Transformers

CS 229 Summer 2022

Griffin Young

Outline

Motivation

Architecture

Training

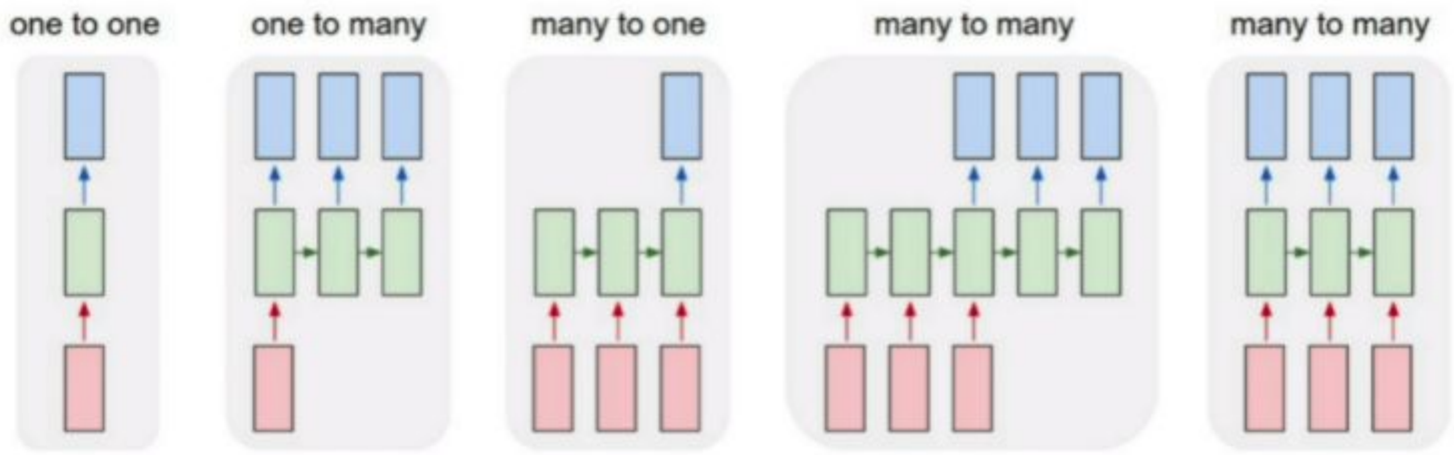
Results

Strengths and Limitations

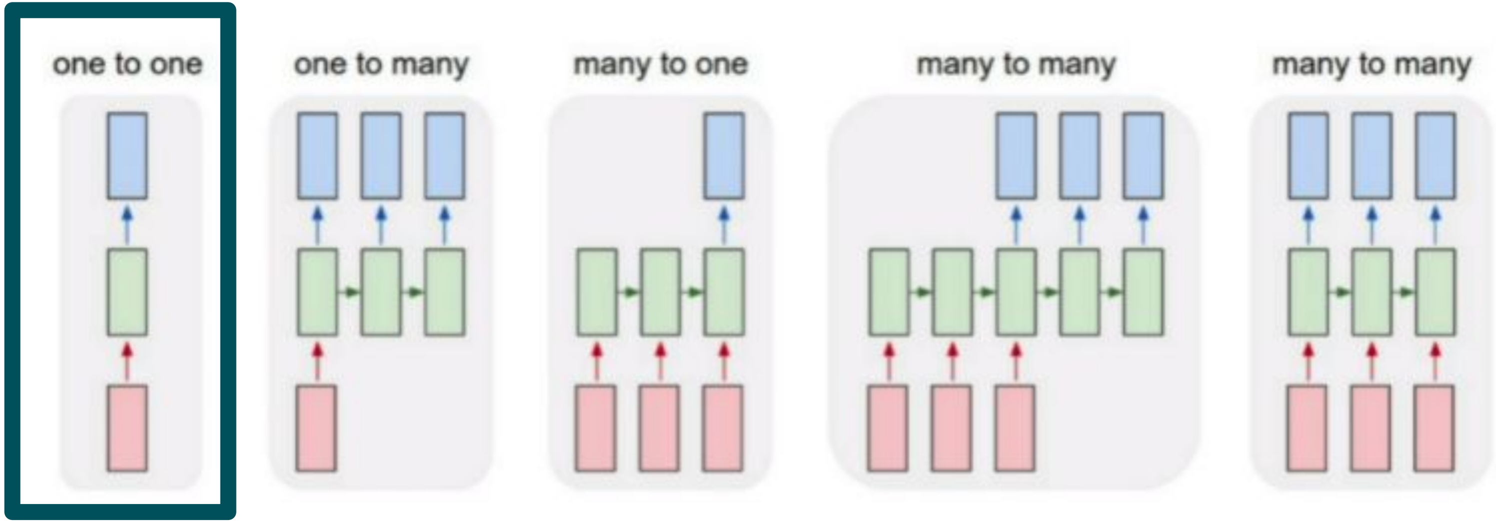
Motivation


"NEVER GIVE UP ON YOUR DREAMS, THE FUTURE IS BUILT ON DREAMS"
OPTIMUS PRIME

Sequence Problems

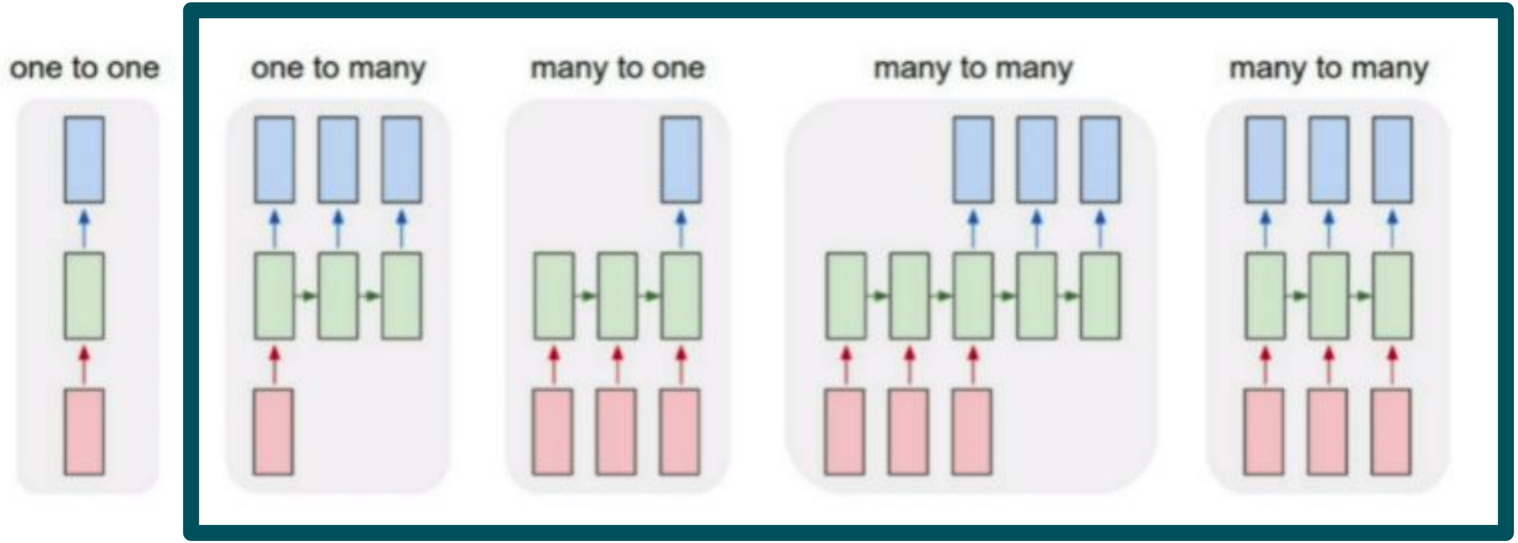


Sequence Problems



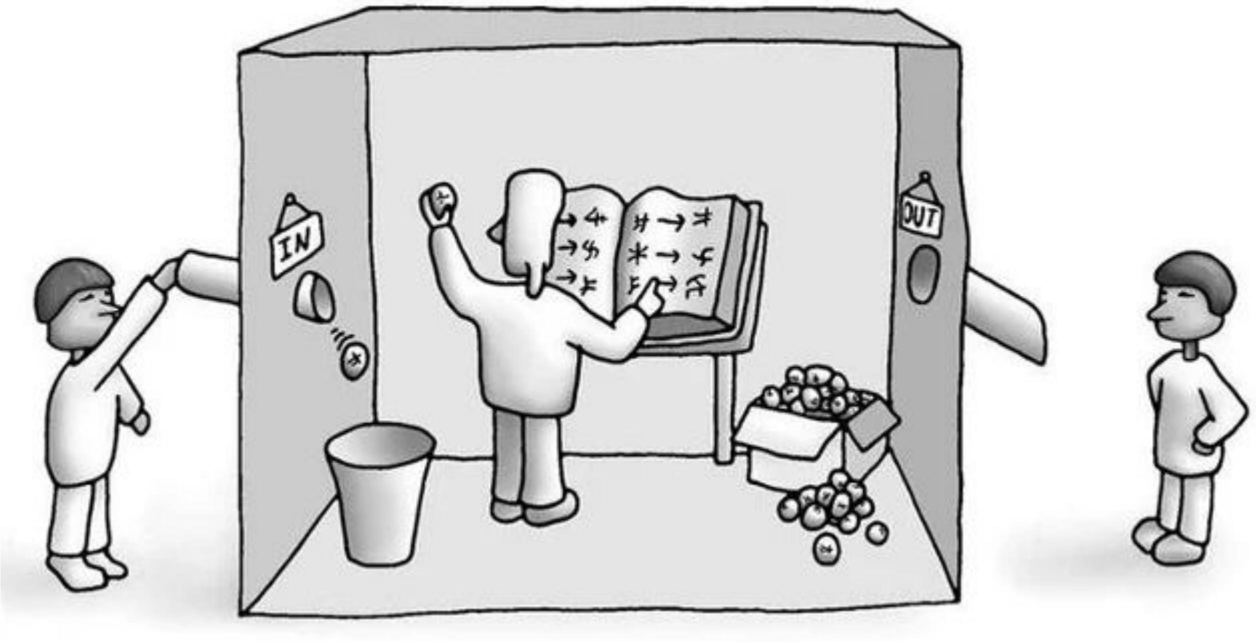
So far…

Sequence Problems

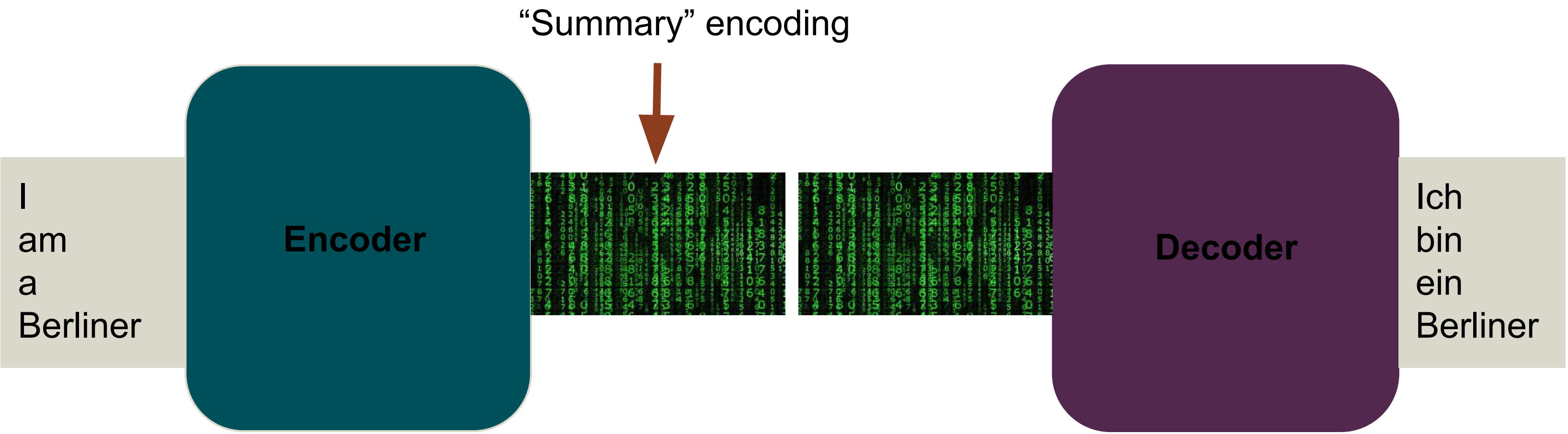


Today

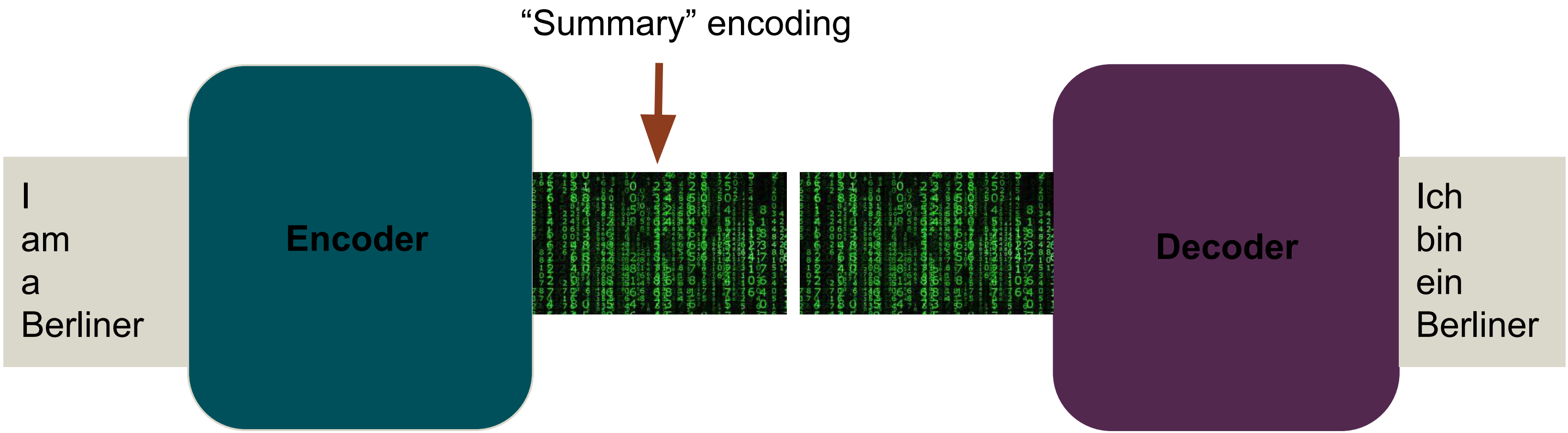
The Task: Machine Translation



The Paradigm: Encoder/Decoder



Recurrent Neural Networks



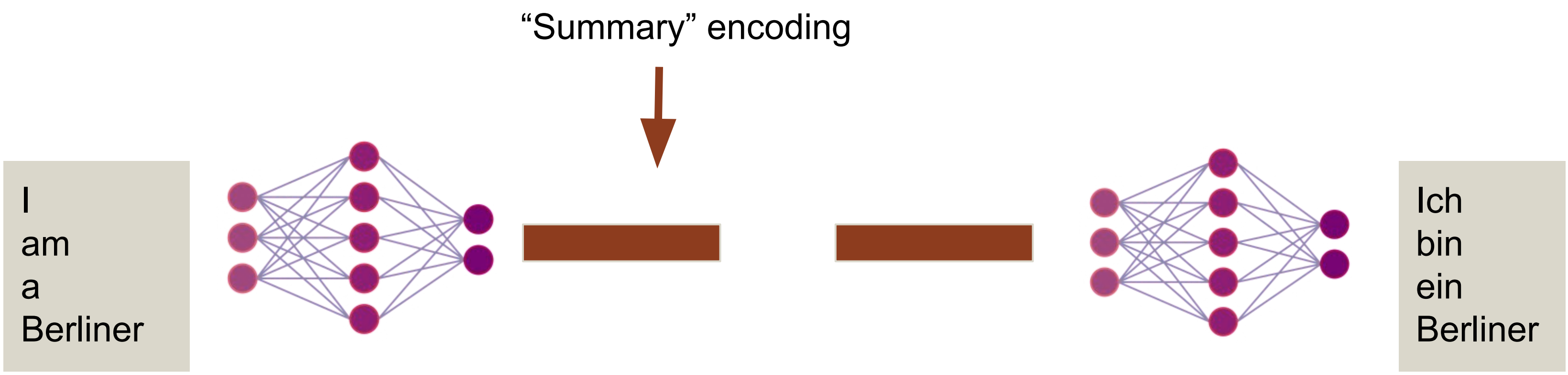
Recurrent Neural Networks

I
am
a
Berliner

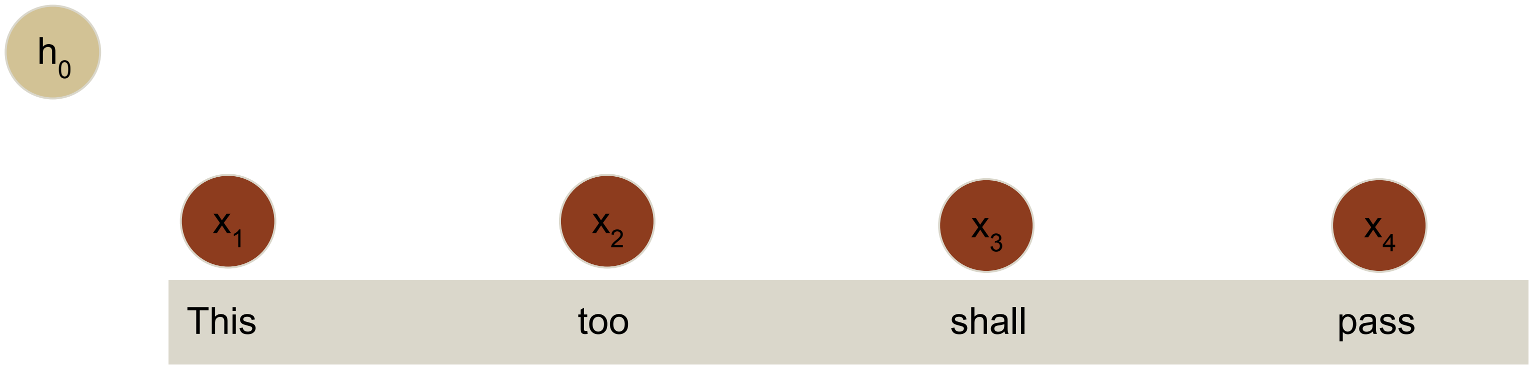
“Summary” encoding

Ich
bin
ein
Berliner

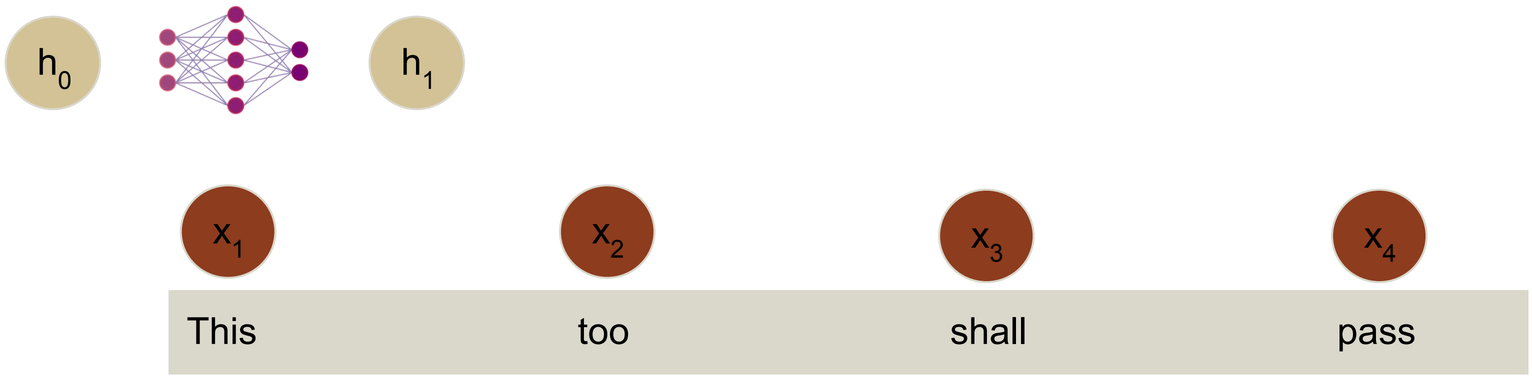
Recurrent Neural Networks



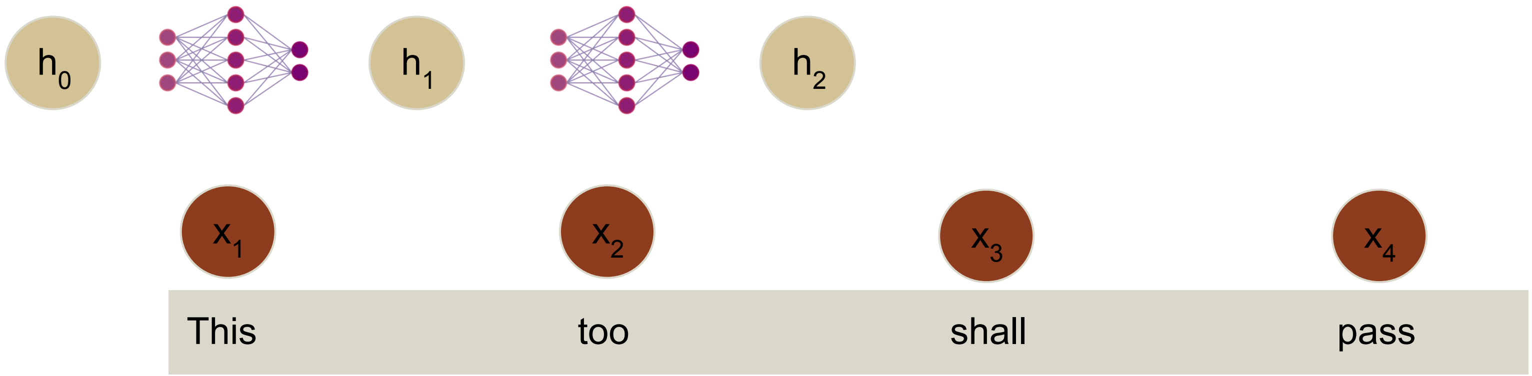
Recurrent Neural Networks: Encoder



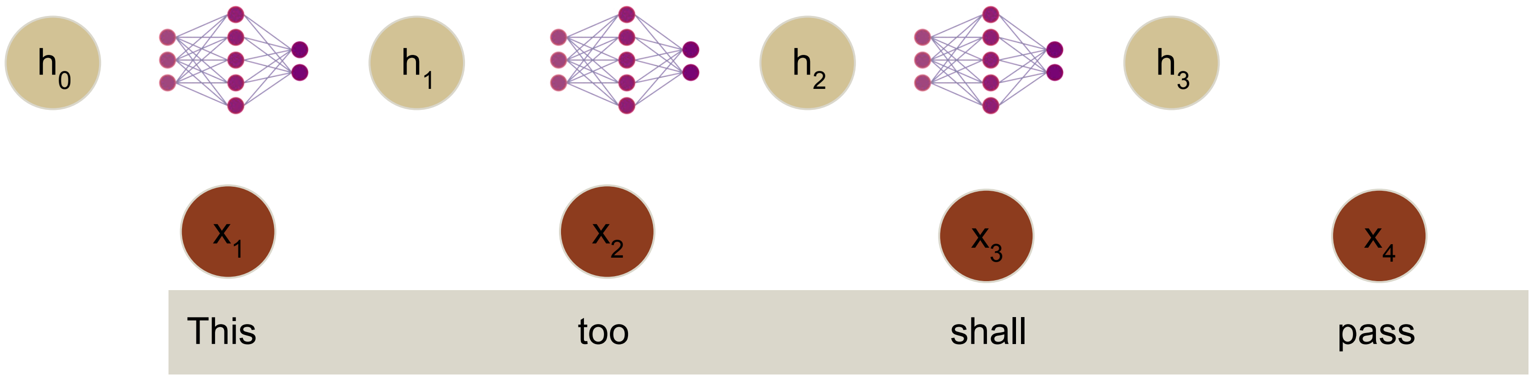
Recurrent Neural Networks: Encoder



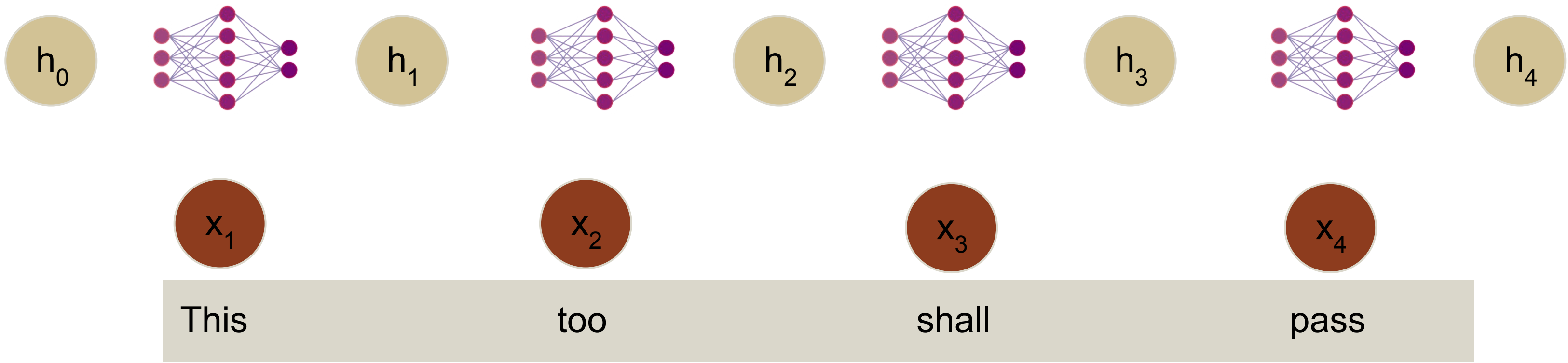
Recurrent Neural Networks: Encoder



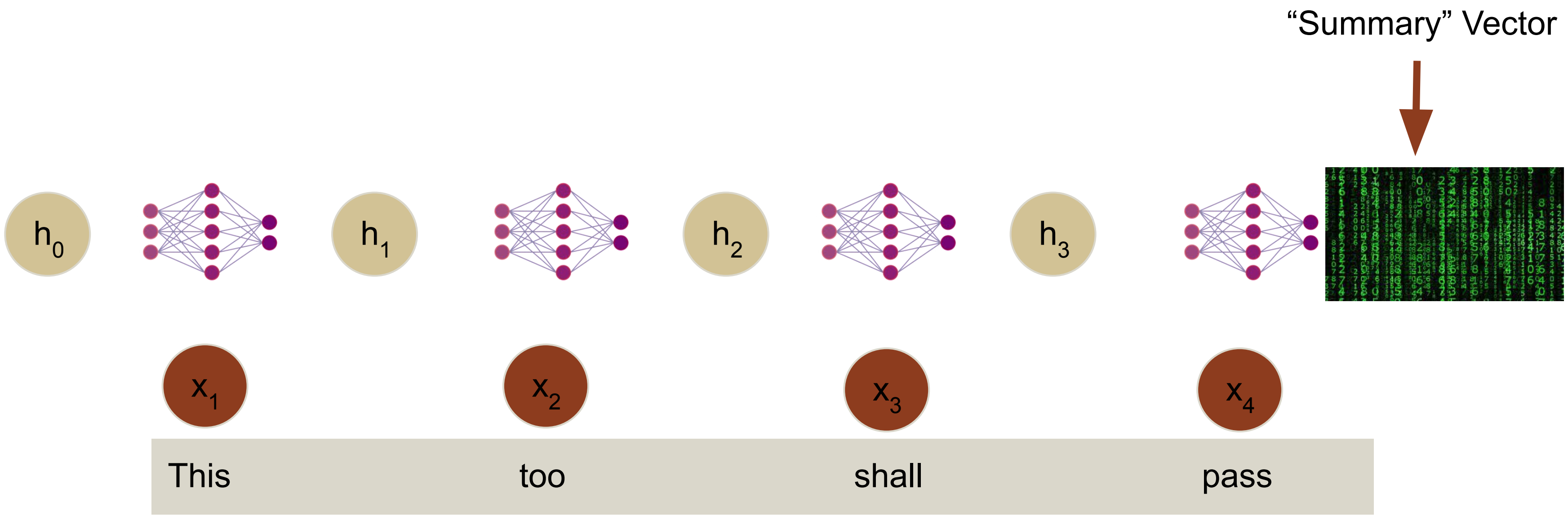
Recurrent Neural Networks: Encoder



Recurrent Neural Networks: Encoder

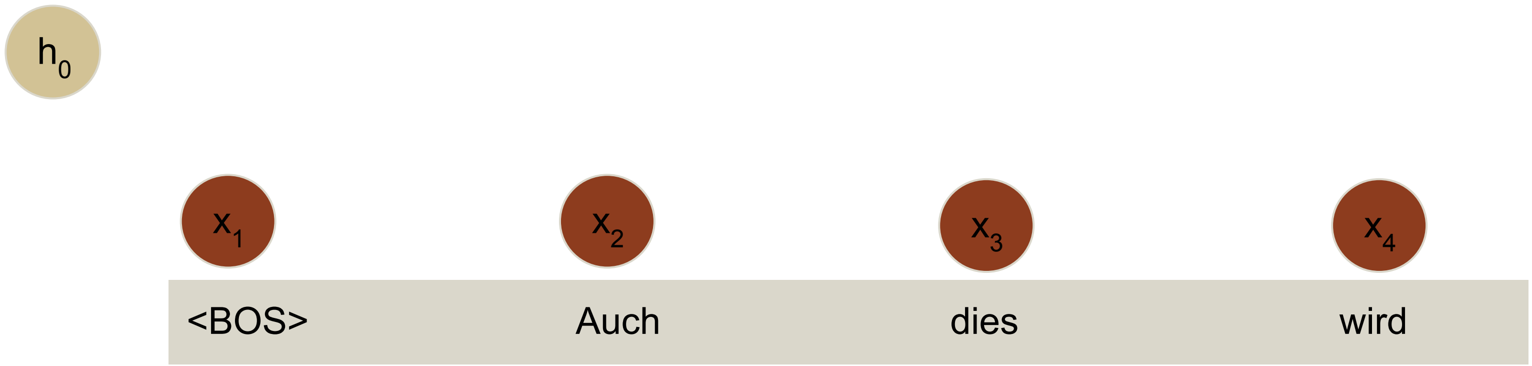


Recurrent Neural Networks: Encoder

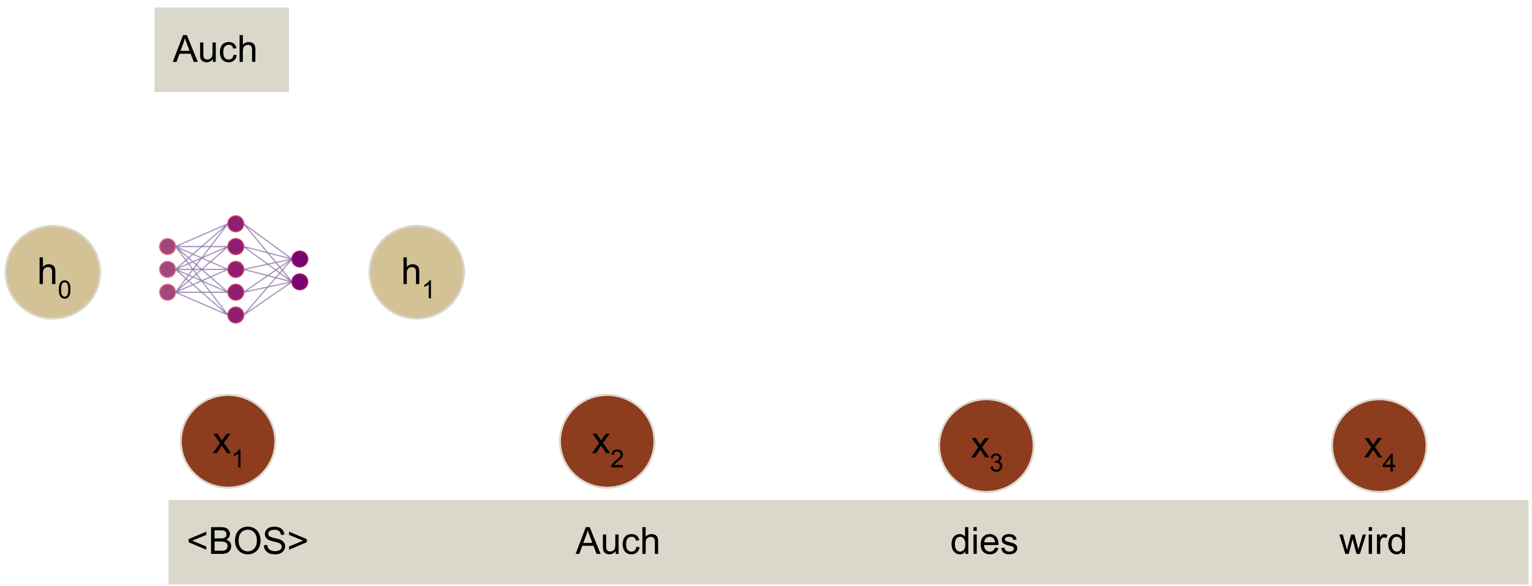


Recurrent Neural Networks: Decoder (Training Time)

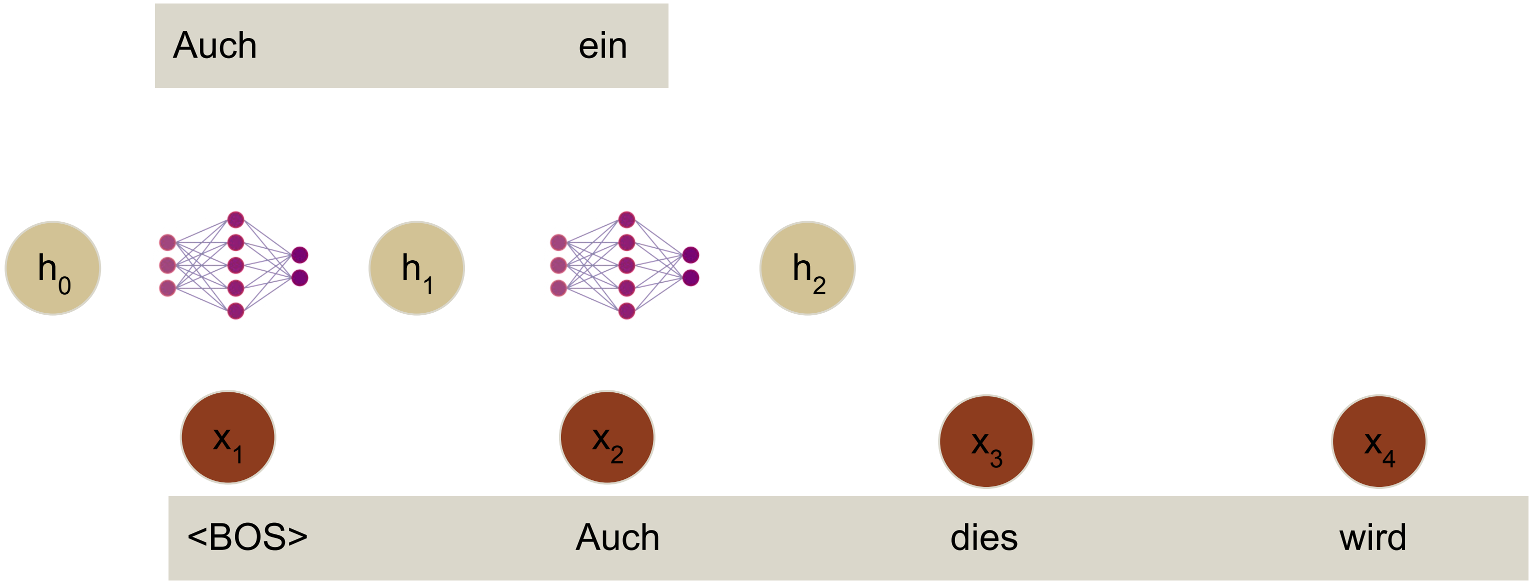
