (VAEs)

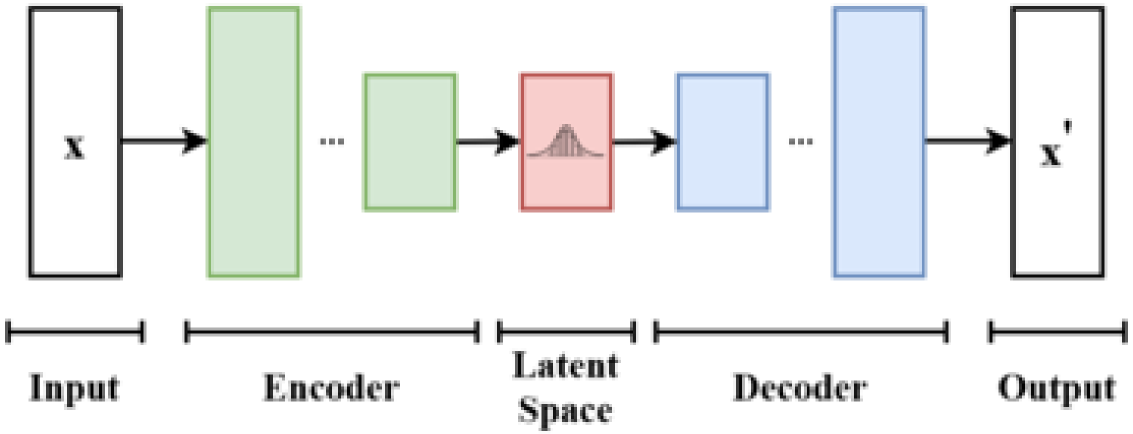
- Autoencoders learn latent representations



# Variational Autoencoders (VAEs)

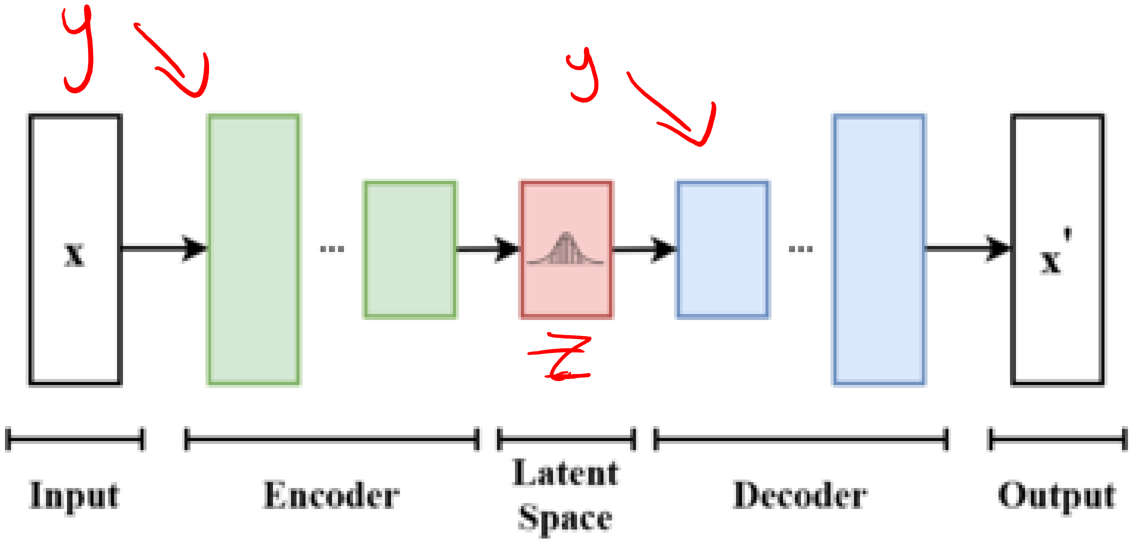
- Autoencoders learn latent representations
- VAEs map input into a distribution over latent variables z
- Loss function is reconstruction plus KL divergence




$$\mathcal{L} = \mathbb{E}_{q(z|x)}[\log p(x|z)] - D_{\mathrm{KL}}(q(z|x)||p(z))$$

# Conditional Variational Autoencoders (CVAEs)

- Encoder and decoder both condition on extra info y




- Loss function is reconstruction plus KL divergence

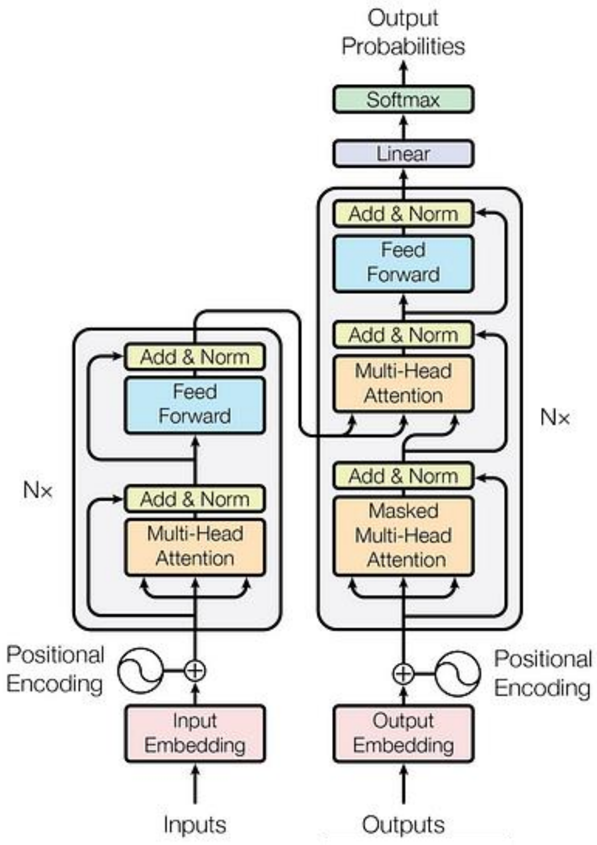
$$\mathcal{L} = \mathbb{E}_{q(z|x,y)}[\log p(x|z,y)] - D_{\mathrm{KL}}(q(z|x,y)||p(z|y))$$

# Transformers

- State of the art ways to ingest and output sequential data.