Recurrent Neural Networks: Decoder (Training Time)



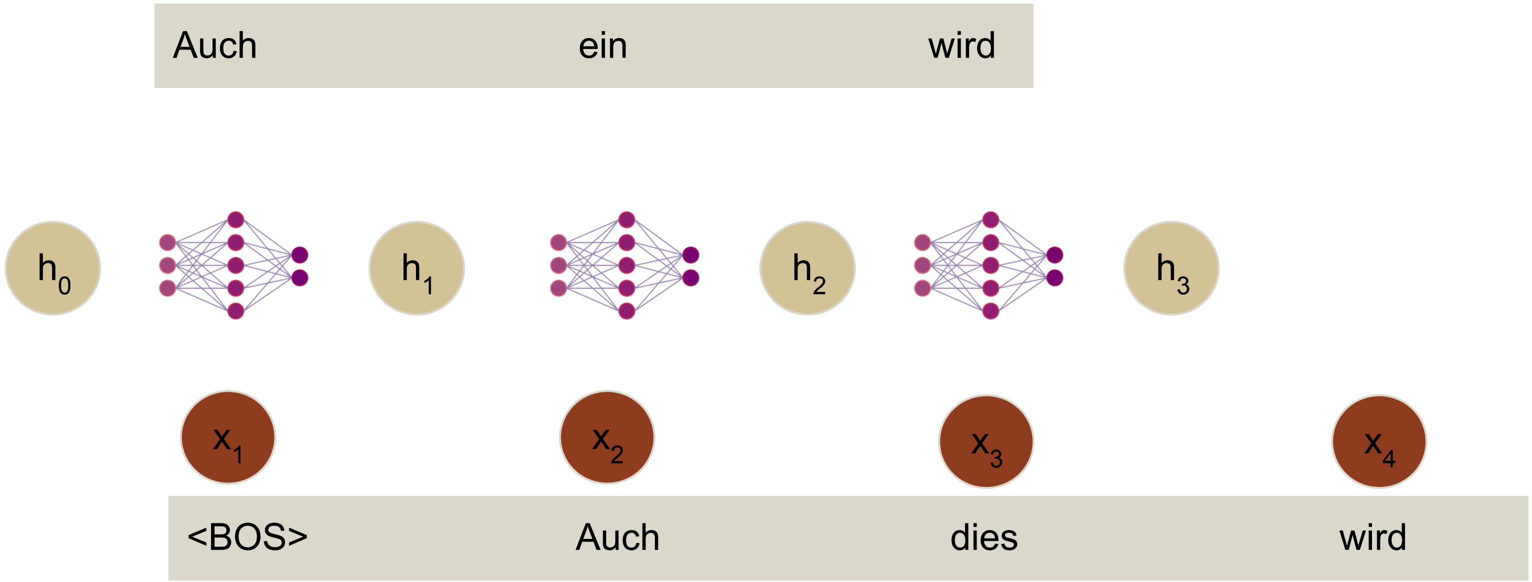
Recurrent Neural Networks: Decoder (Training Time)



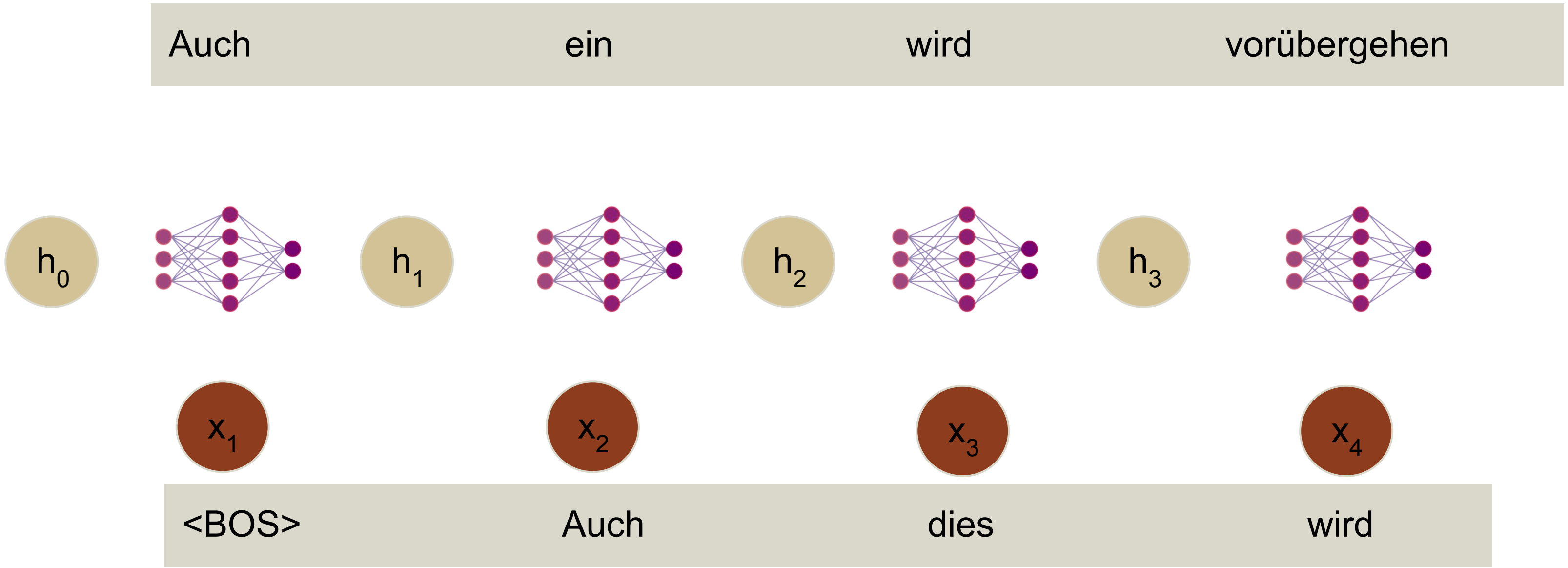
Recurrent Neural Networks: Decoder (Training Time)



Recurrent Neural Networks: Decoder (Training Time)



Recurrent Neural Networks: Decoder (Training Time)



Problems:

- Fundamental dependence of training time on length of sequence

Problems:

- Fundamental dependence of training time on length of sequence
- Vanishing/Exploding Gradients

Problems:

- Fundamental dependence of training time on length of sequence
- Vanishing/Exploding Gradients
- O(n) for words to ‘interact’

Desiderata:

Desiderata:

1. Low computational complexity per layer

Desiderata:

1. Low computational complexity per layer
2. Parallelizability

Desiderata:

1. Low computational complexity per layer
2. Parallelizability
3. Low path length between tokens

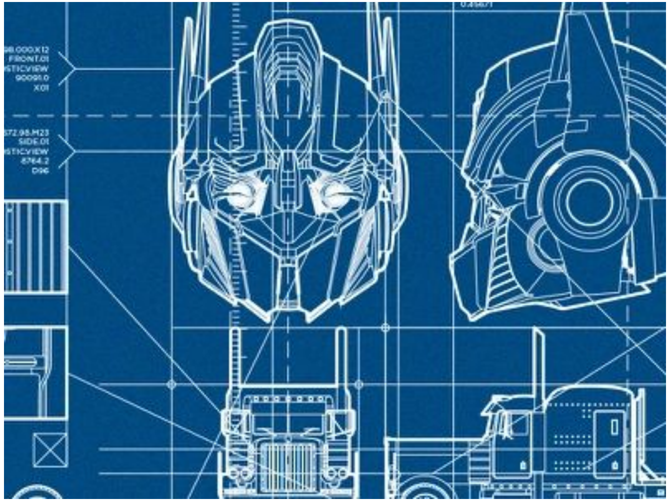
A sneak peak

Layer Type	Complexity per Layer	Sequential Operations	Maximum Path Length
Self-Attention	$O(n^2 \cdot d)$	$O(1)$	$O(1)$
Recurrent	$O(n \cdot d^2)$	$O(n)$	$O(n)$
Convolutional	$O(k \cdot n \cdot d^2)$	$O(1)$	$O(log_k(n))$

A sneak peak

Layer Type	Complexity per Layer	Sequential Operations	Maximum Path Length
Self-Attention	$O(n^2 \cdot d)$	$O(1)$	$O(1)$
Recurrent	$O(n \cdot d^2)$	$O(n)$	$O(n)$
Convolutional	$O(k \cdot n \cdot d^2)$	$O(1)$	$O(log_k(n))$

Architecture



Highest Level of Abstraction: Encoder/Decoder

Encoder

I
am
a
Berliner

Encoder

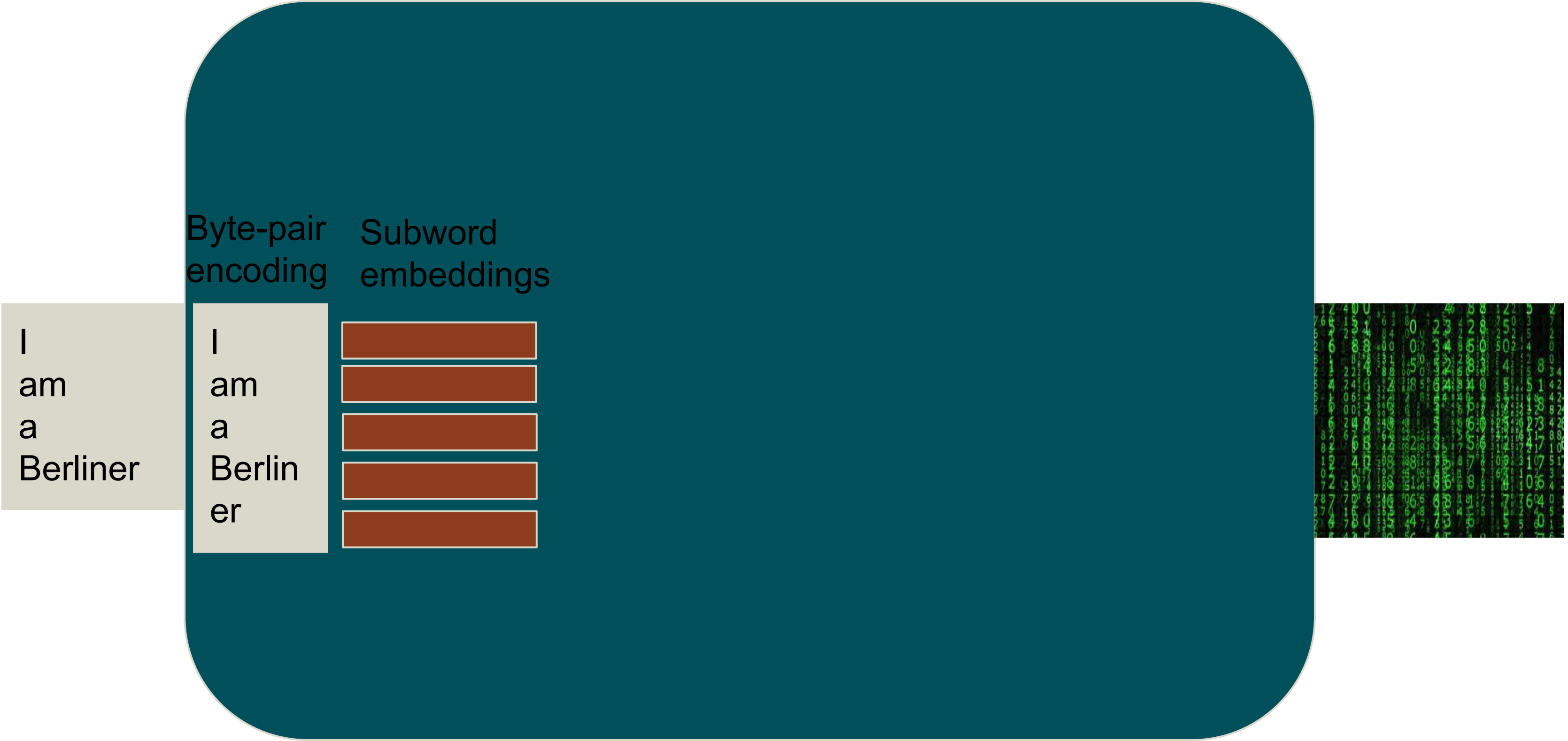
Encoder

Byte-pair encoding

I
am
a
Berliner

I
am
a
Berlin
er

Encoder


Byte-pair
encoding
Subword
embeddings
I
am
a
Berliner
I
am
a
Berlin
er

Encoder

Byte-pair encoding

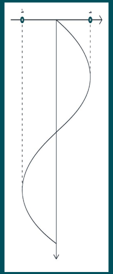
Subword embeddings

Positional embeddings

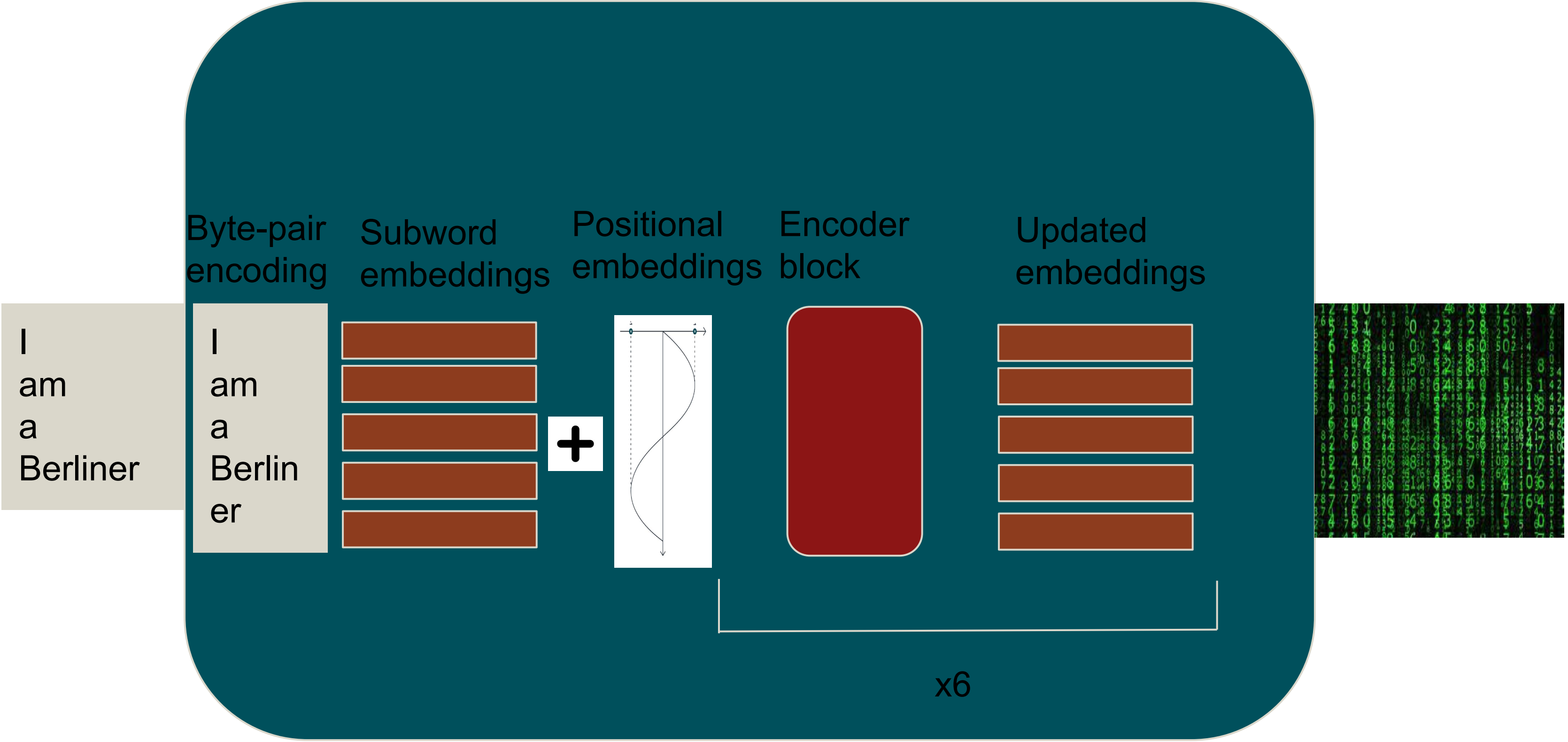
I
am
a
Berliner

I
am
a
Berlin
er

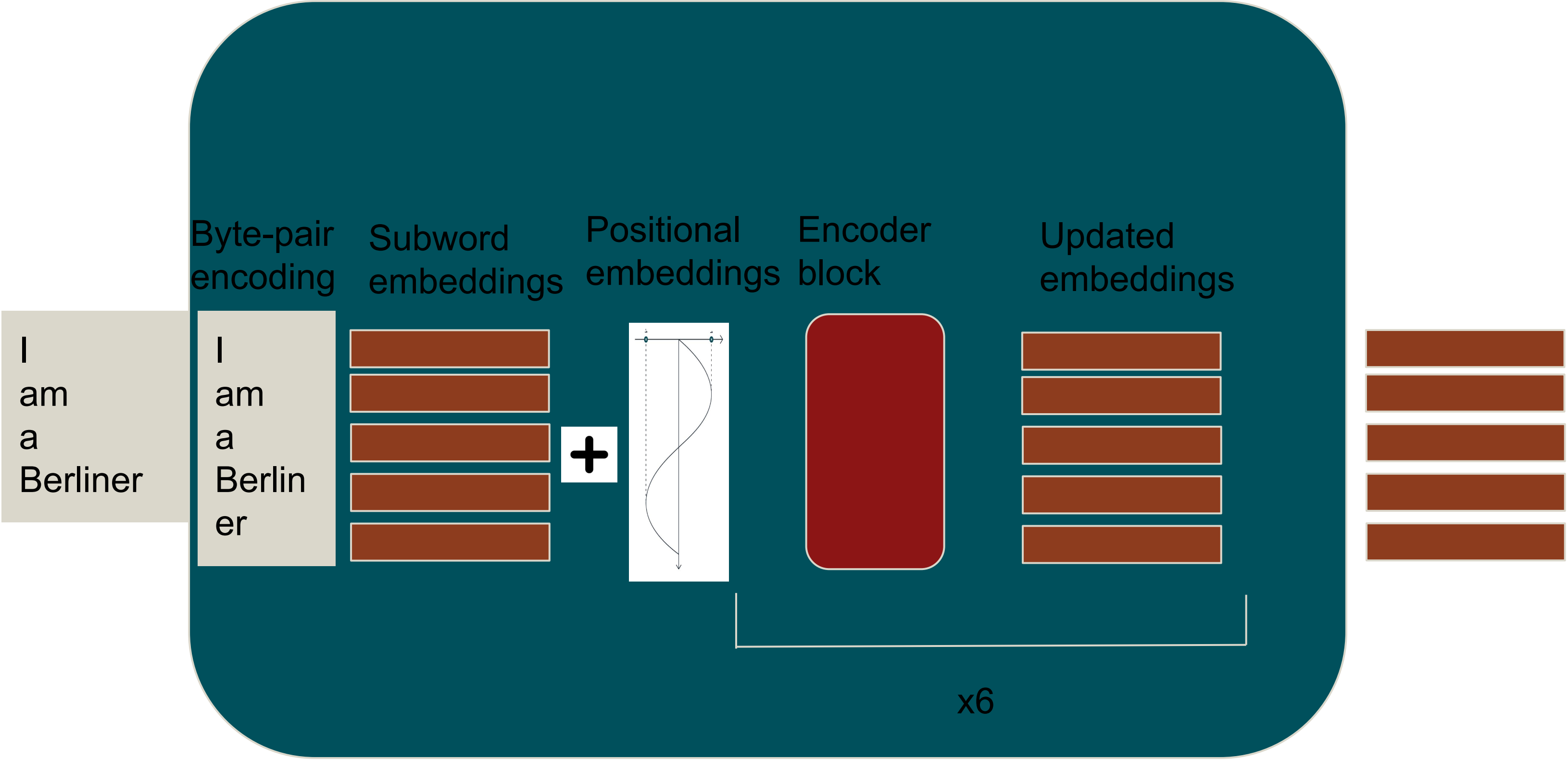
+



Encoder

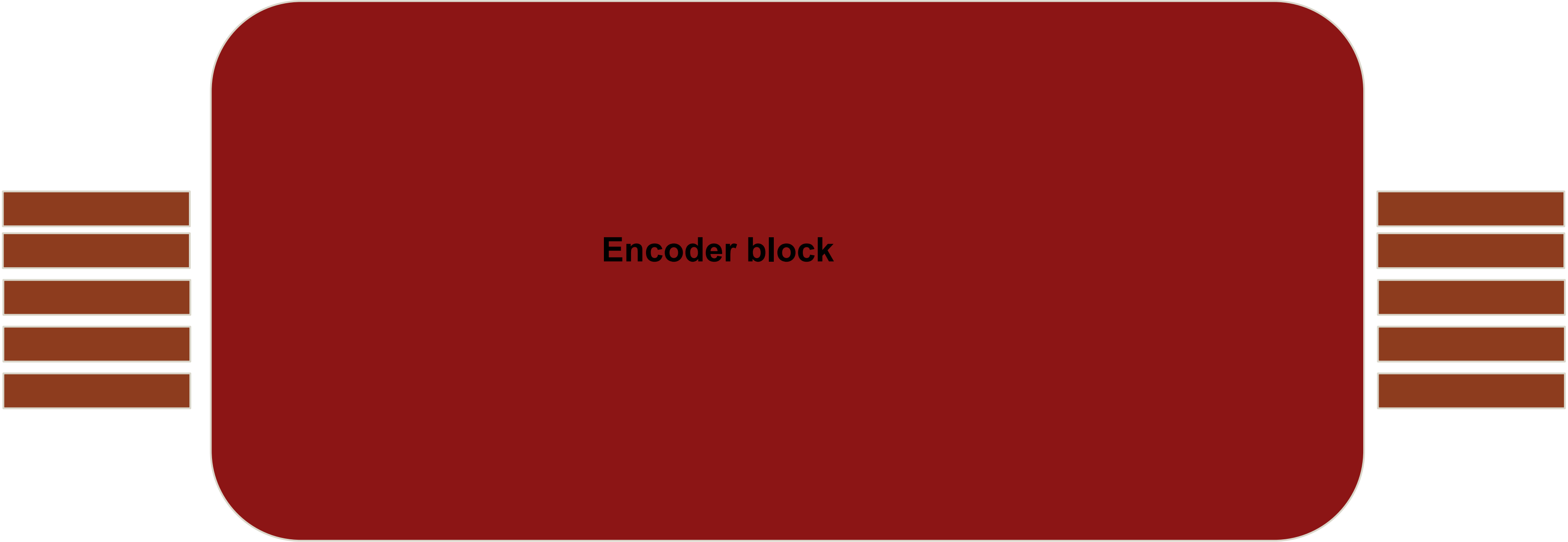

I
am
a
Berliner
Byte-pair encoding
I
am
a
Berlin
er
Subword embeddings
+
Positional embeddings
Encoder block
Updated embeddings
x6

Encoder

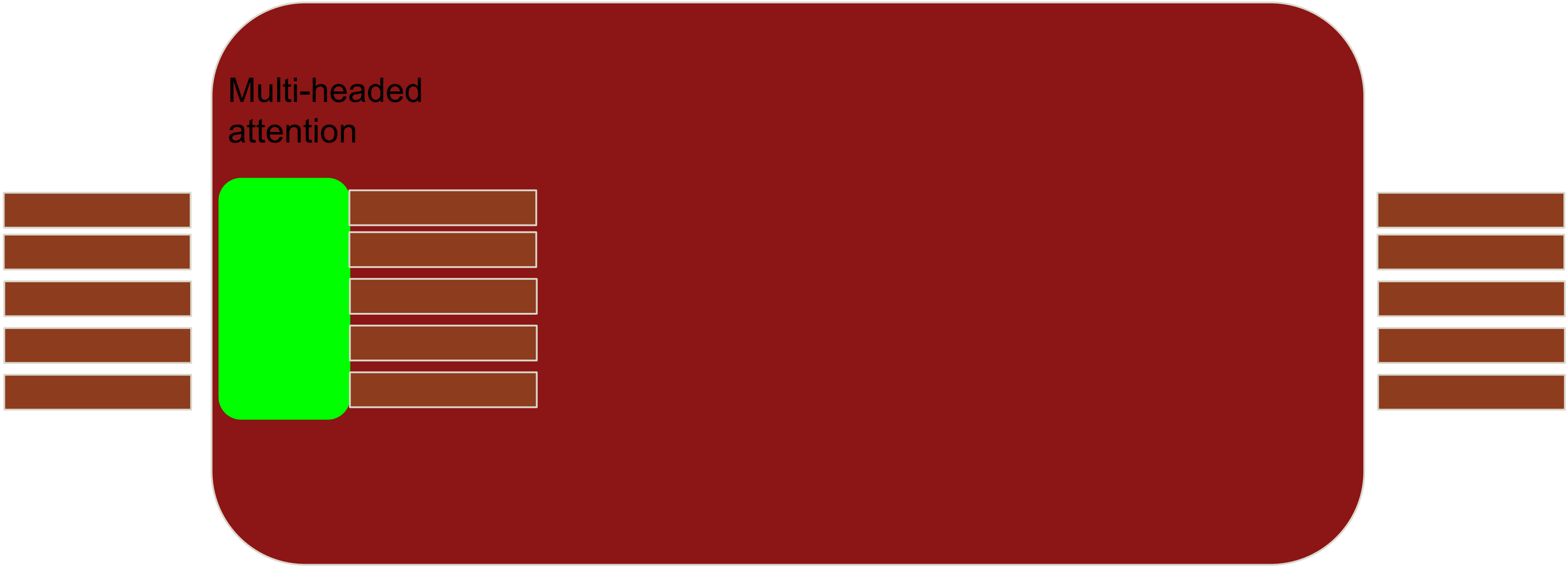


I
am
a
Berliner
Byte-pair
encoding
I
am
a
Berlin
er
Subword
embeddings
+
Positional
embeddings
Encoder
block
Updated
embeddings
x6