BERT

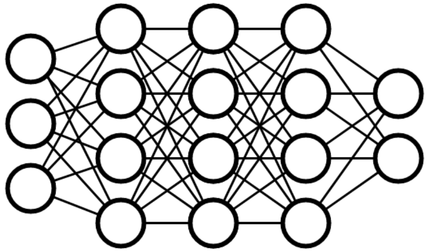
Encoder



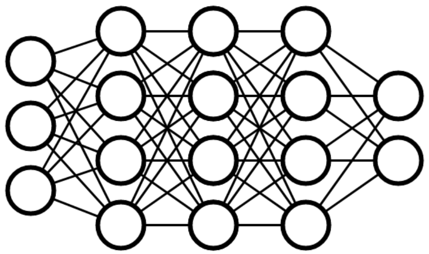

GPT

Decoder

# **Neural** language modeling

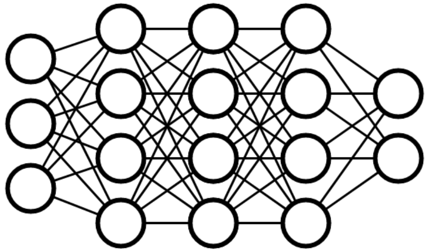
[BOS] →  → Sylvester

# Neural language modeling

[BOS] Sylvester →  → Stallone

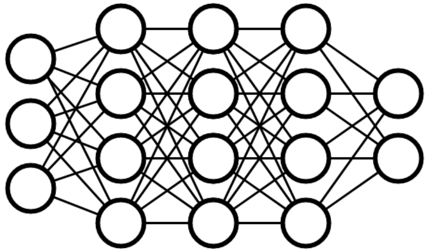
# **Neural** language modeling

[BOS] Sylvester Stallone →

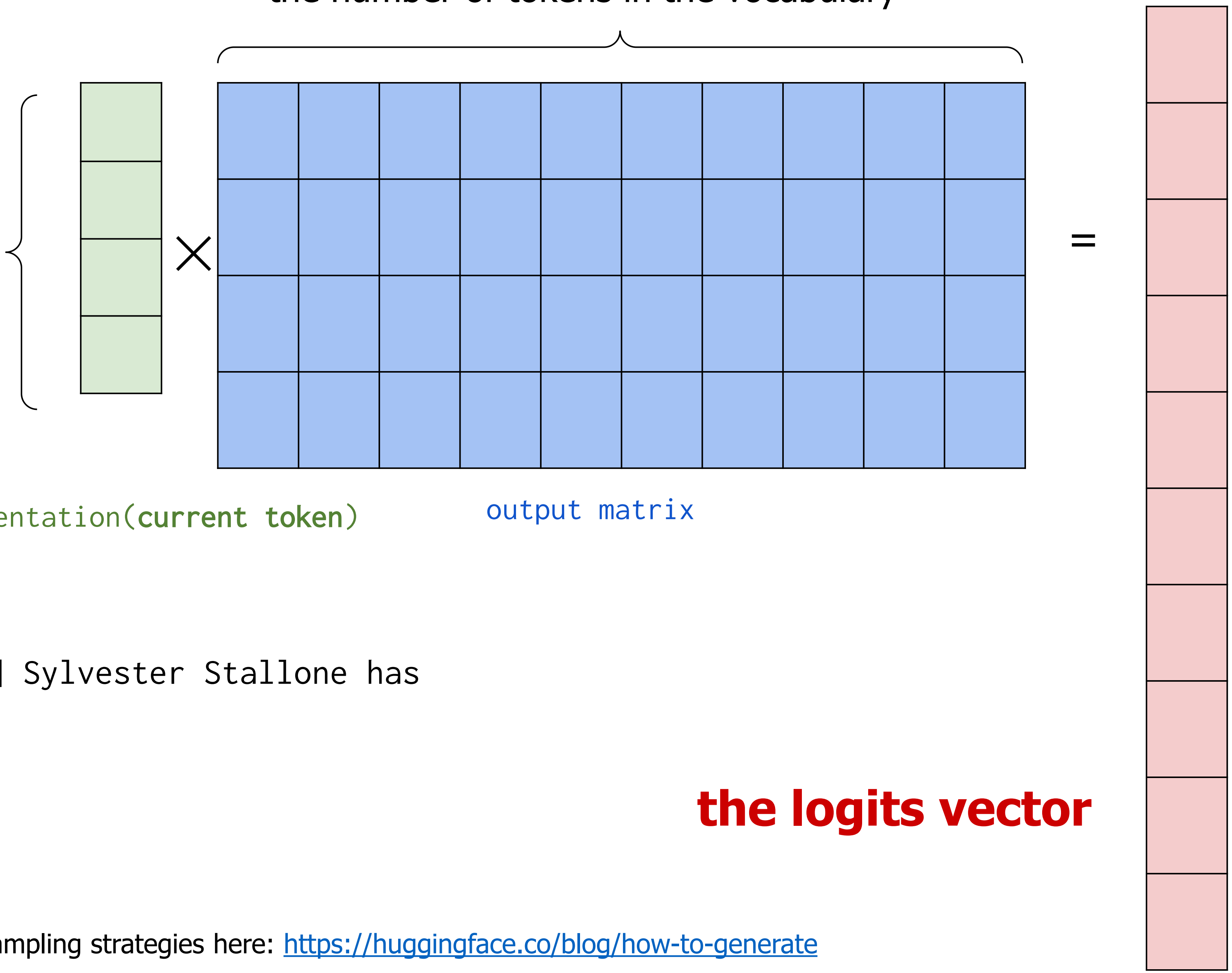


→ has

# **Neural** language modeling

[BOS] Sylvester Stallone has →  → made the number of tokens in the vocabulary

the size of the vector representation up to and including the **current token**



representation(**current token**)

output matrix

"+" softmax

i-th dimension ~ the "probability" [not really] that the **next token** is the i-th token in the vocabulary

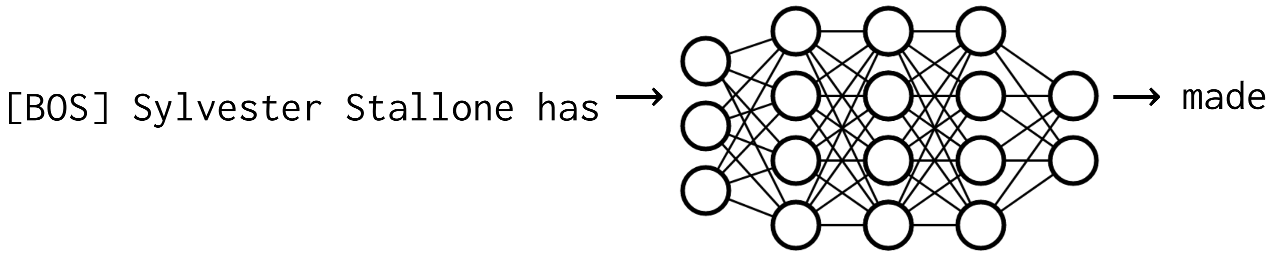
select the token with the high(est) "probability" as a token to display (generate)

```
[BOS] Sylvester Stallone has
```