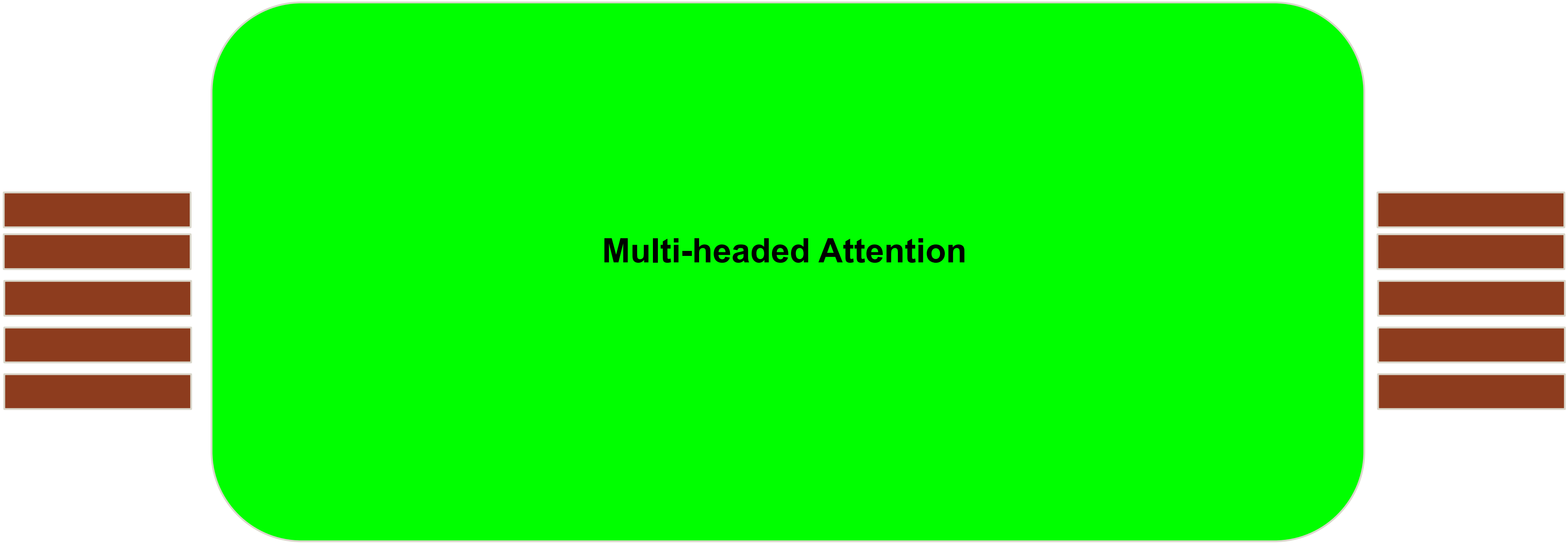
Encoder Block



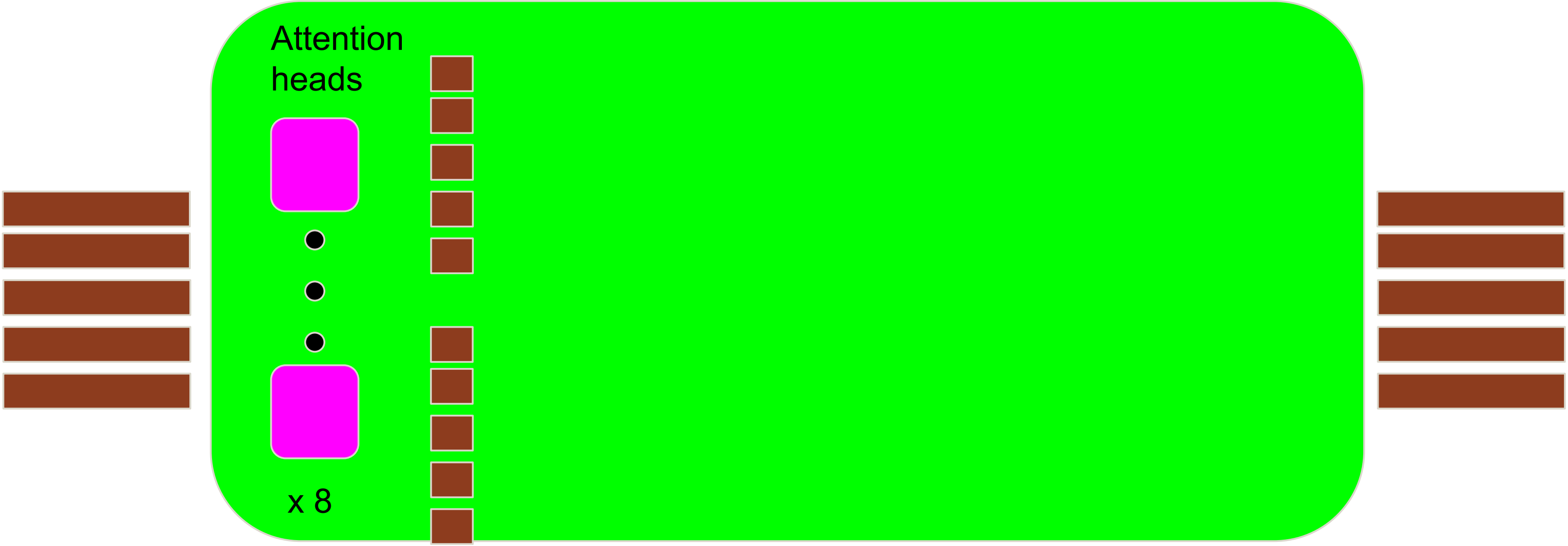
Encoder Block



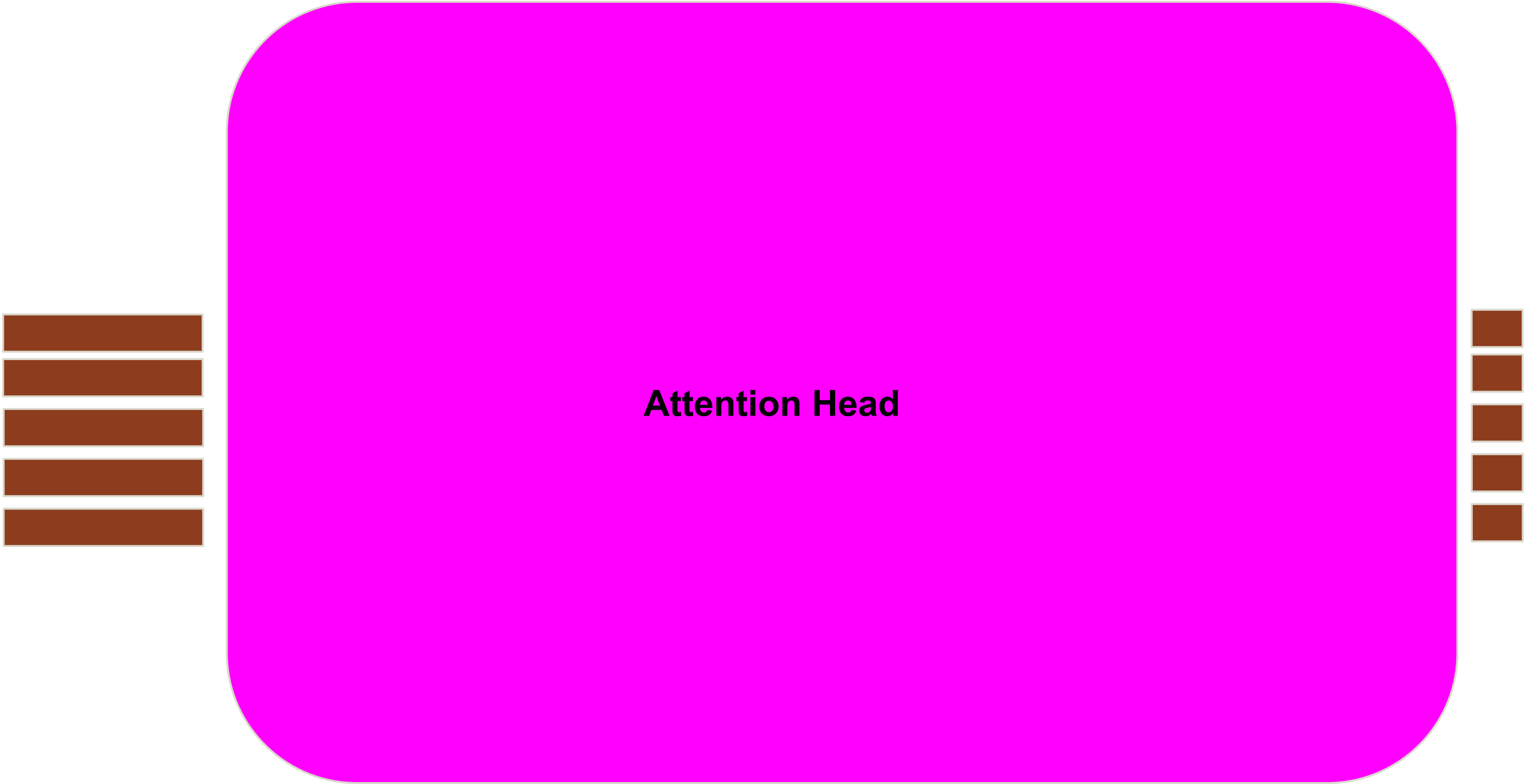
Multi-headed Attention Layer



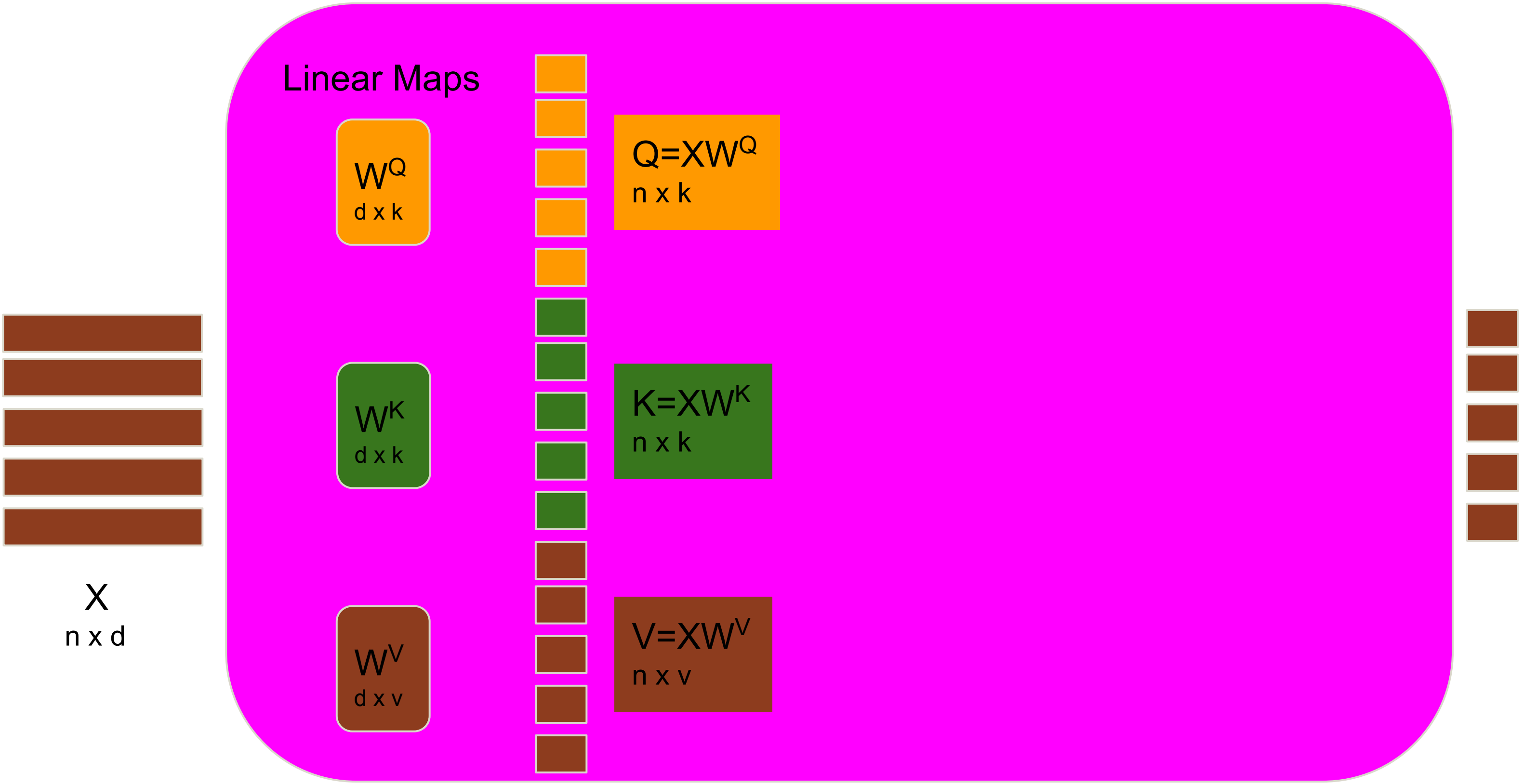
Multi-headed Attention Layer



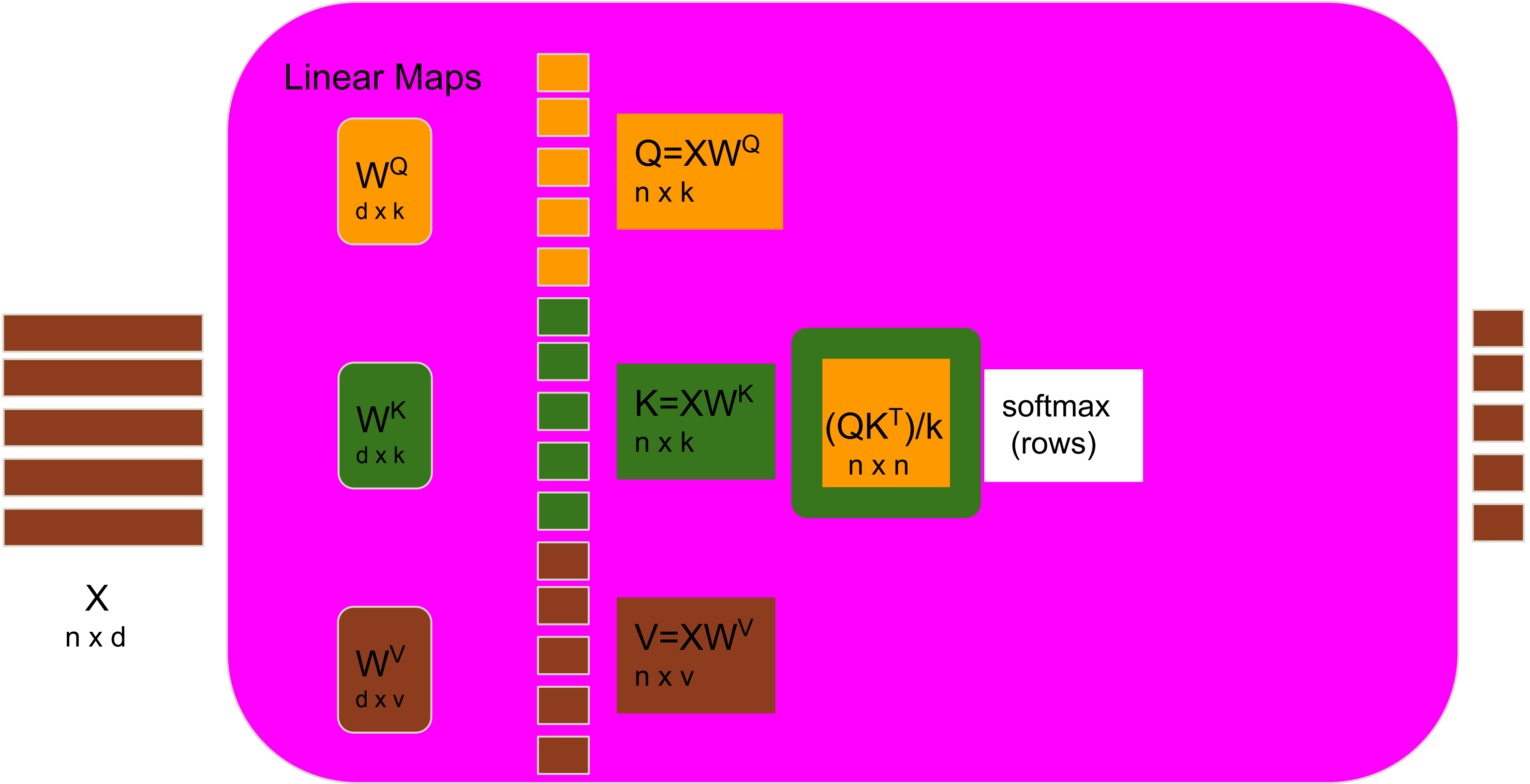
Attention Head



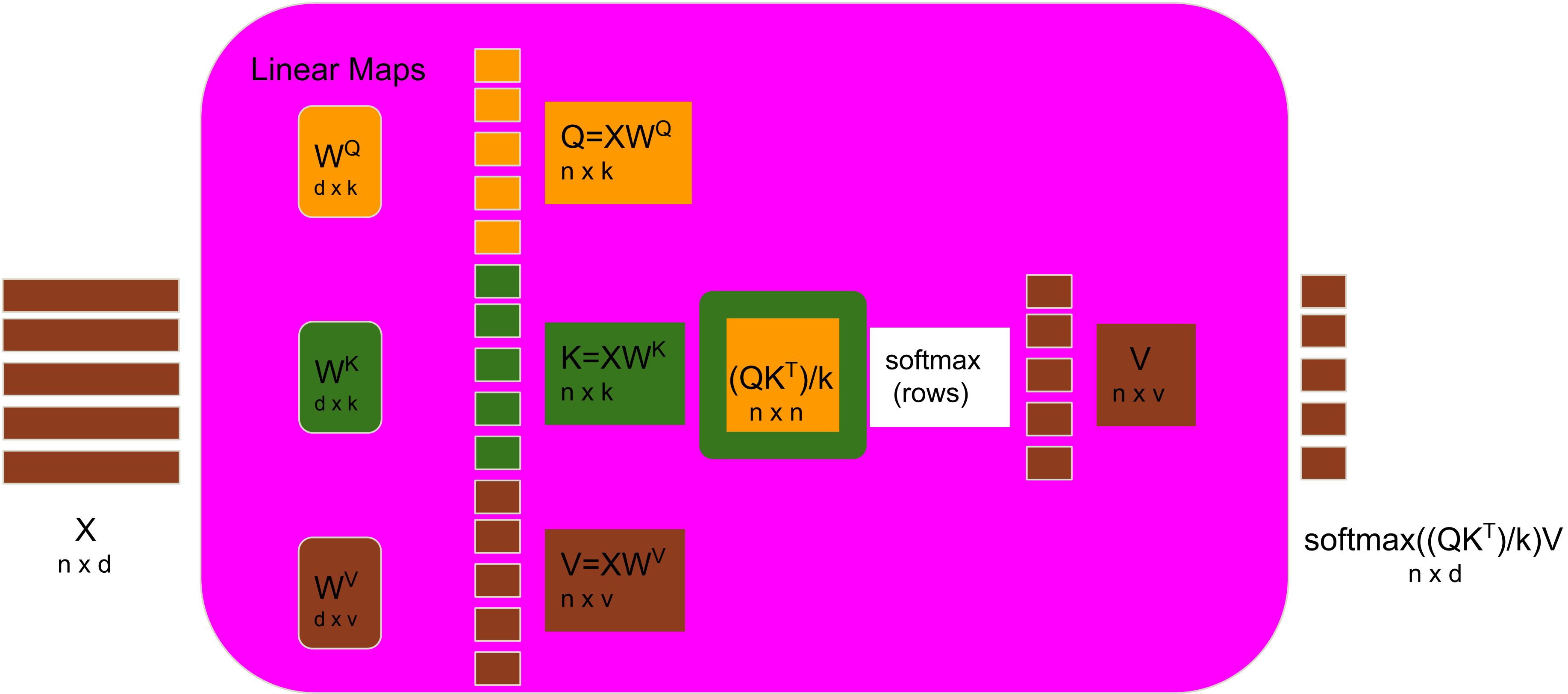
Attention Head



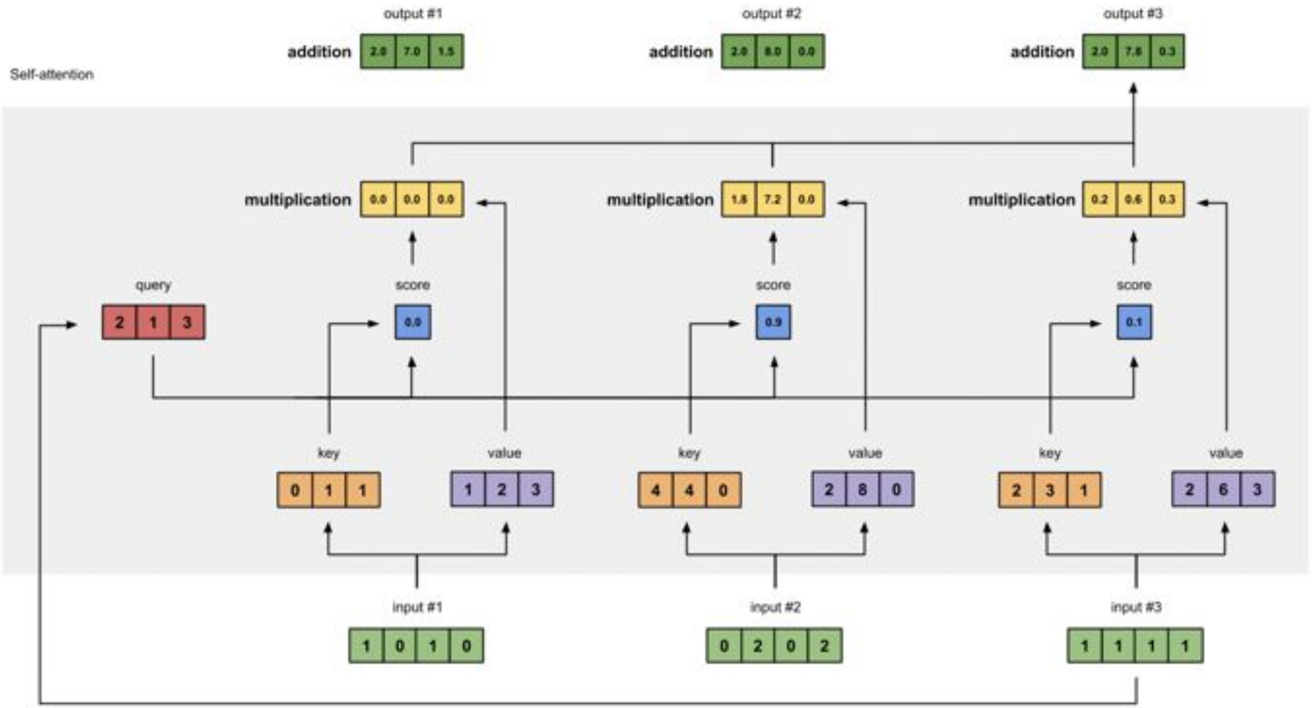
Attention Head



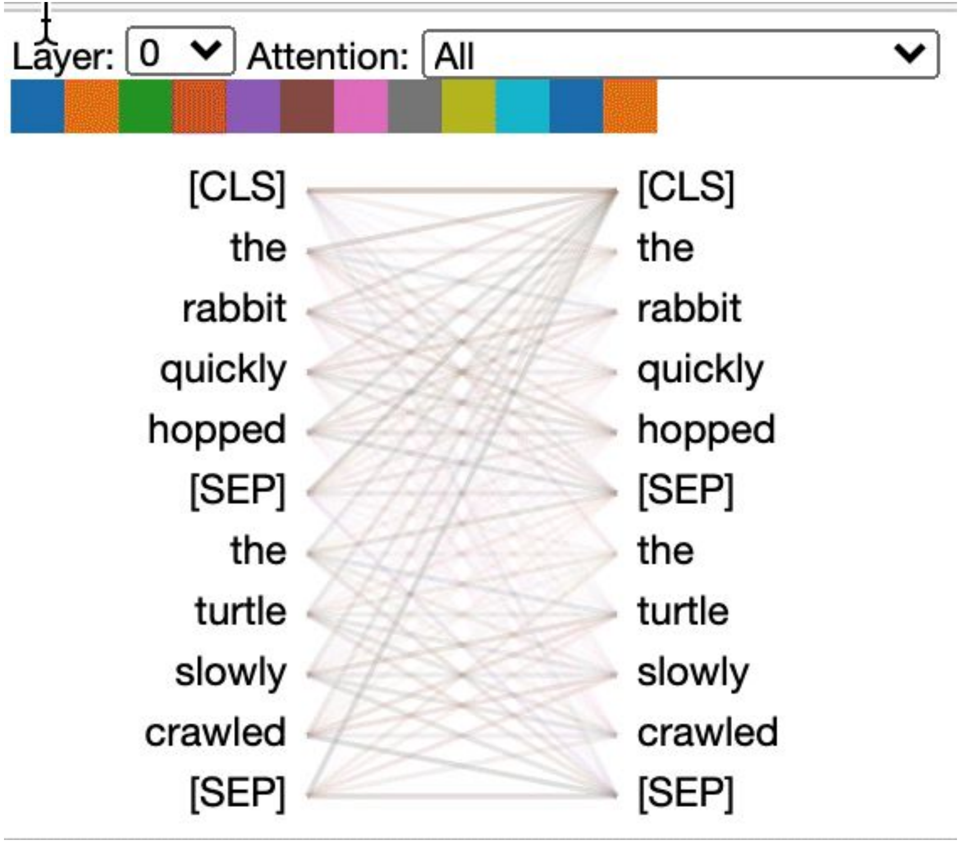
Attention Head



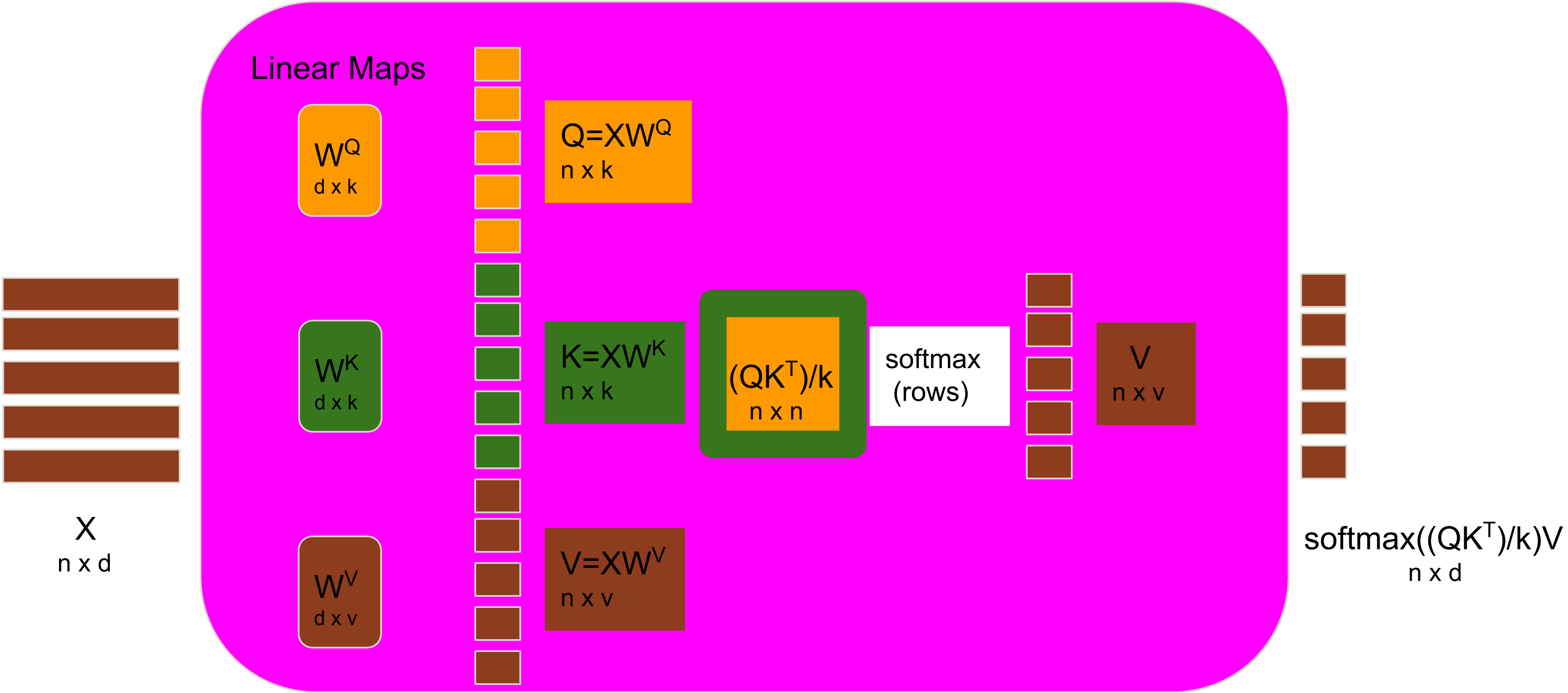
Attention Head, POV a single embedding



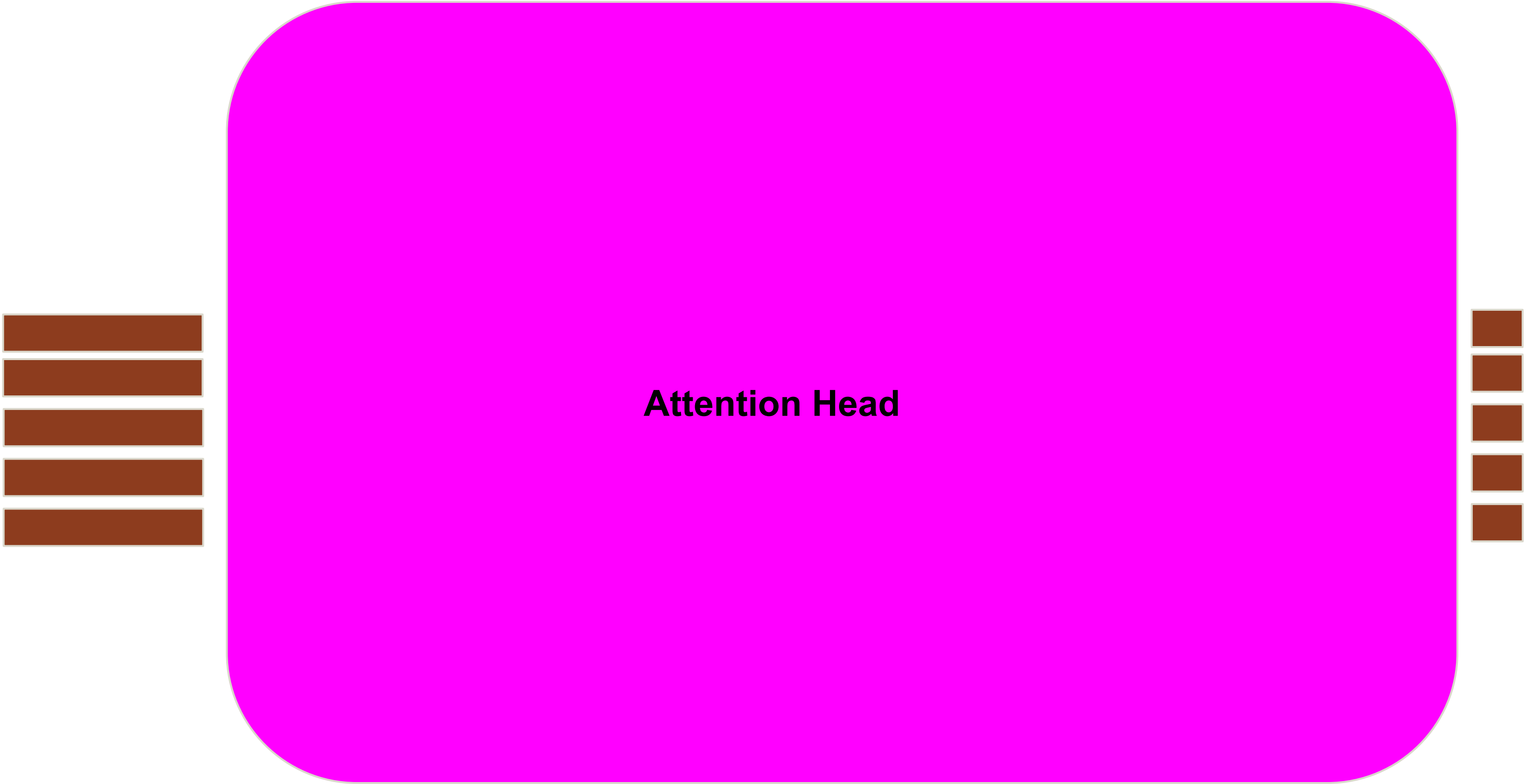
Similarity Scores Visualized



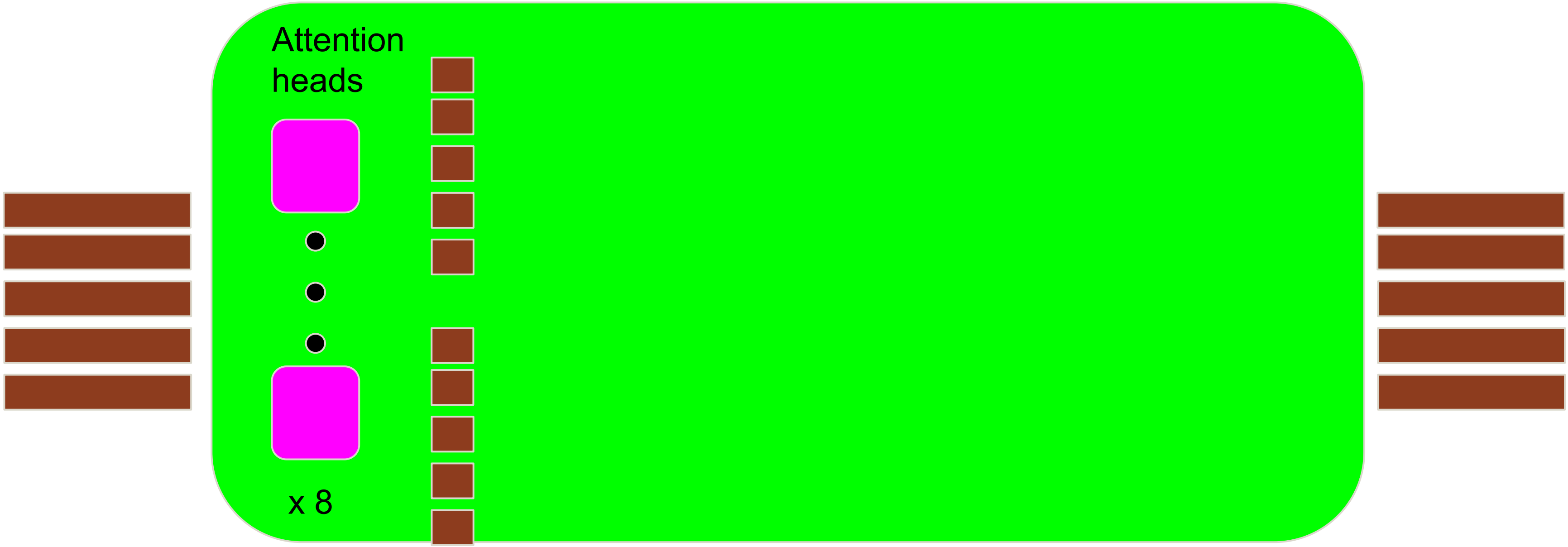
Attention Head



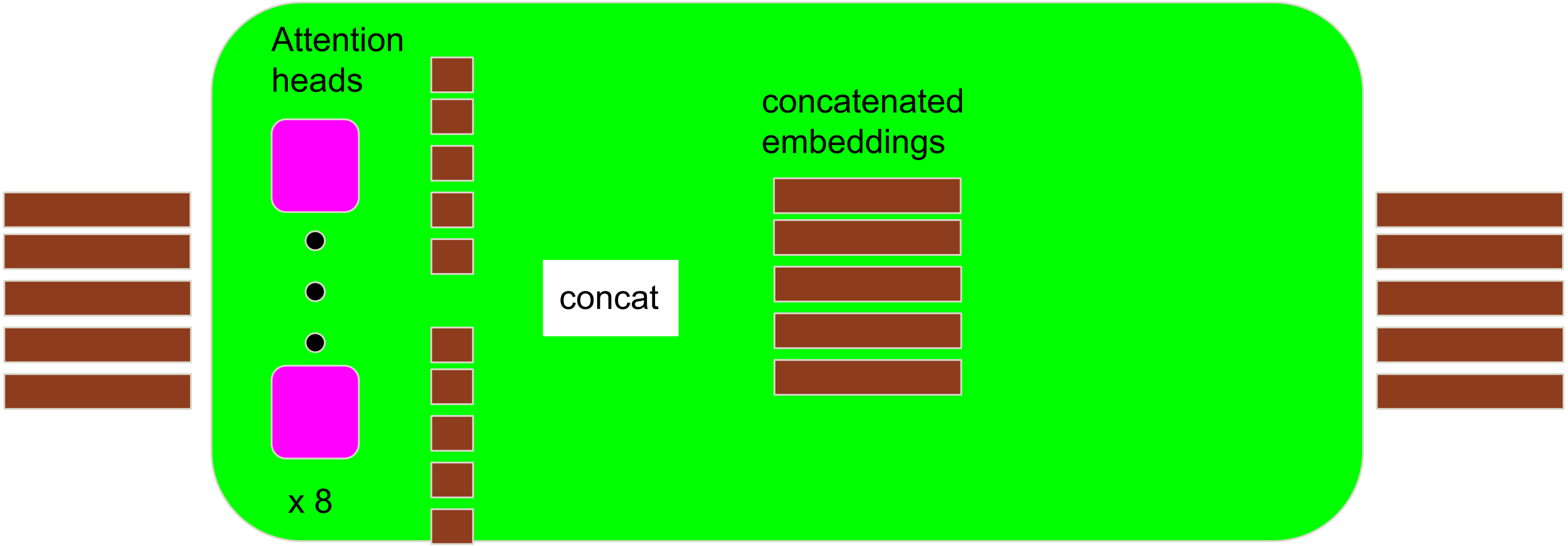
Attention Head



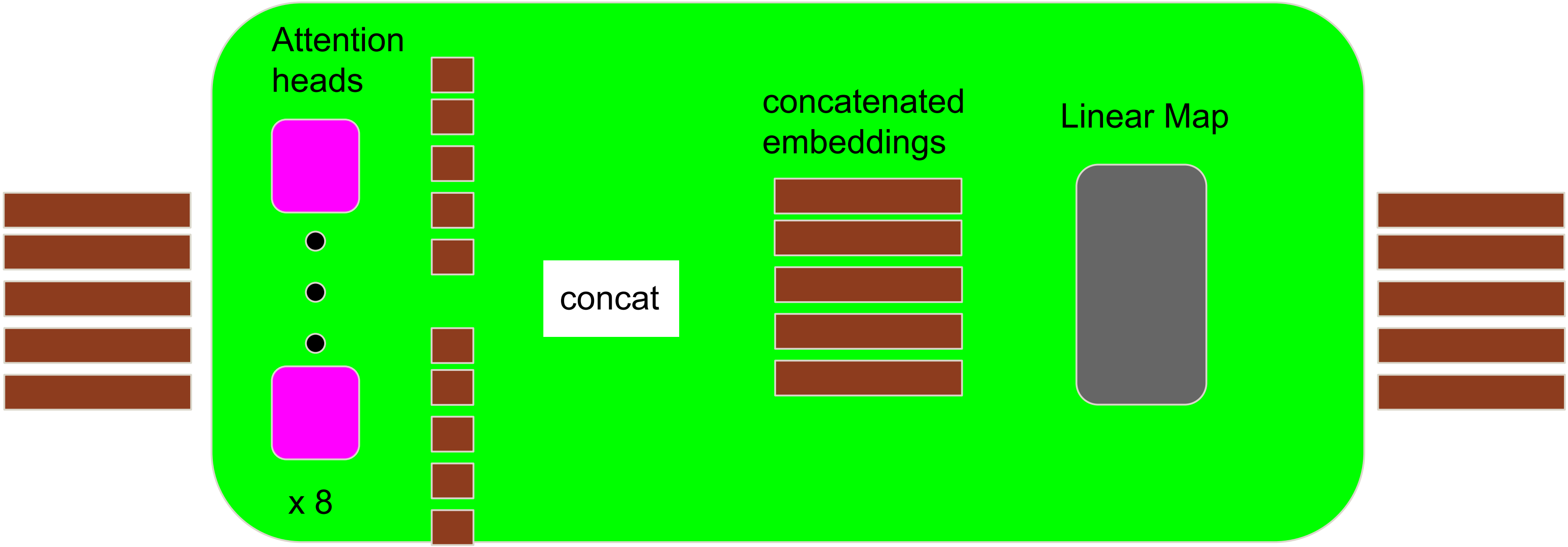
Multi-headed Attention Layer



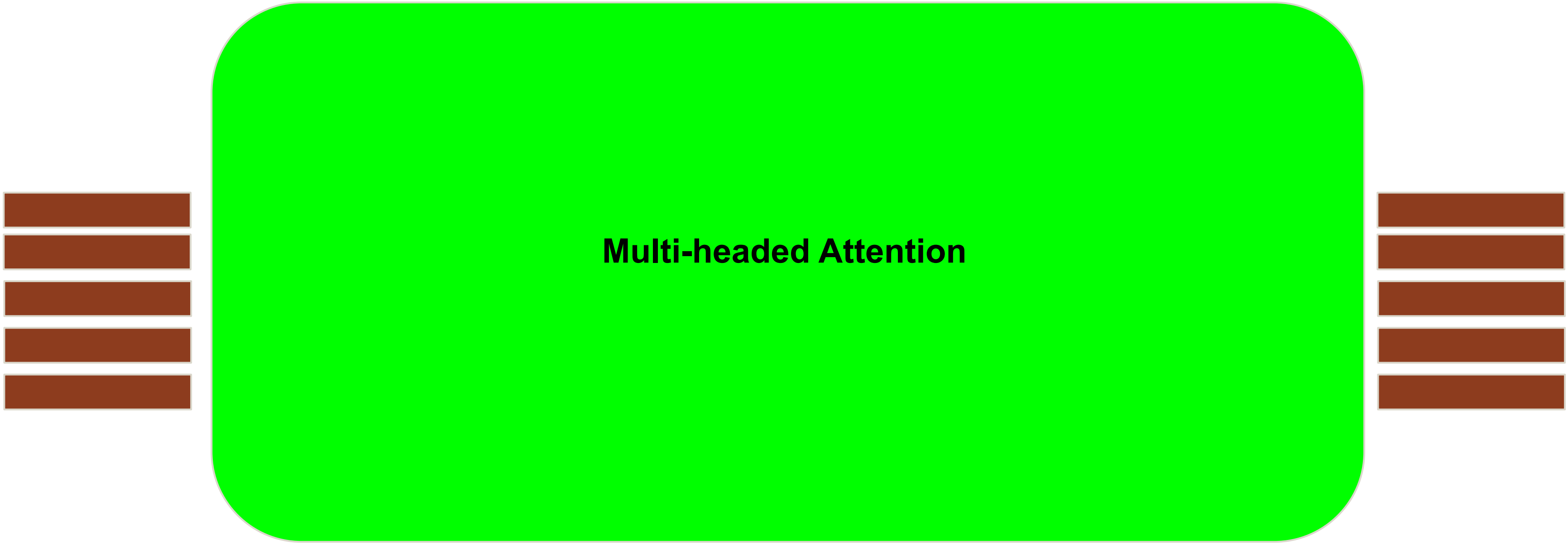
Multi-headed Attention Layer



Multi-headed Attention Layer

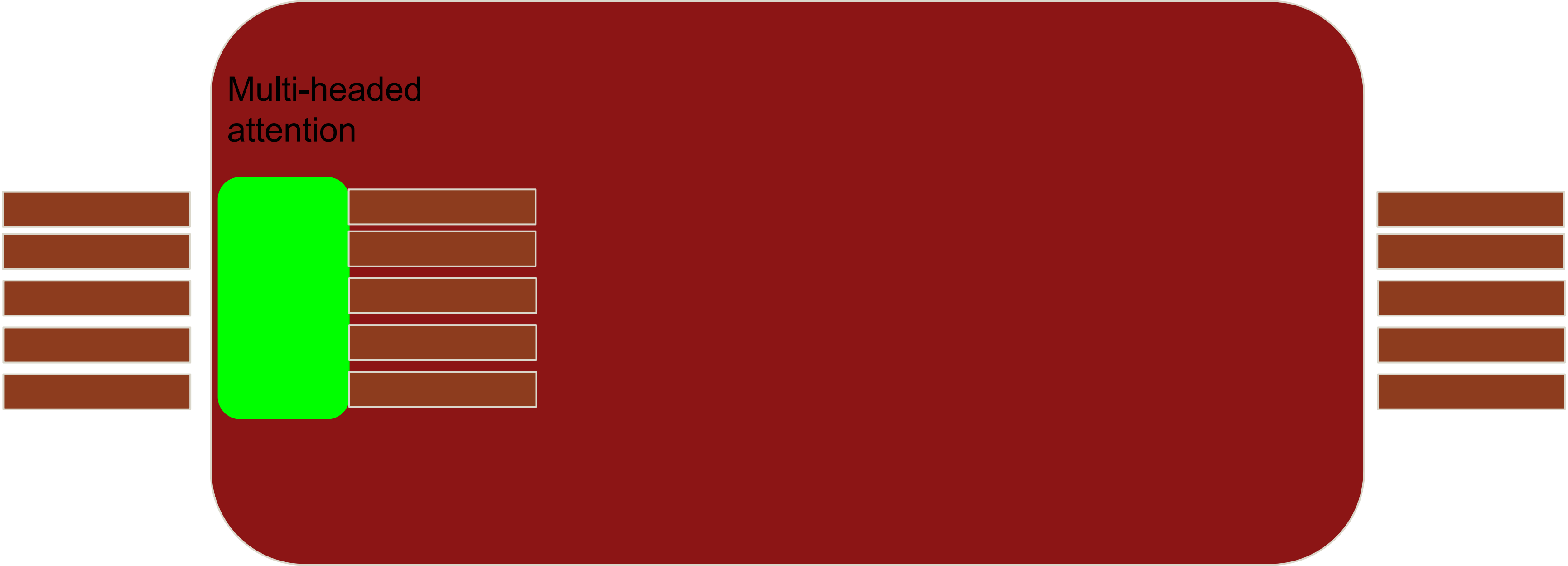


Multi-headed Attention Layer

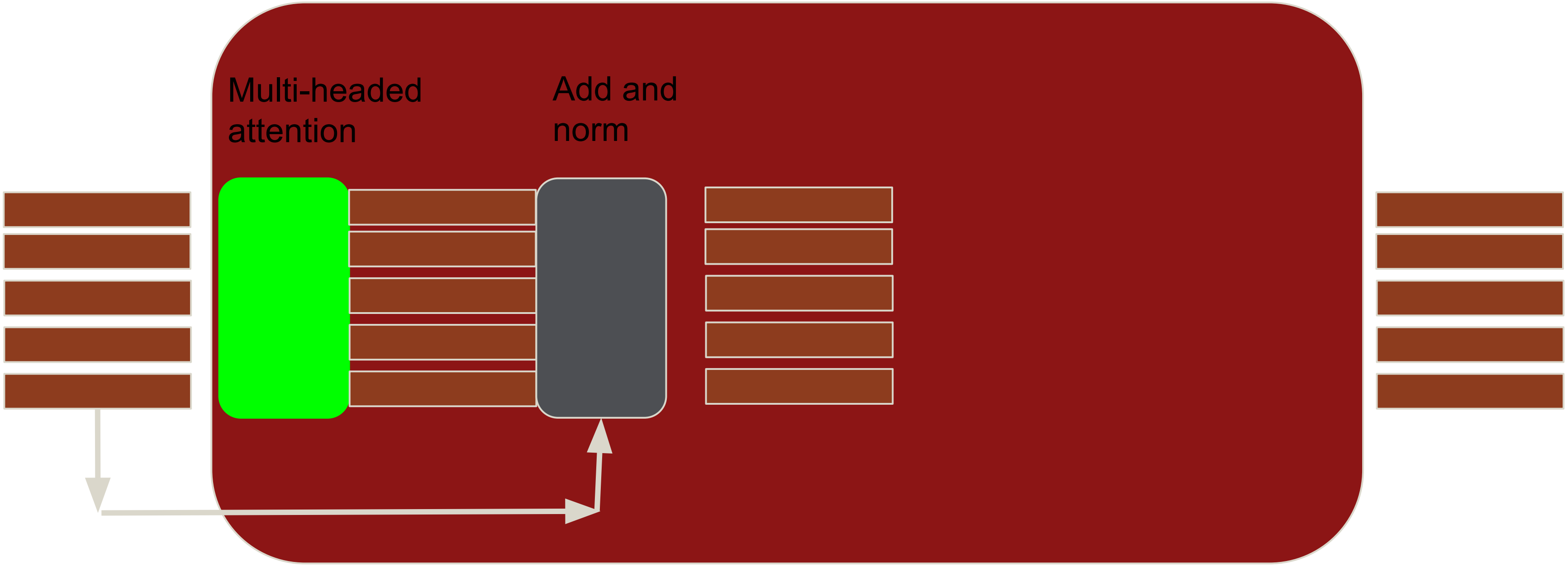


Stanford University

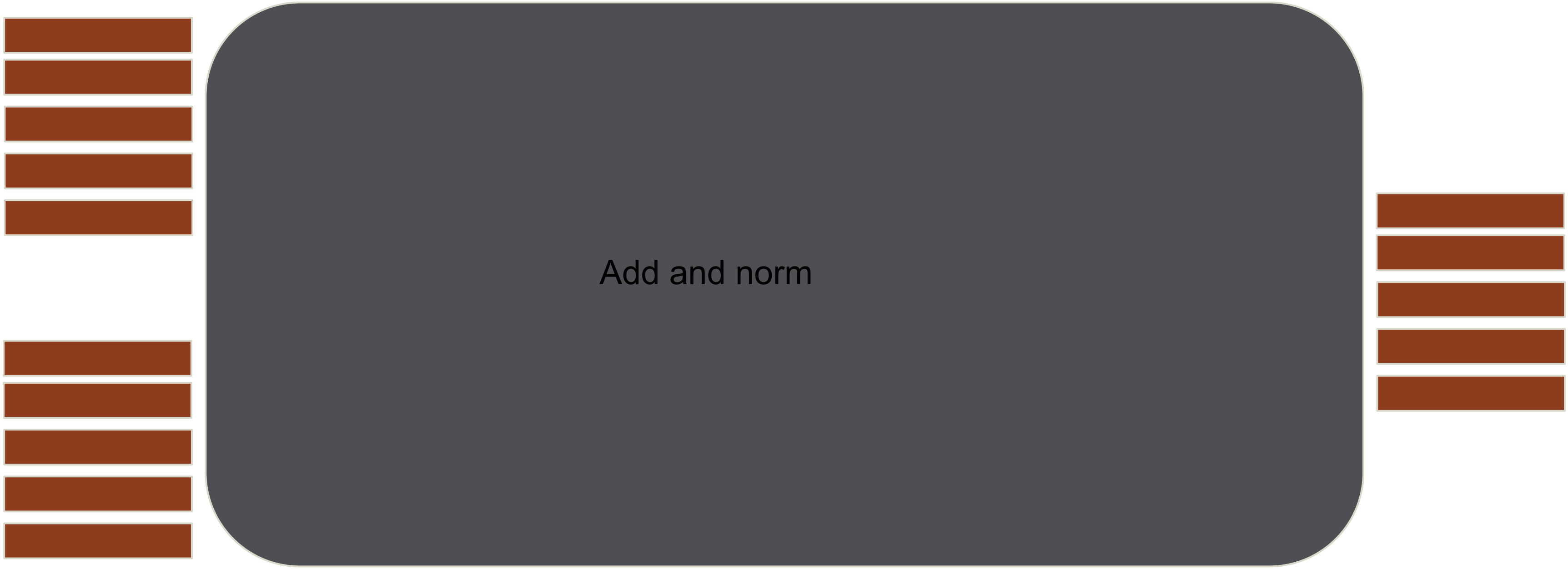
Encoder Block



Encoder Block

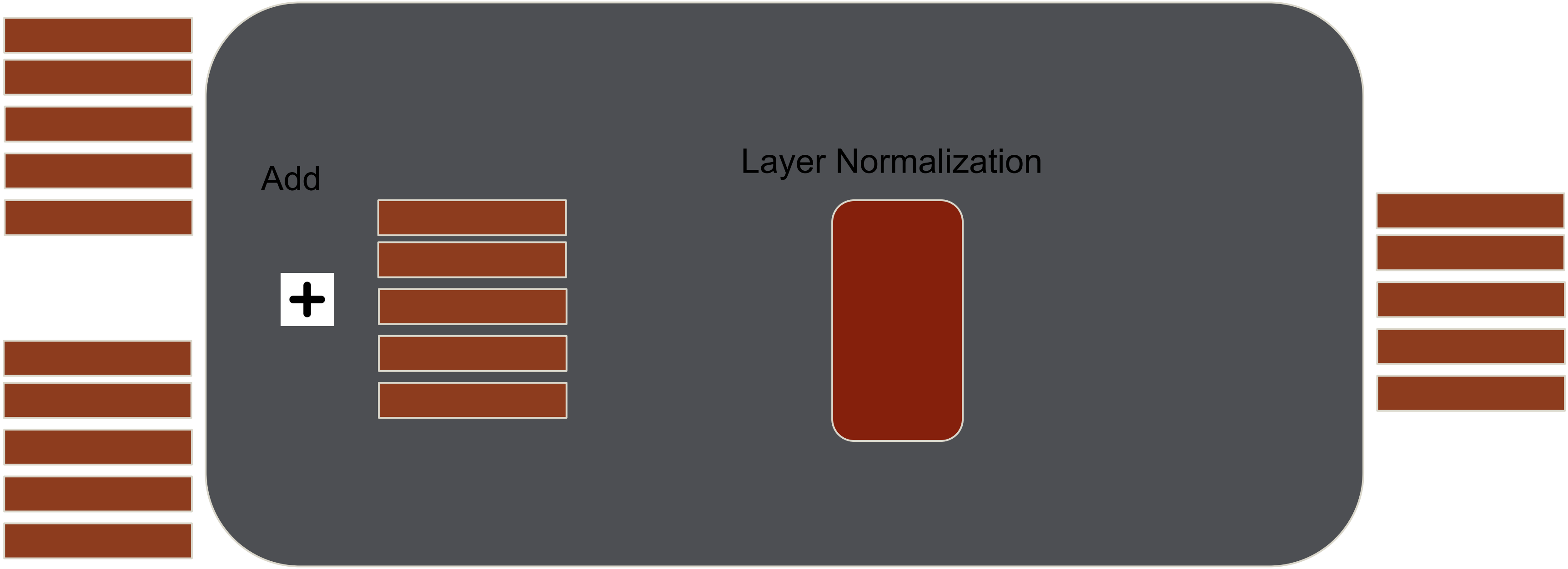


Add and Norm Layer



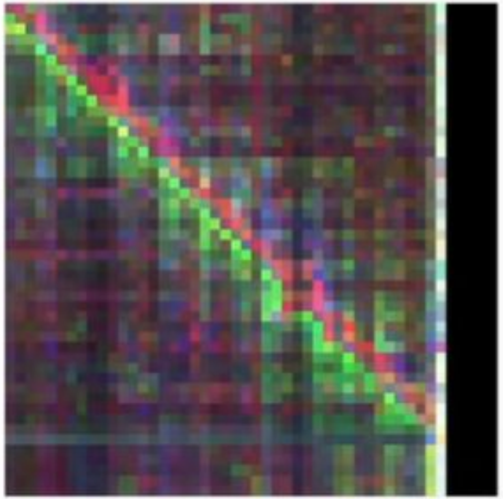
Add and Norm Layer

Add and Norm Layer

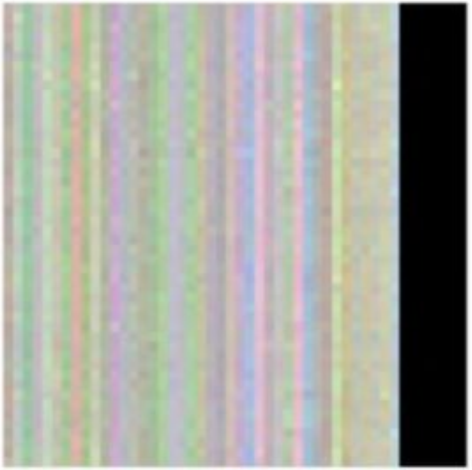


What do the Residual Layers do?

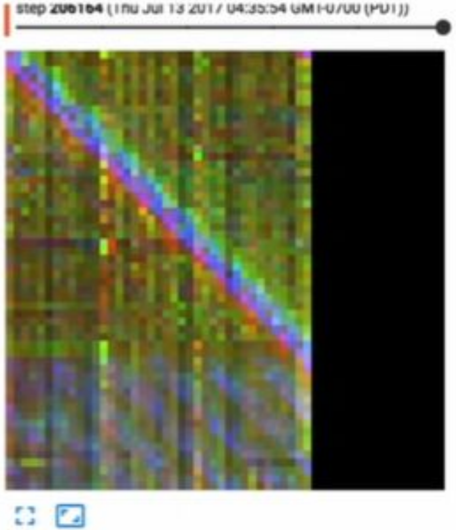
Residuals carry positional information to higher layers, among other information.