**the logits vector**

Read about other sampling strategies here: [https://huggingface.co/blog/how-to-generate](https://huggingface.co/blog/how-to-generate)

# **Neural** sequence modeling

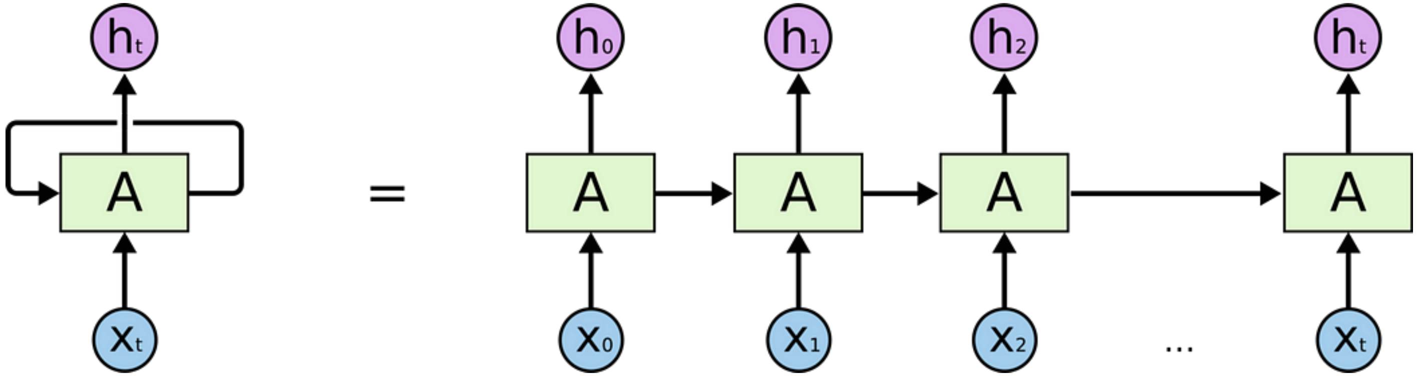



Problems:

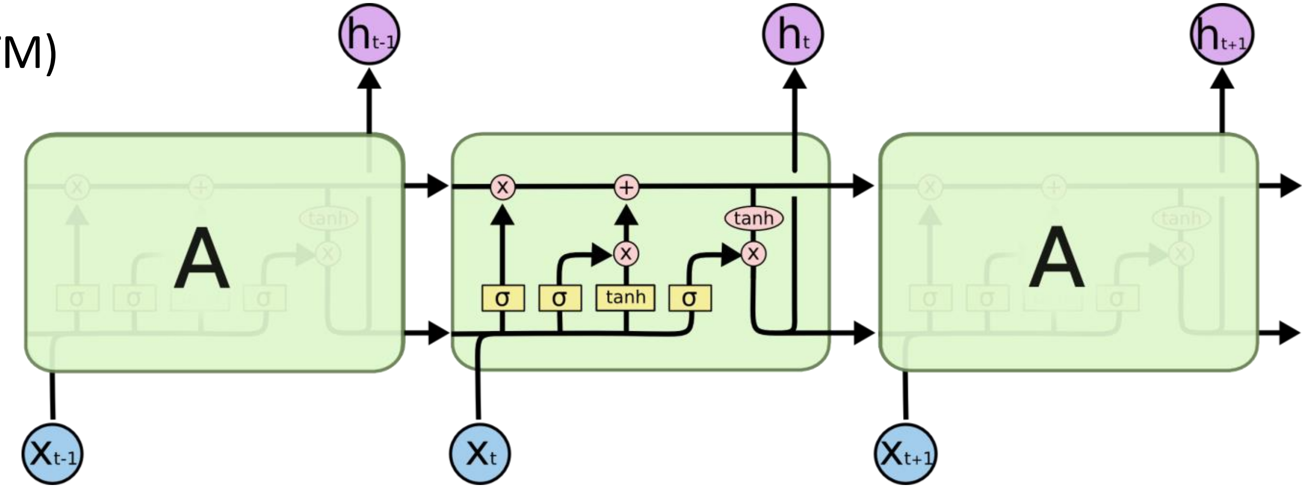
- How do we deal with different length inputs?
- How do we model long-range dependencies?