With residuals

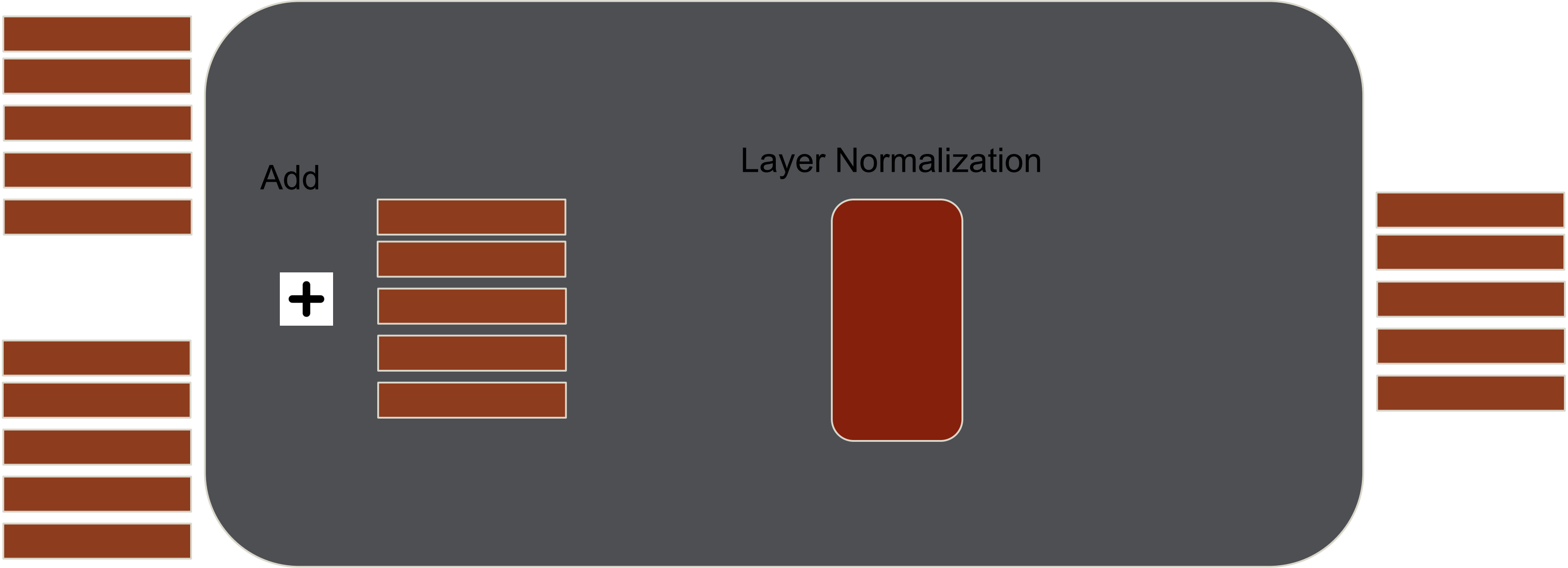


Without residuals

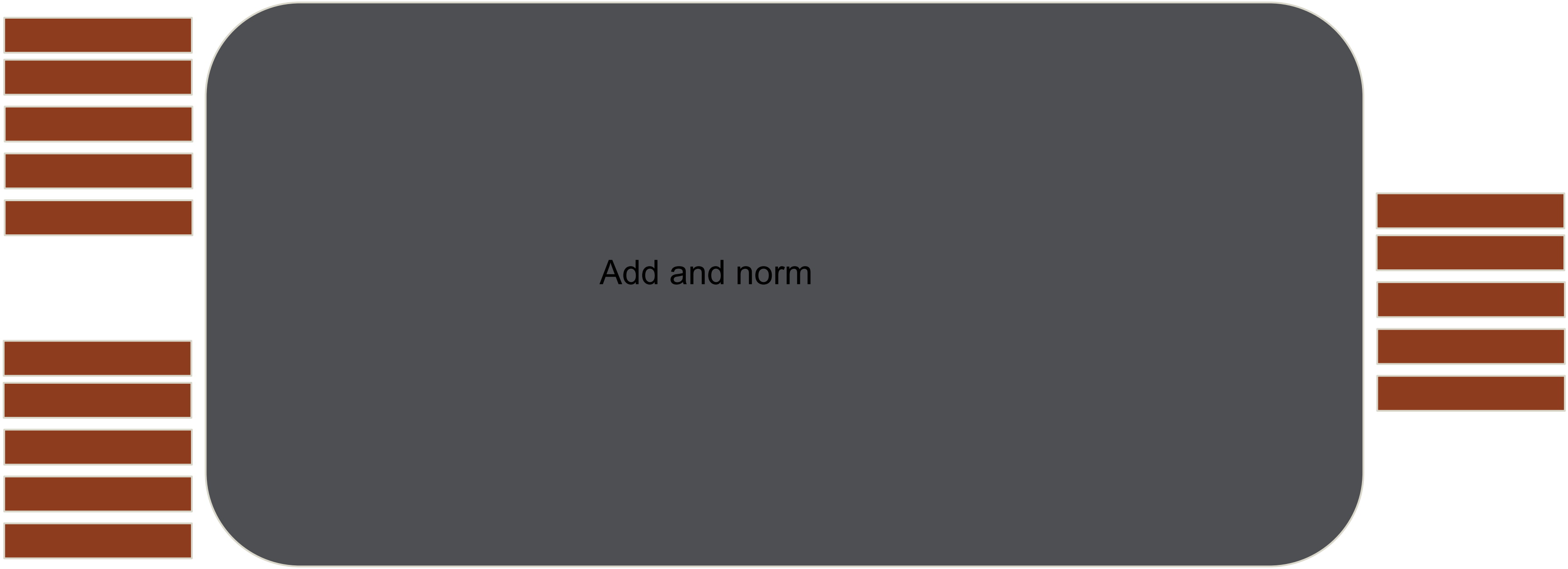


Without residuals,
with timing signals

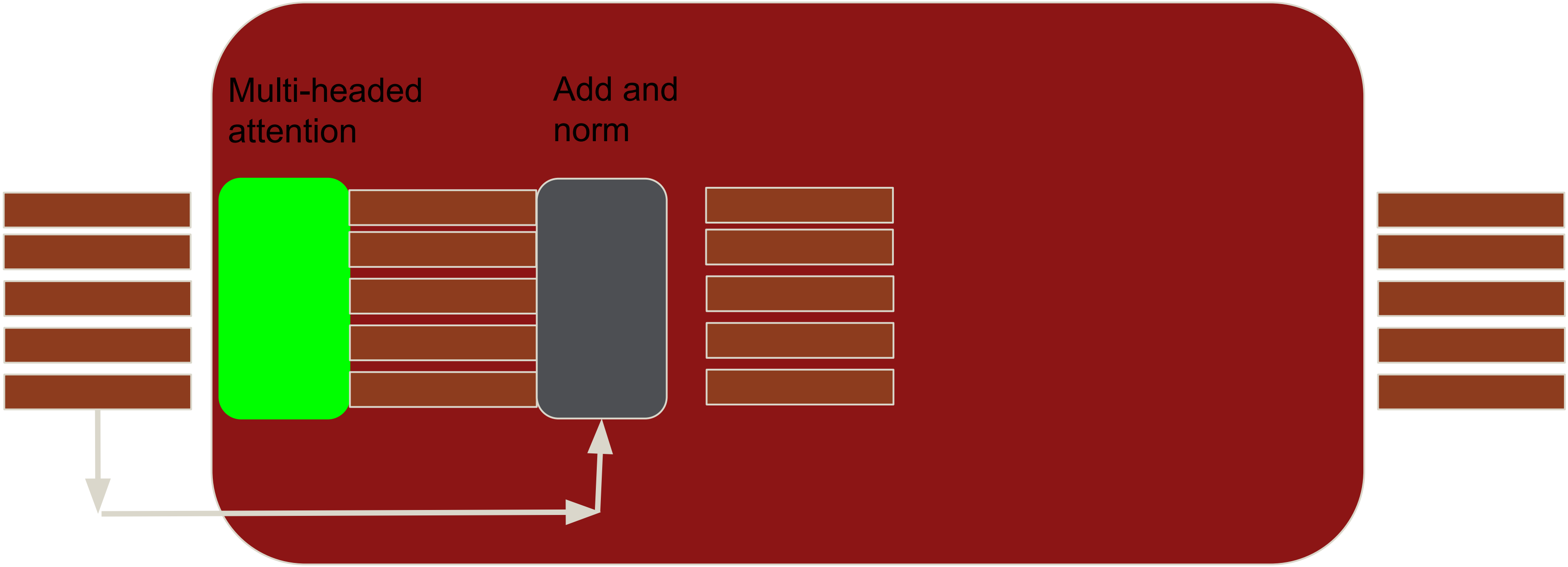
Add and Norm Layer



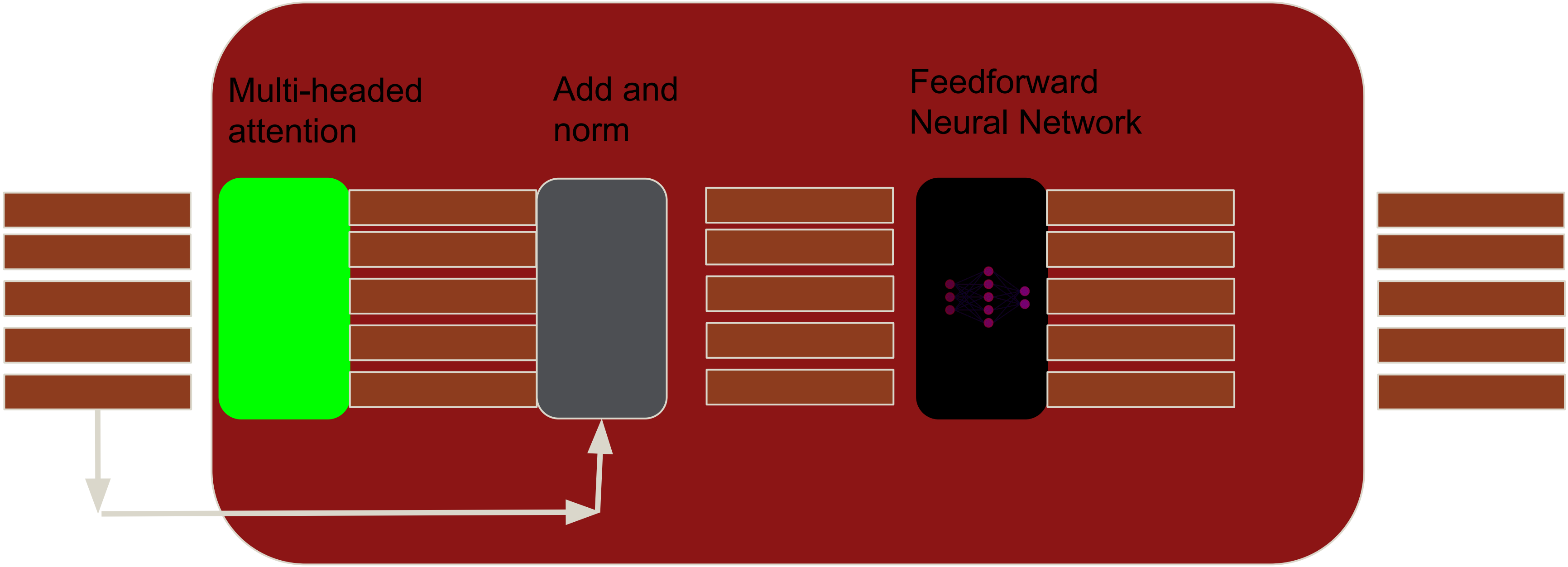
Add and Norm Layer



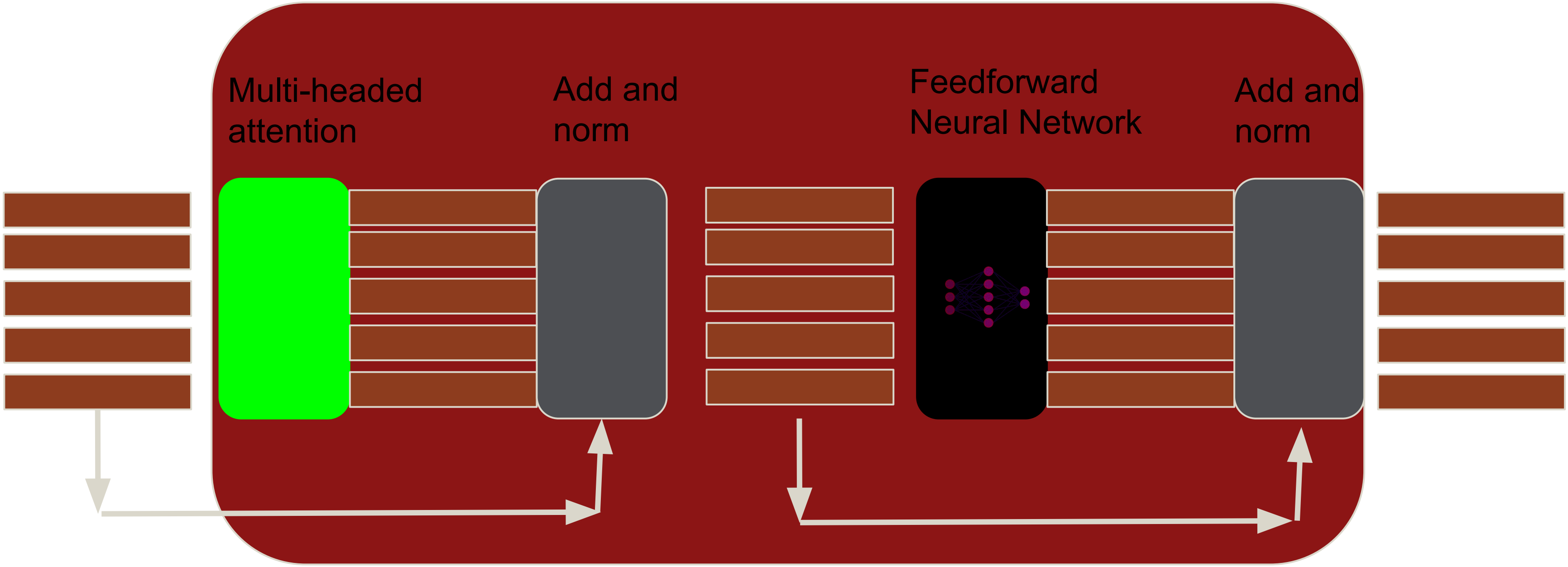
Encoder Block



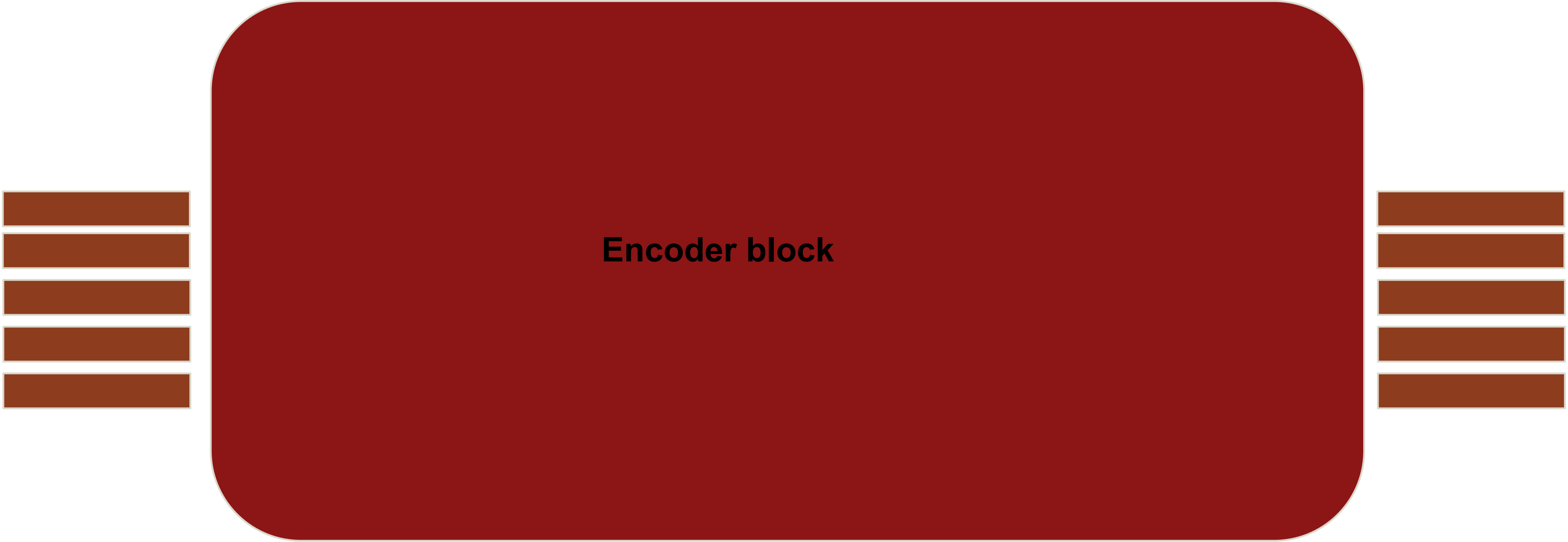
Encoder Block



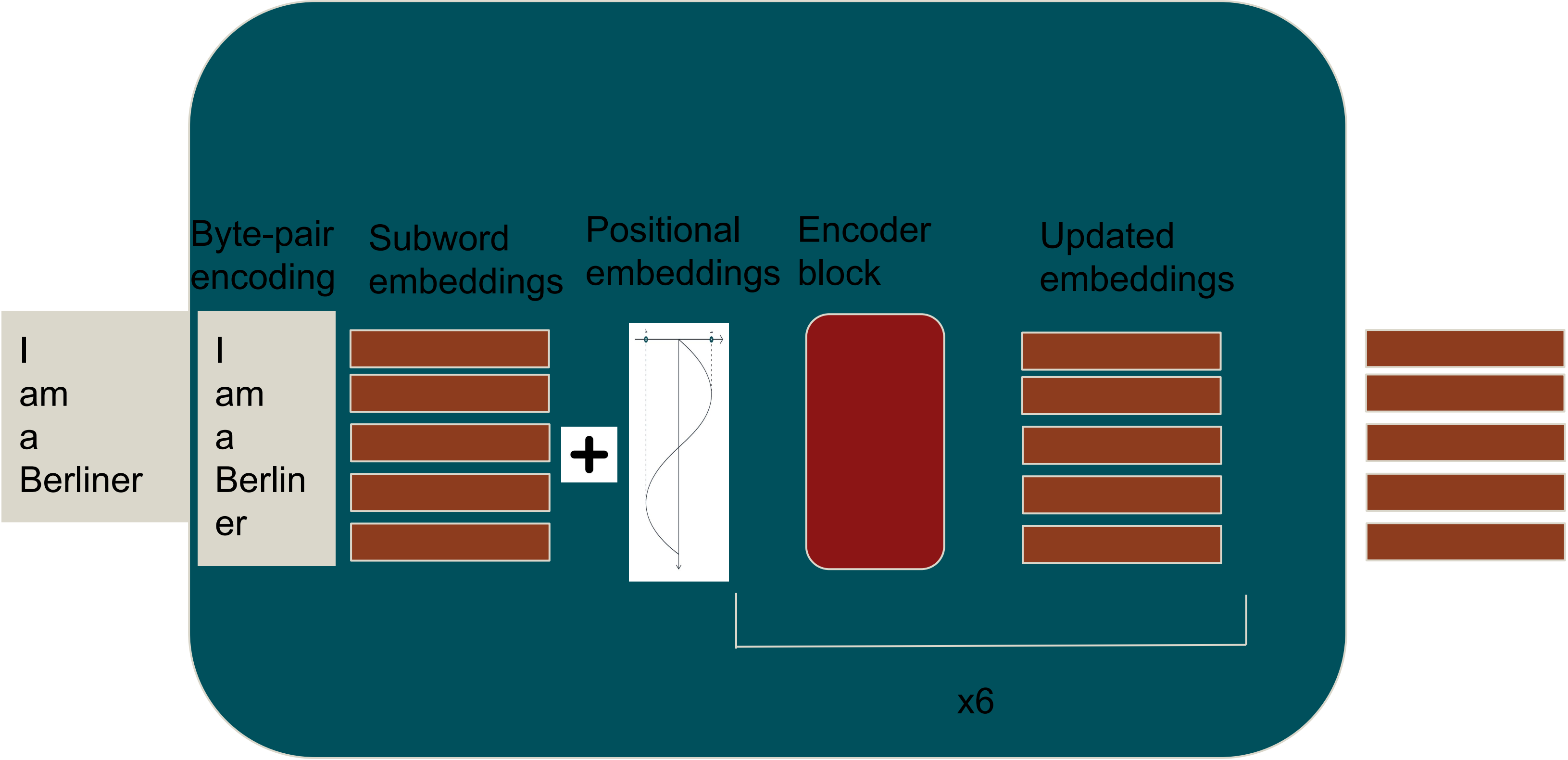
Encoder Block