# Recurrent Neural Networks

- Standard RNN

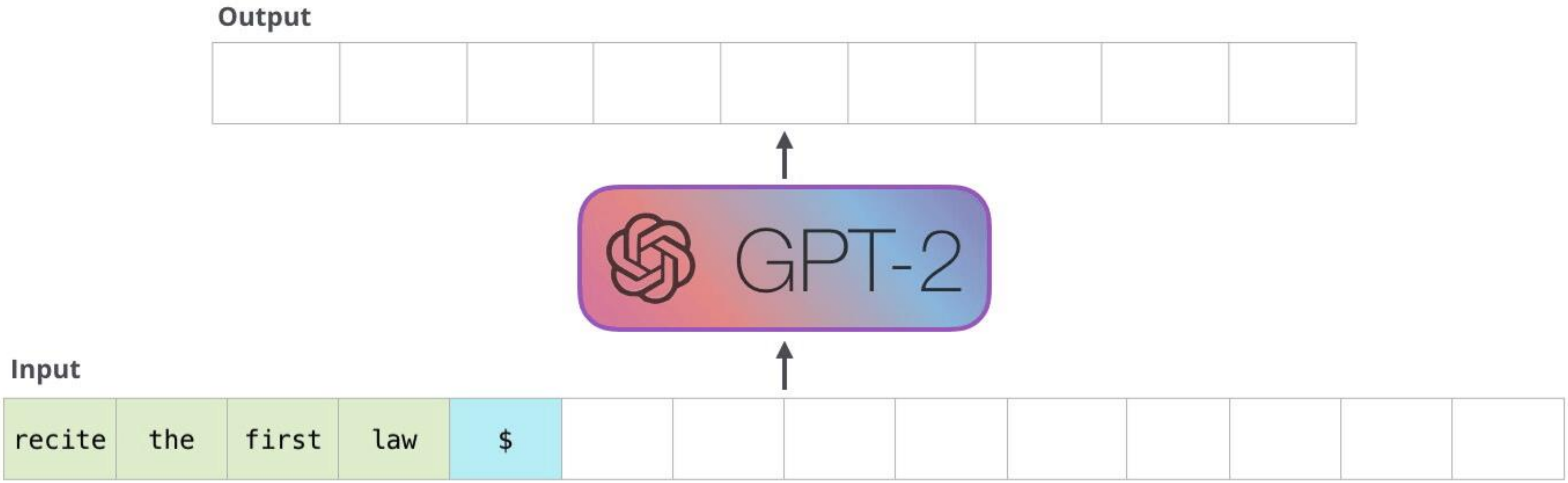


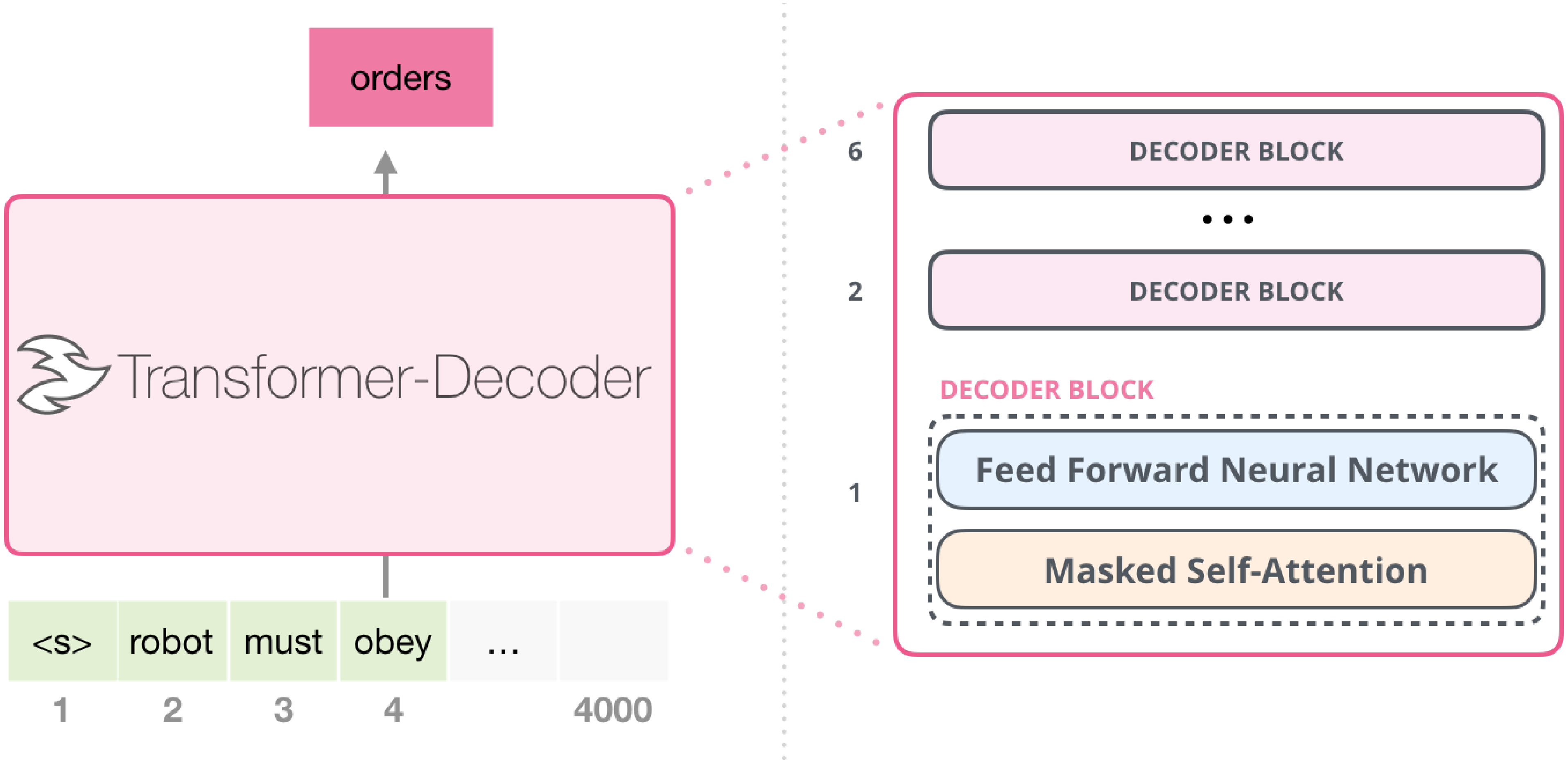

- Long short-term memory (LSTM)



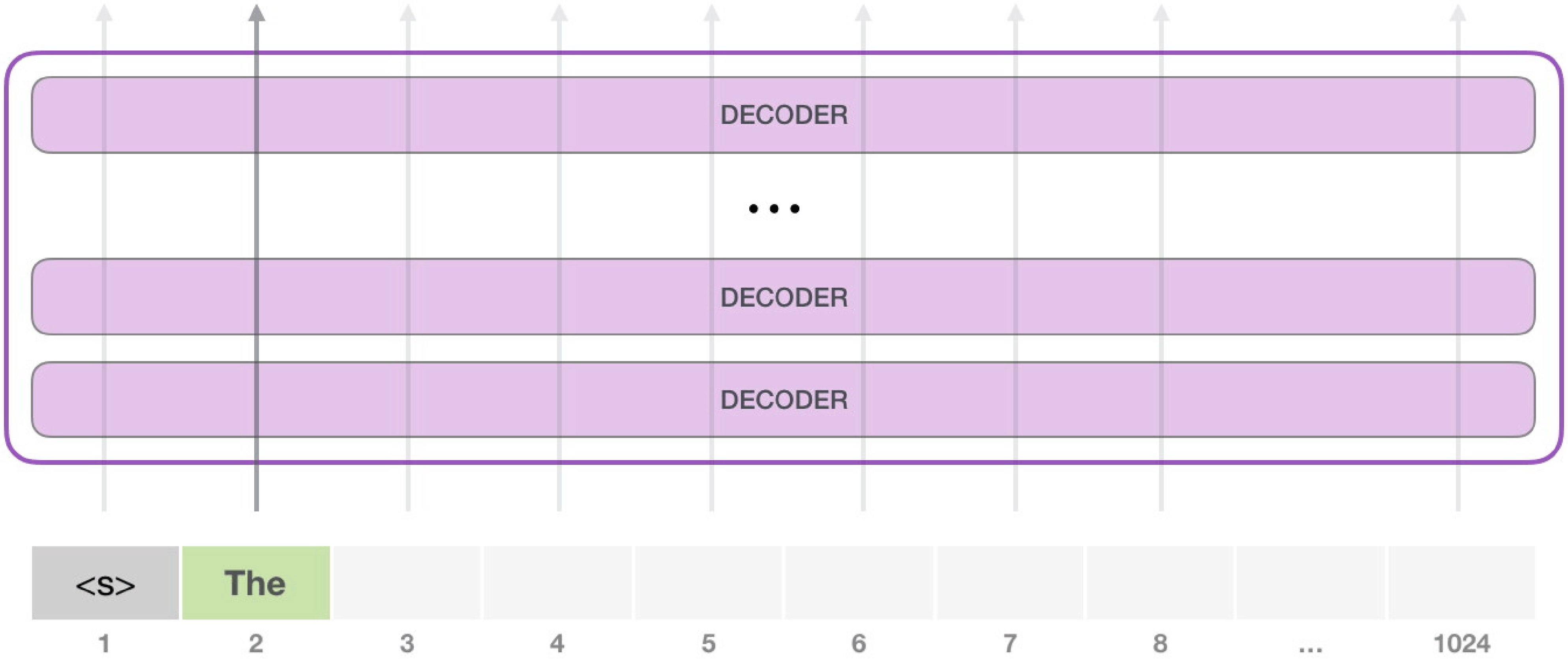

https://colah.github.io/posts/2015-08-Understanding-LSTMs/

# Large Language Models

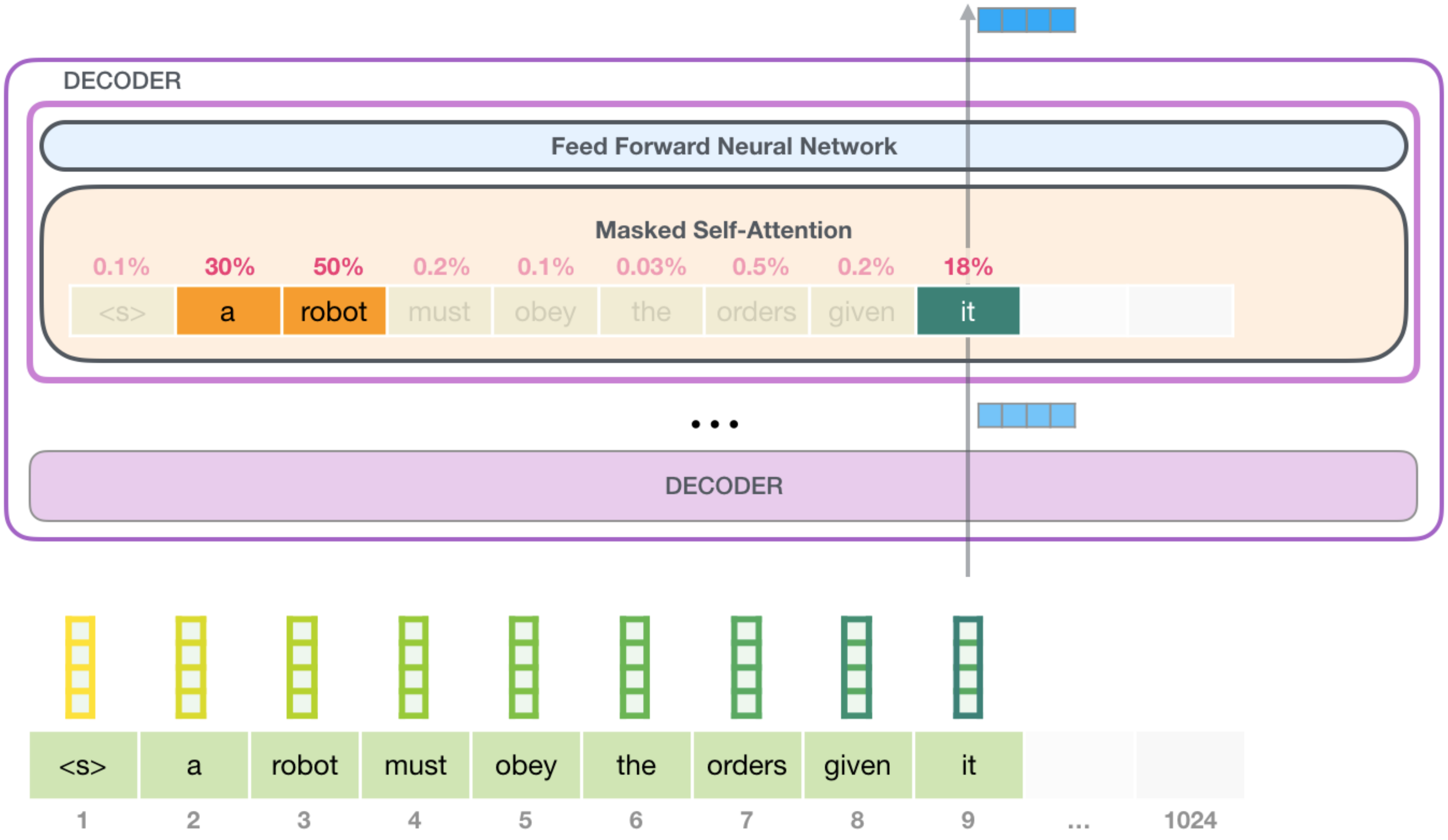


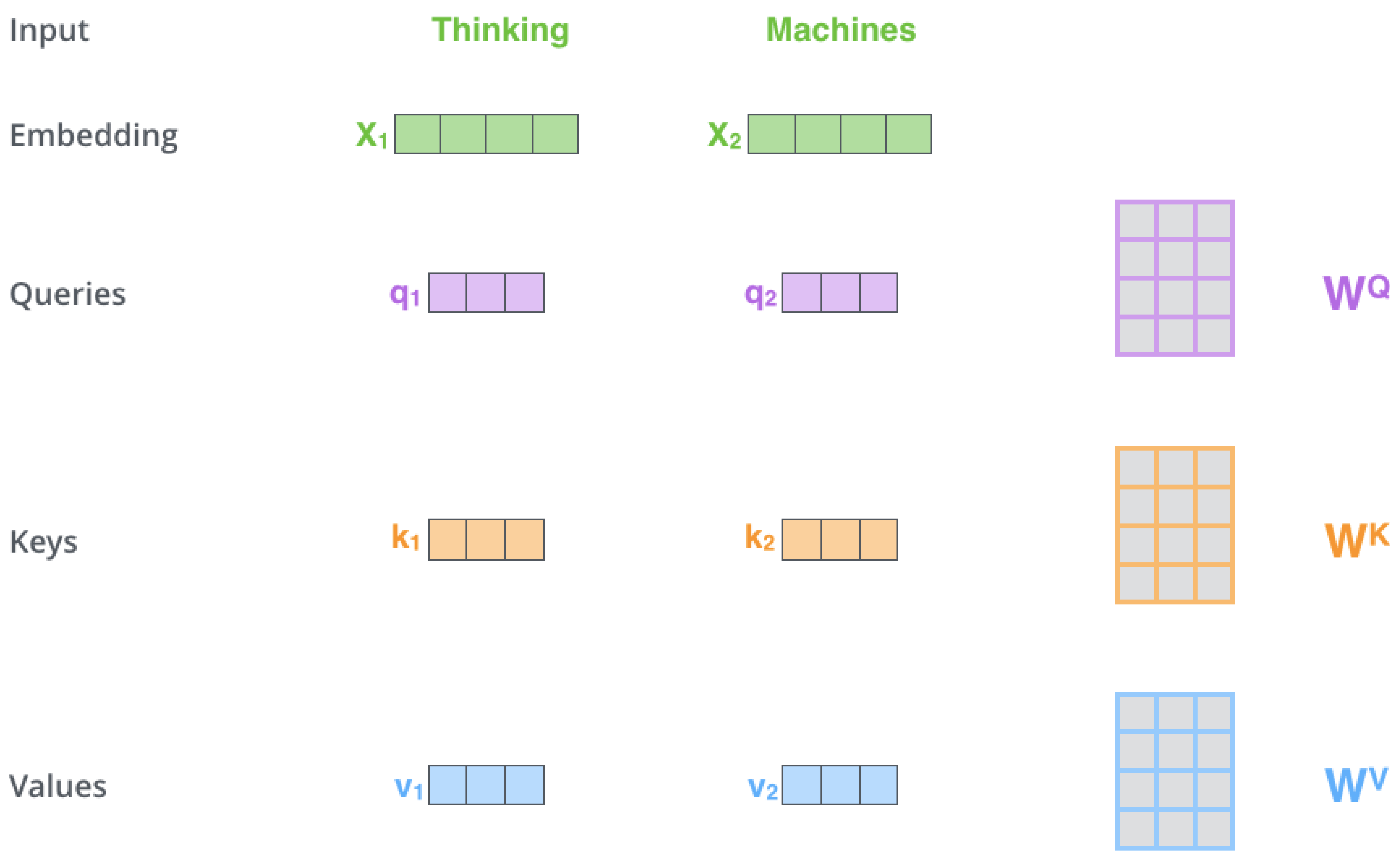


orders
Transformer-Decoder
<s>
robot
must
obey
...
1
2
3
4
4000
6
DECODER BLOCK
2
DECODER BLOCK