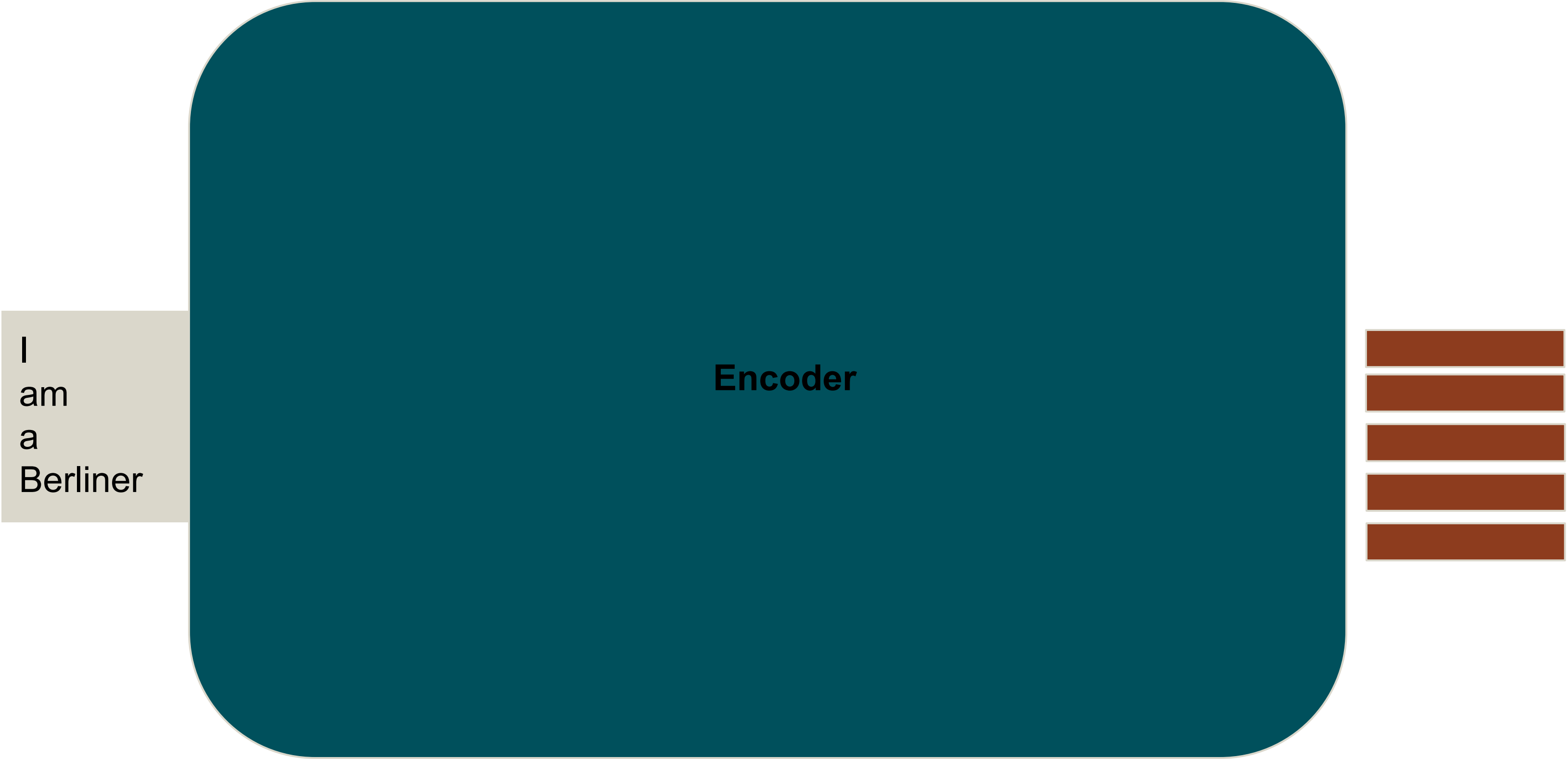
Encoder Block



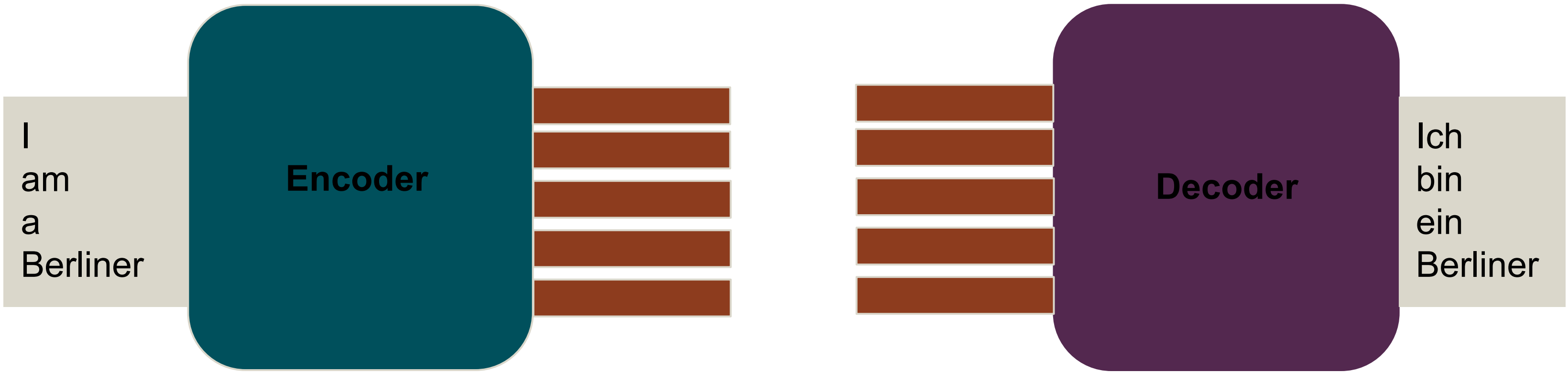
Encoder


I
am
a
Berliner
Byte-pair encoding
I
am
a
Berlin
er
Subword embeddings
+
Positional embeddings
Encoder block
Updated embeddings
x6

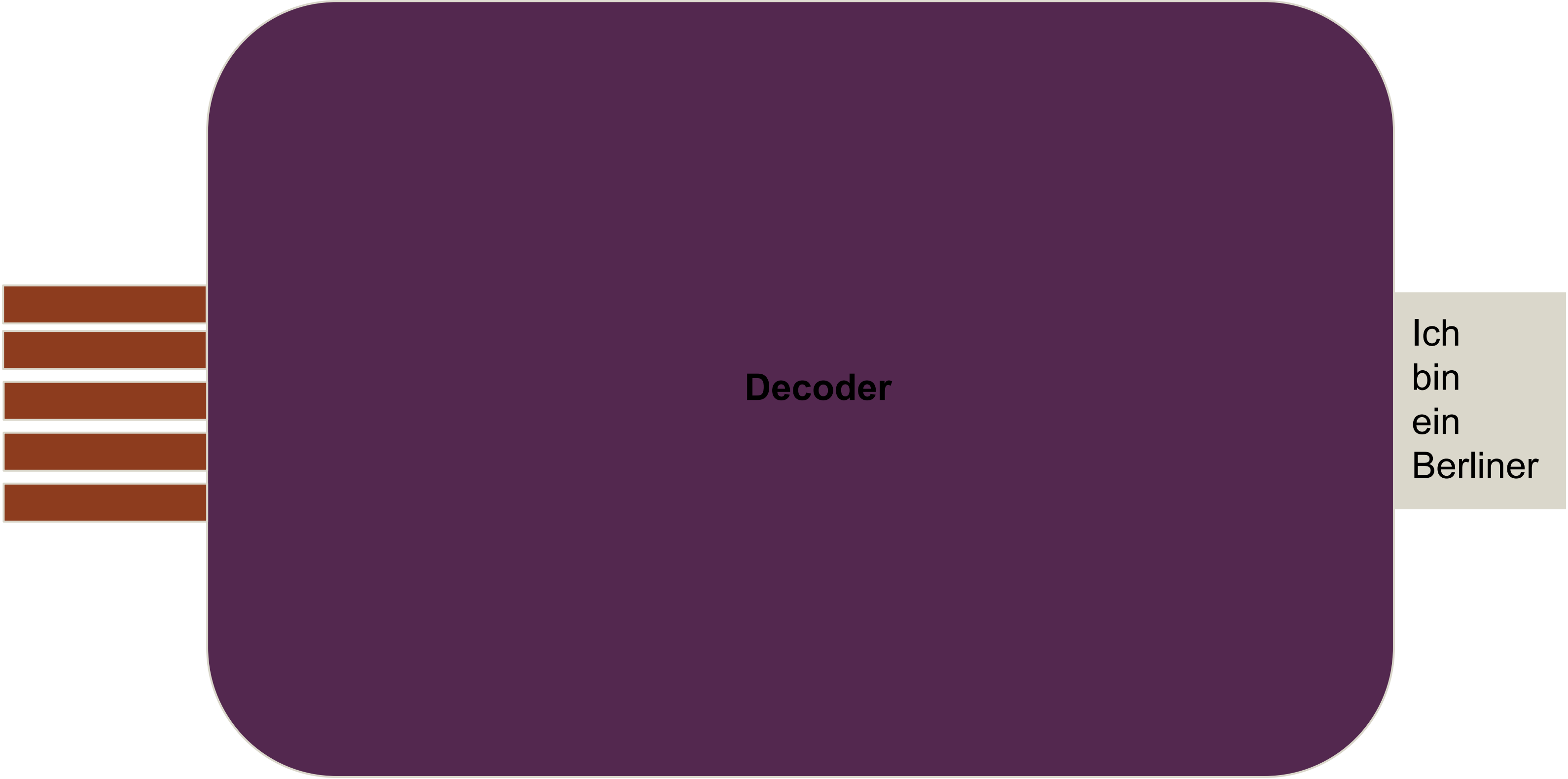
Encoder



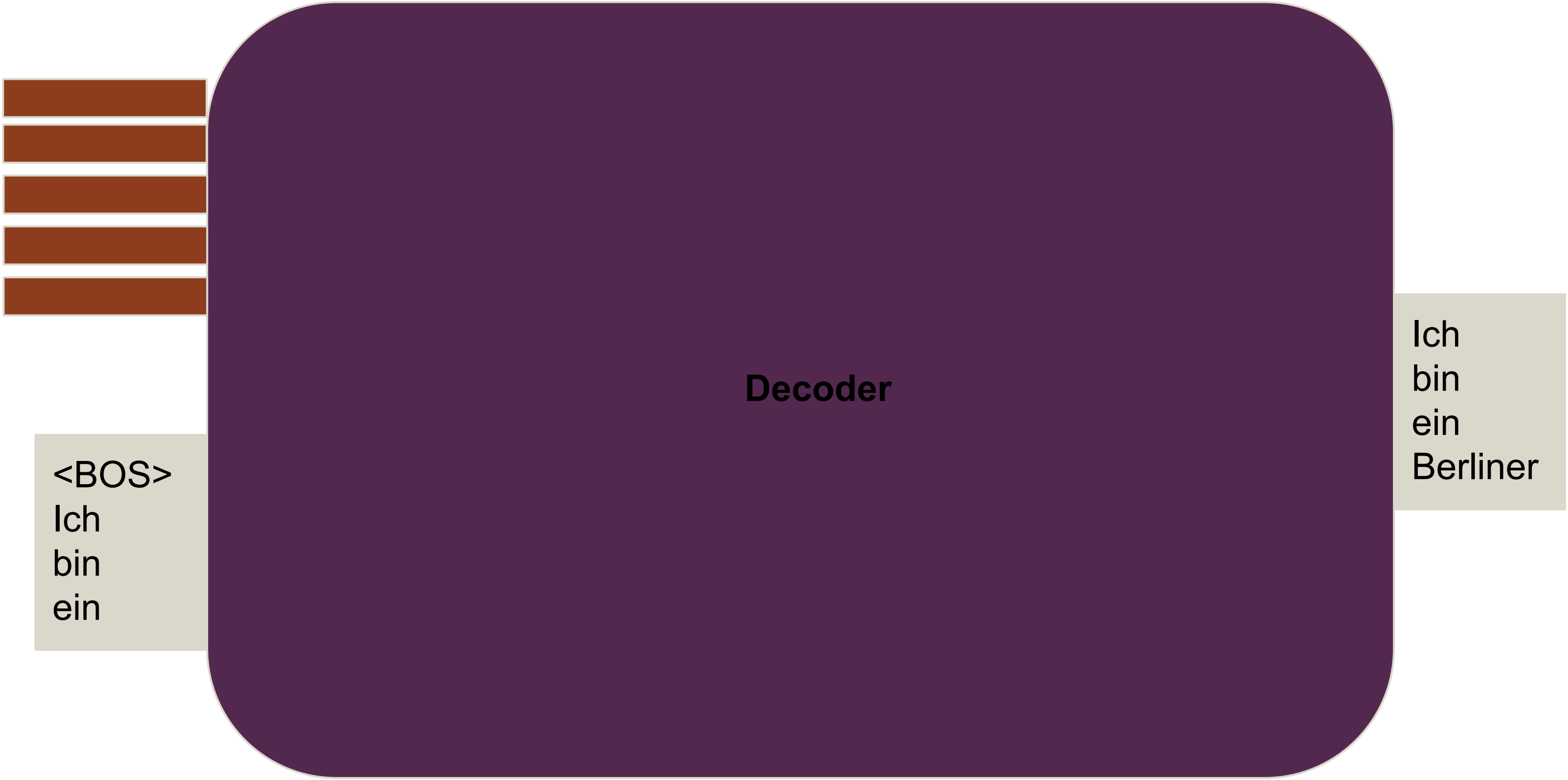
Highest Level of Abstraction: Encoder/Decoder



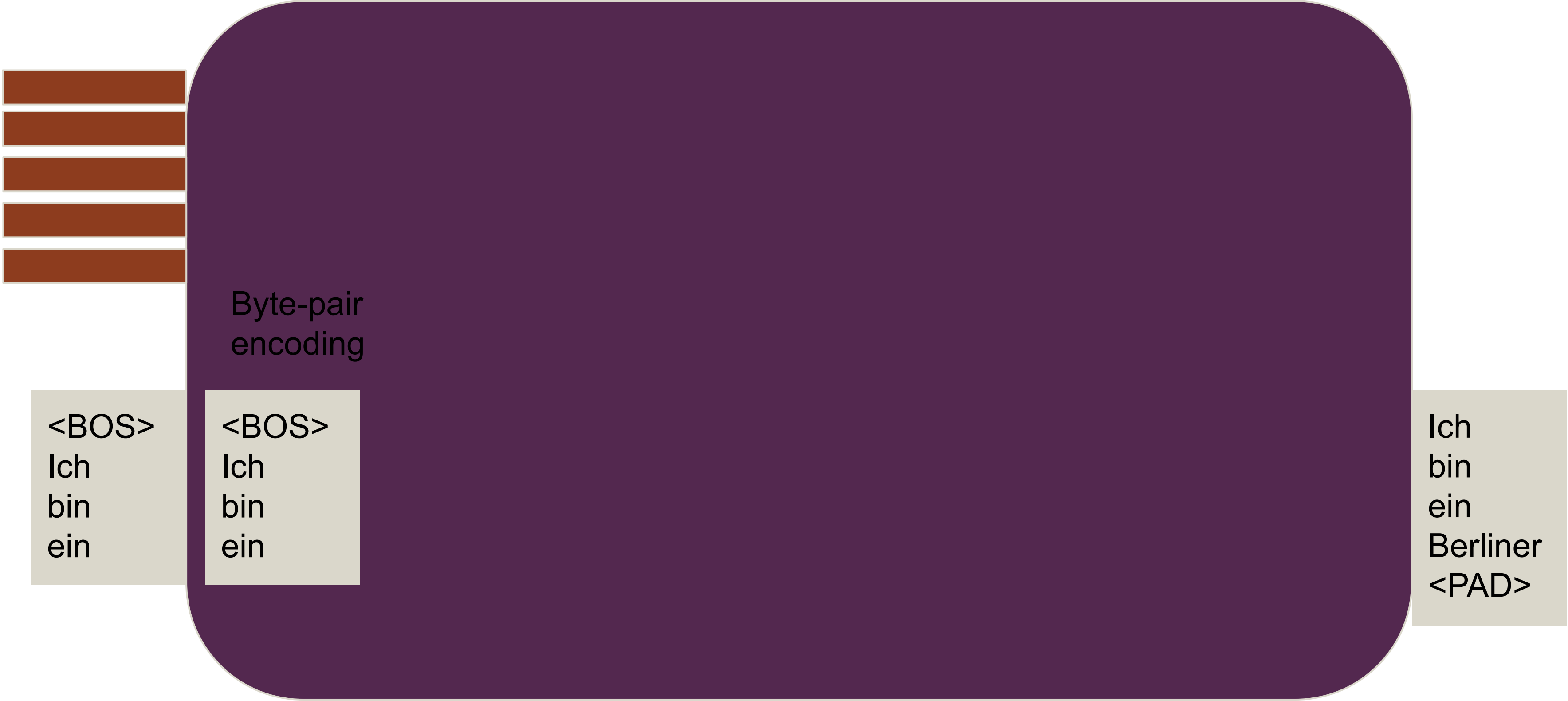
Decoder



Decoder (Training Time)