DECODER BLOCK
1
Feed Forward Neural Network
Masked Self-Attention

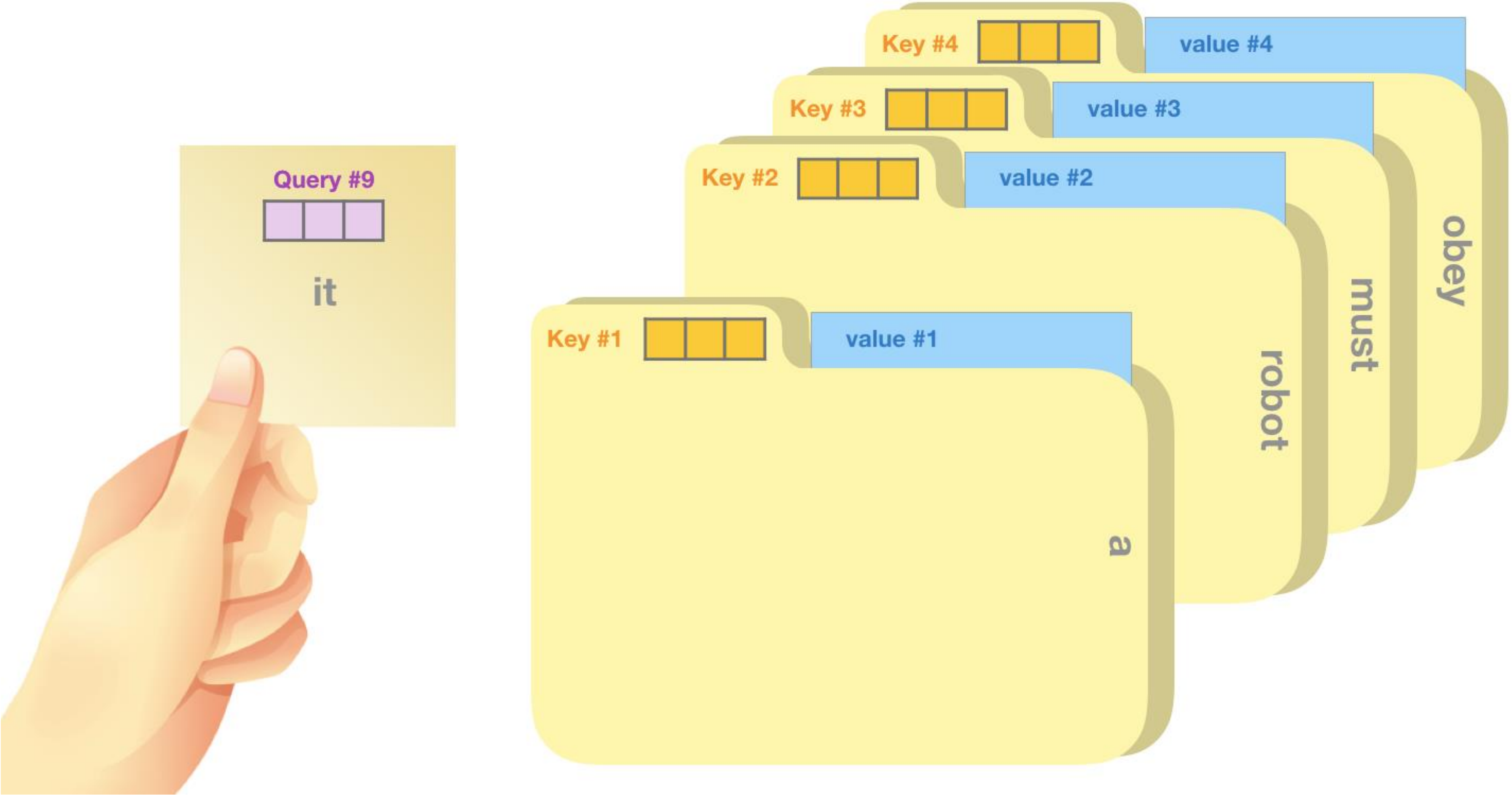

DECODER
...
DECODER
DECODER
<s>
The
1
2
3
4
5
6
7
8
...
1024

DECODER
• • •
DECODER
Token Embeddings
Positional Encodings
=
Positional encoding for token #1
+
Token embedding of <s>
<s>
1
2
...
1024

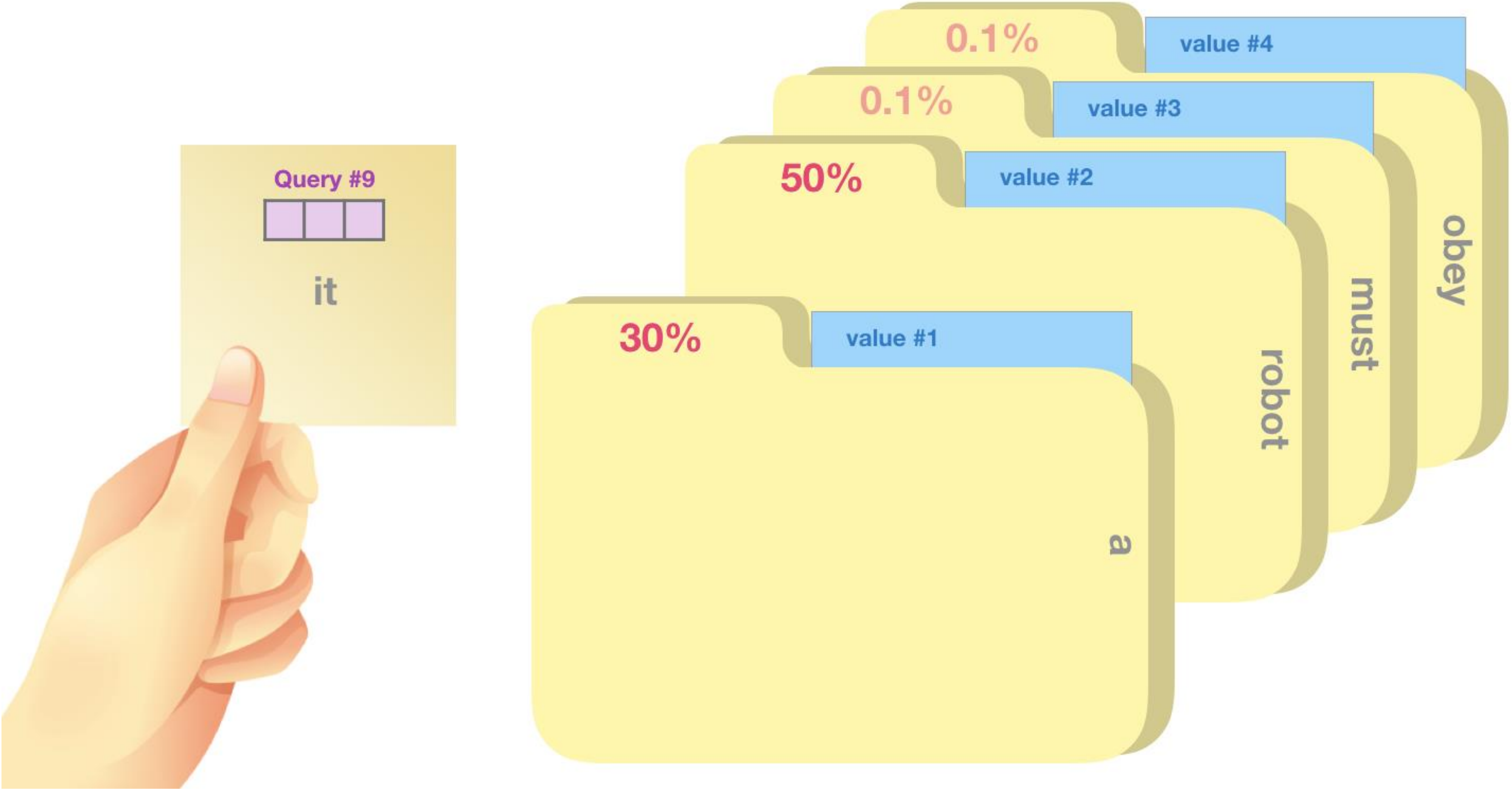

DECODER
Feed Forward Neural Network
Masked Self-Attention
0.1%
30%
50%
0.2%
0.1%
0.03%
0.5%
0.2%
18%
<s>
a
robot
must
obey
the
orders
given
it
DECODER
<s>
a
robot
must
obey
the
orders
given
it
1
2
3
4
5
6
7
8
9
...
1024


Input
Thinking
Machines
Embedding
$X_1$
$X_2$
Queries
$q_1$
$q_2$
$W^Q$
Keys
$k_1$
$k_2$
$W^K$
Values
$v_1$
$v_2$
$W^V$

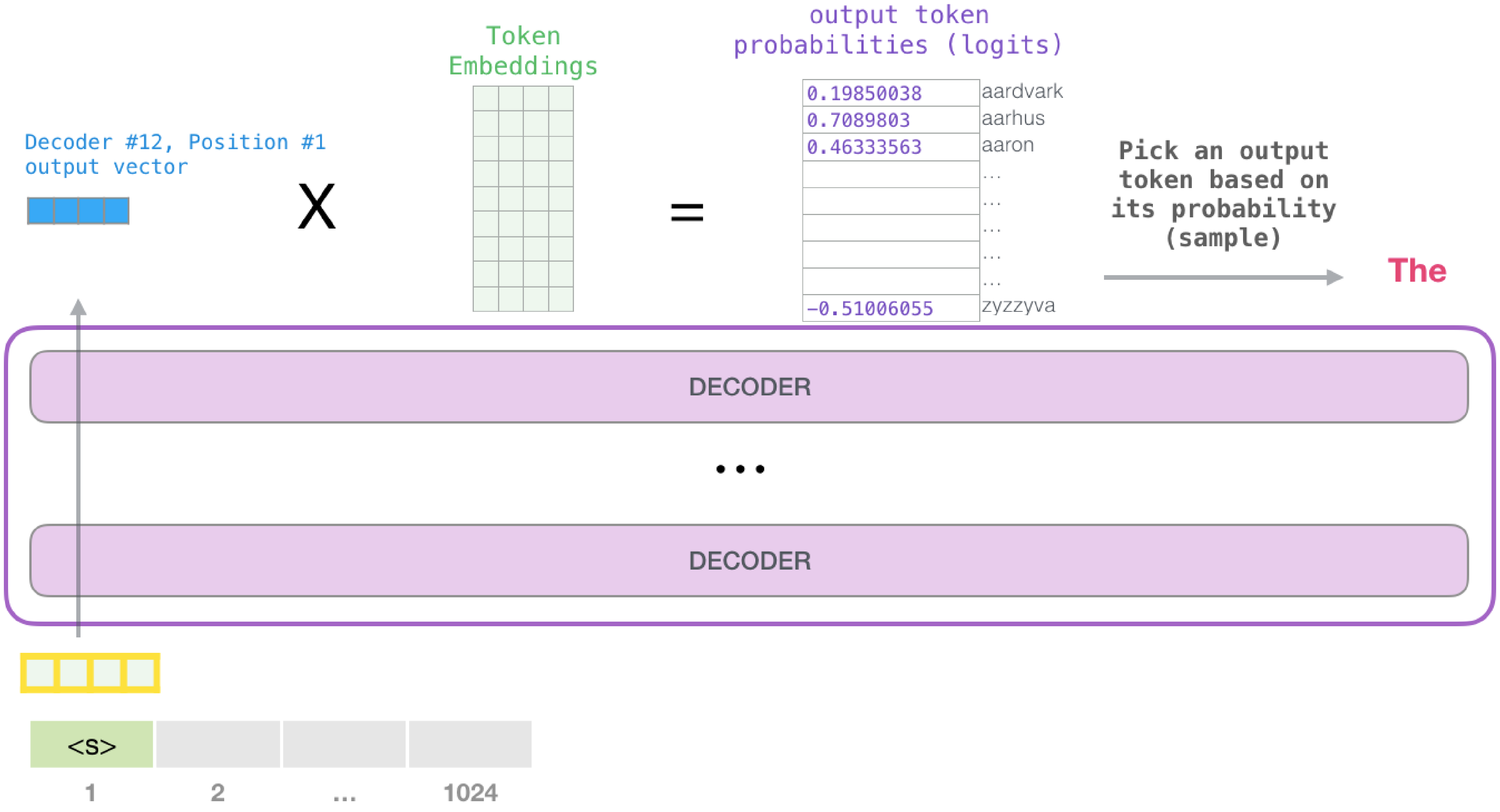
Perform dot product between query and all keys to get a raw score for each previous word (including current word).

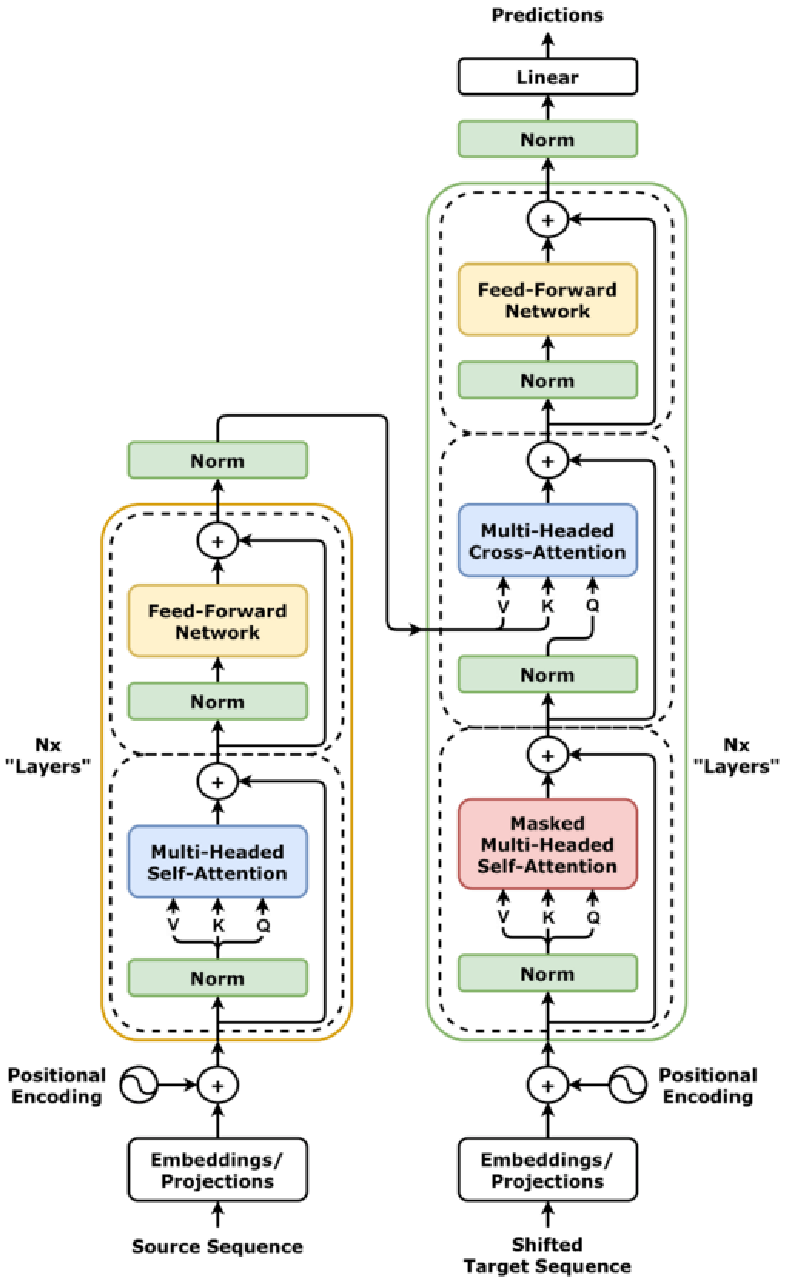


Normalize these scores via a softmax to get a probability distribution. Then return a weighted sum of the values.



| Word | Value vector | Score | Value X Score |
|---|---|---|---|
| <s> | | 0.001 | |
| a | | 0.3 | |
| robot | | 0.5 | |
| must | | 0.002 | |
| obey | | 0.001 | |
| the | | 0.0003 | |
| orders | | 0.005 | |
| given | | 0.002 | |
| it | | 0.19 | |
| | | | |
| | | Sum: | |


Decoder #12, Position #1
output vector
X
Token
Embeddings
=
output token
probabilities (logits)
0.19850038 aardvark
0.7089803 aarhus
0.46333563 aaron
...
-0.51006055 zyzzyva
Pick an output token based on its probability (sample)
The
DECODER
DECODER
<s>
1
2
...
1024


Predictions
Linear
Norm
Feed-Forward Network
Norm
Multi-Headed Cross-Attention
V K Q
Norm
Masked Multi-Headed Self-Attention
V K Q
Norm
Nx "Layers"
Positional Encoding
Embeddings/ Projections
Shifted Target Sequence
Norm
Feed-Forward Network
Norm
Multi-Headed Self-Attention
V K Q
Norm
Nx "Layers"
Positional Encoding
Embeddings/ Projections
Source Sequence

# Diffusion Policy

- Paper: https://arxiv.org/pdf/2303.04137v4
- Videos: https://diffusion-policy.cs.columbia.edu/

# Denoising Diffusion (high-level)