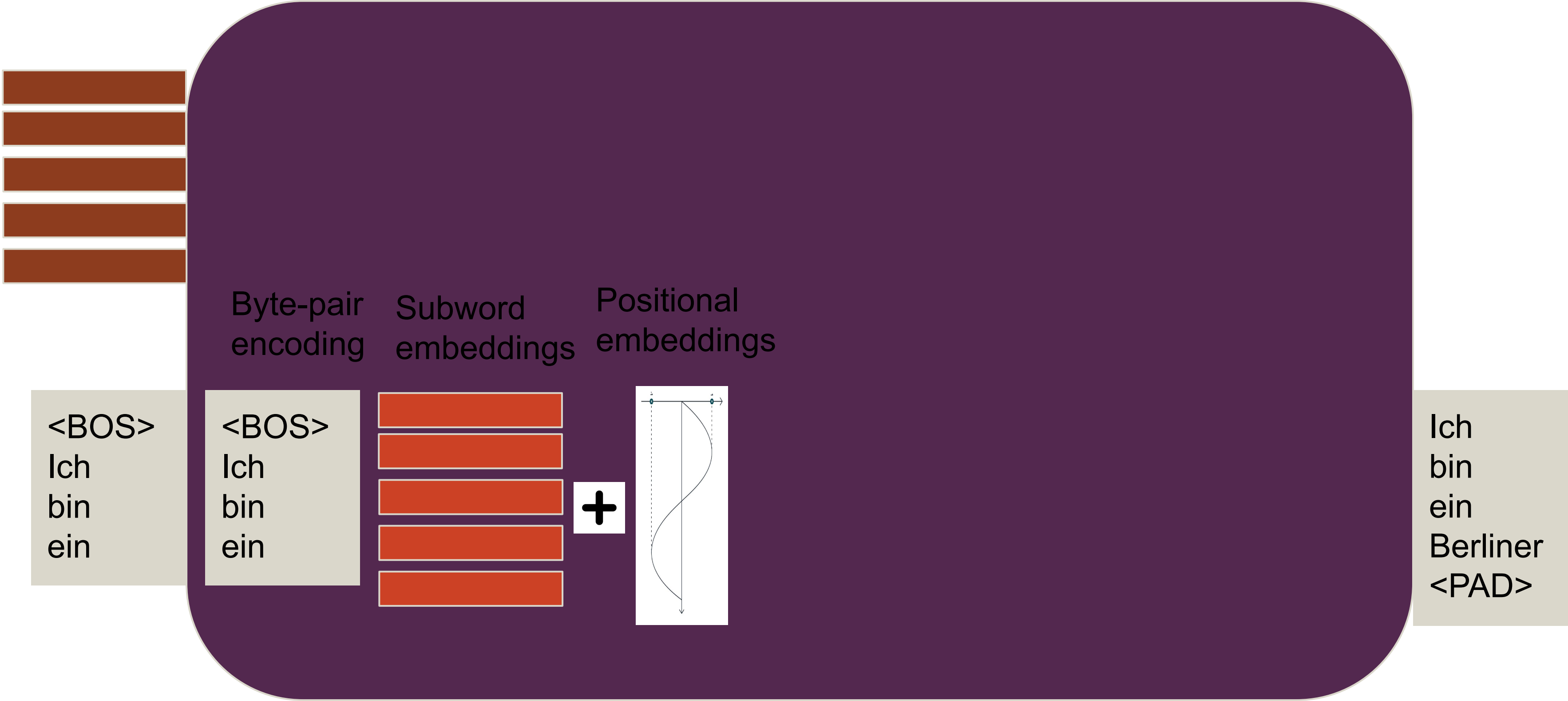


Decoder (Training Time)

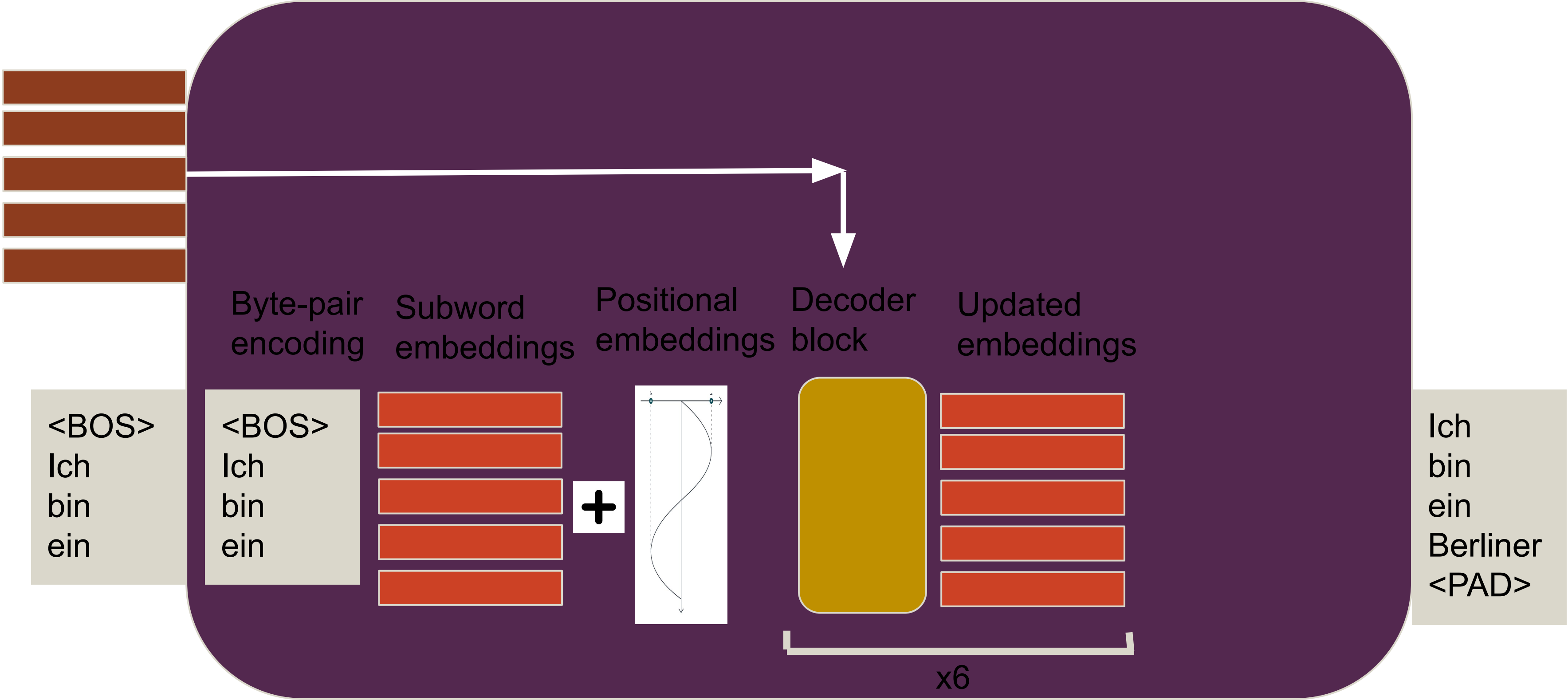


Decoder (Training Time)

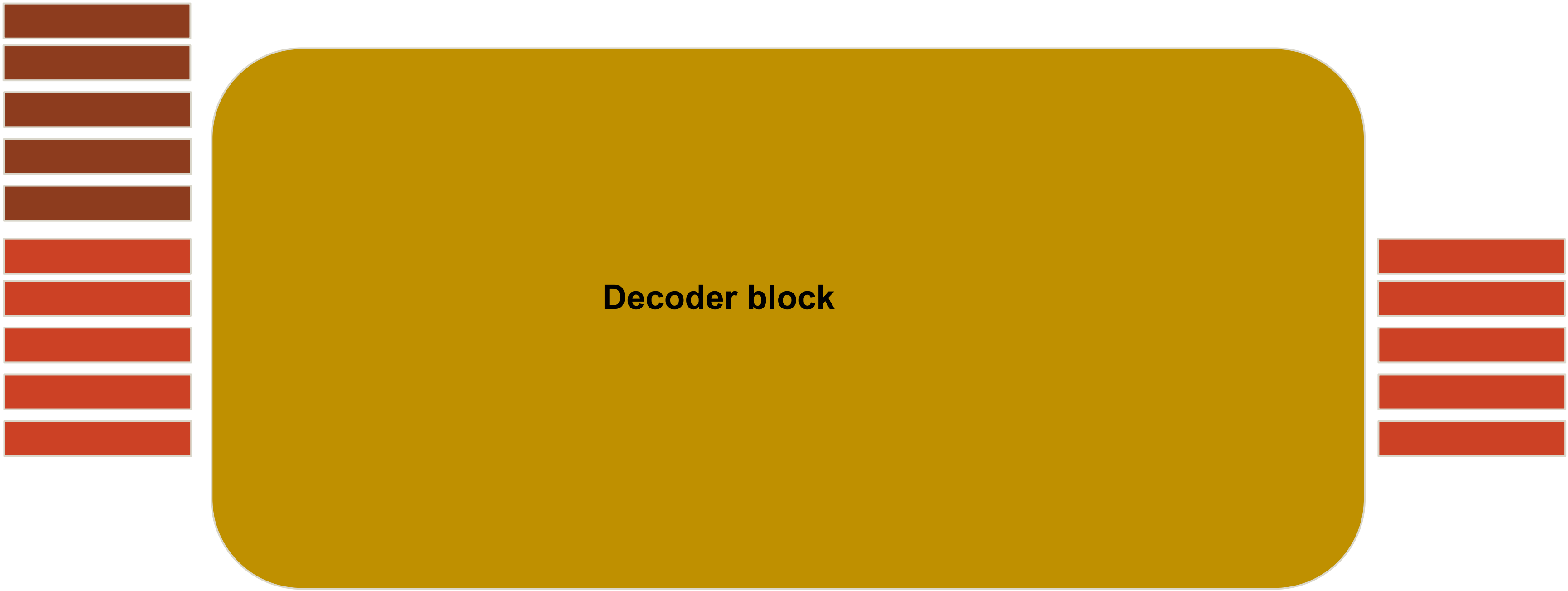
Decoder (Training Time)



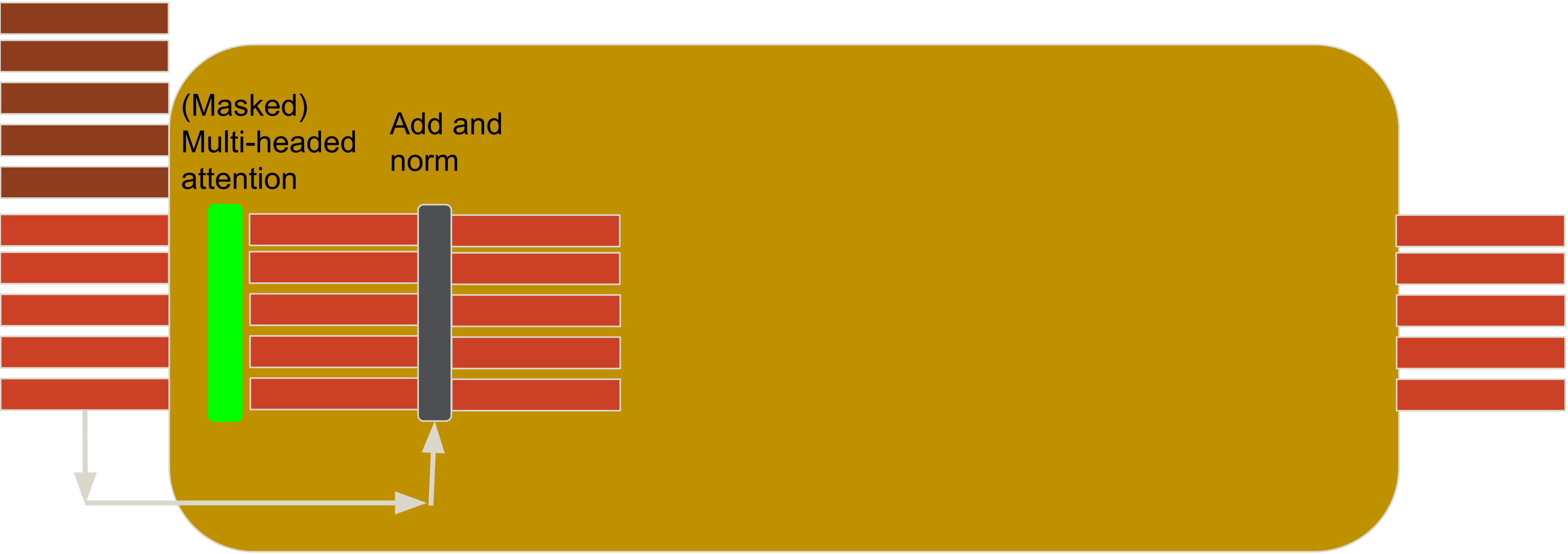
Decoder (Training Time)



Decoder Block



Decoder Block



What is Masked Multi-Headed Attention?

What is Masked Multi-Headed Attention?

- Clamp attention to word embeddings after you to zero

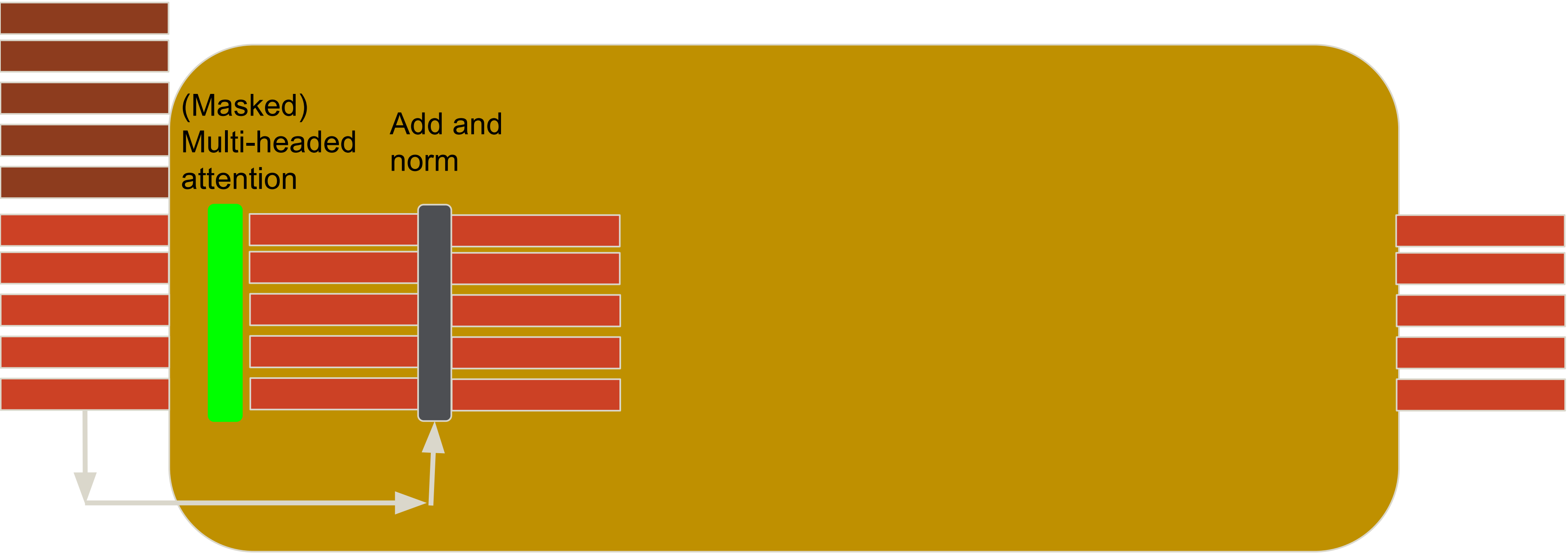
What is Masked Multi-Headed Attention?

- Clamp attention to word embeddings after you to zero
  - Add large negative numbers to similarity scores before softmax

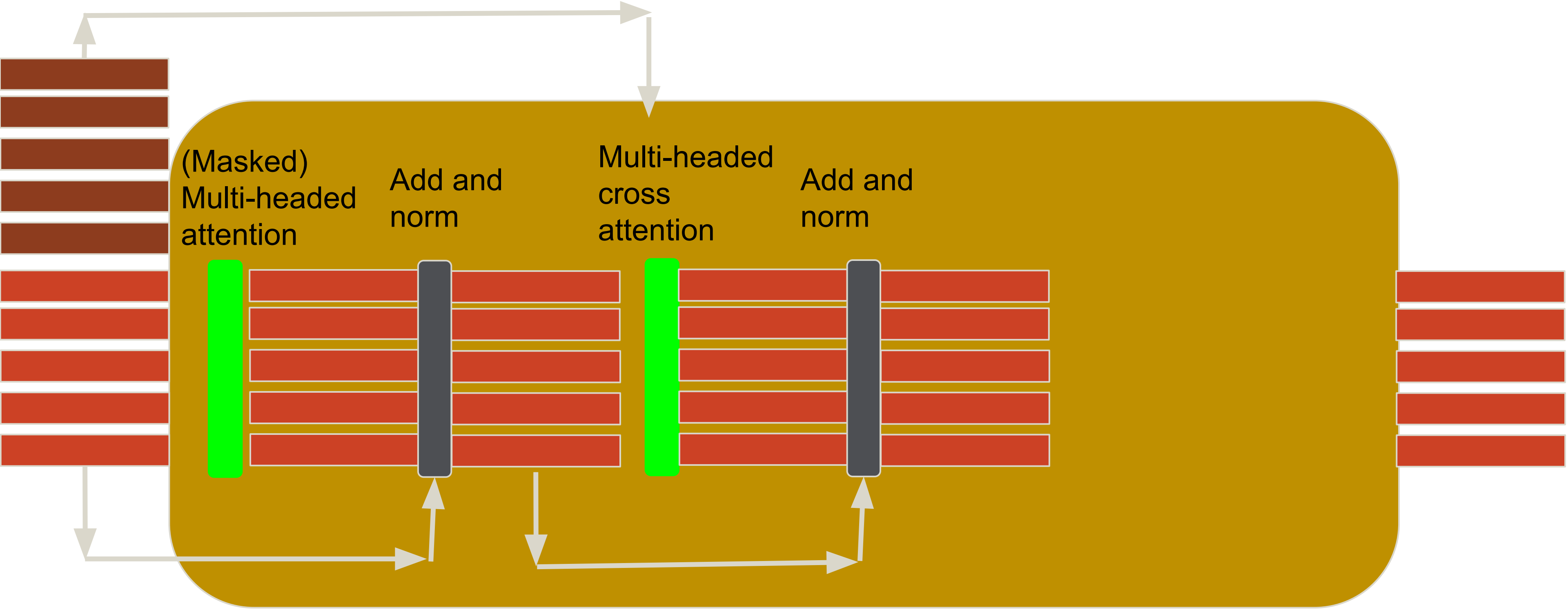
What is Masked Multi-Headed Attention?

- Clamp attention to word embeddings after you to zero
  - Add large negative numbers to similarity scores before softmax
- Didn’t have to worry about this with RNN because sequential

Decoder Block



Decoder Block



What about Multi-Headed Cross-Attention?

What about Multi-Headed Cross-Attention?

- Finally use encoder representation

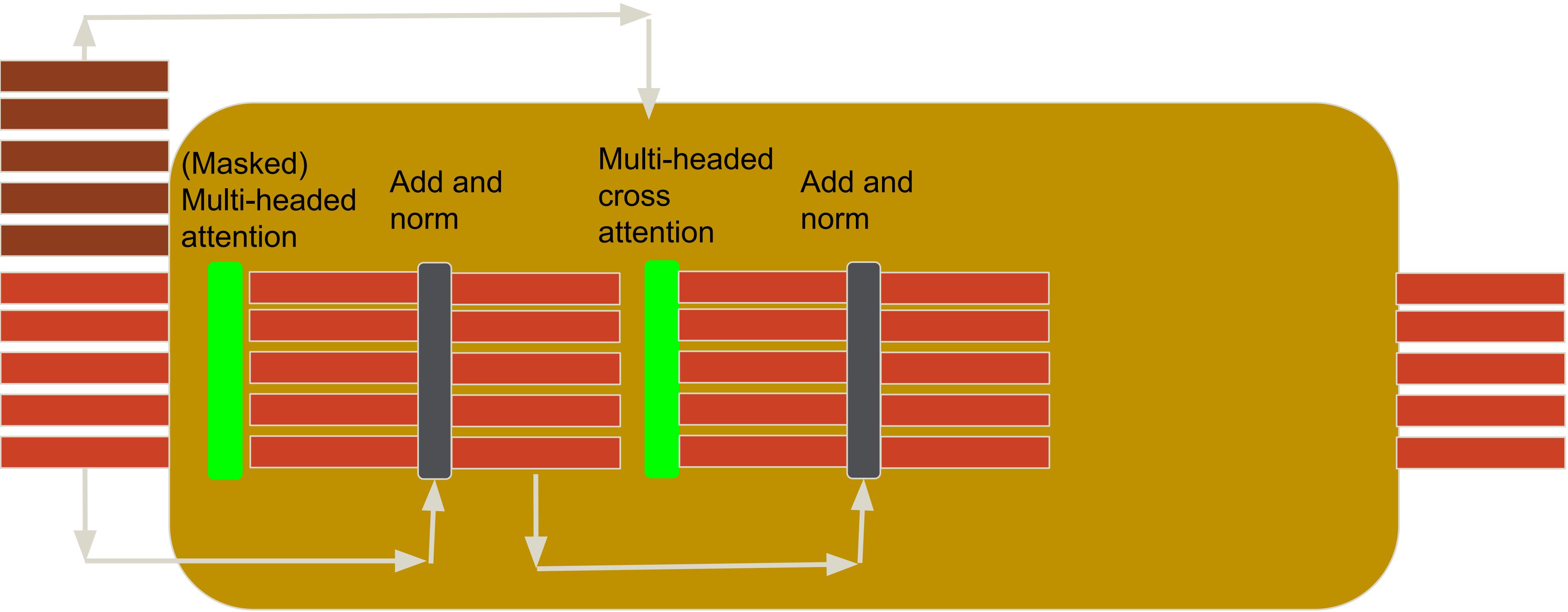
What about Multi-Headed Cross-Attention?

- Finally use encoder representation
  - Keys and values from encoder embeddings

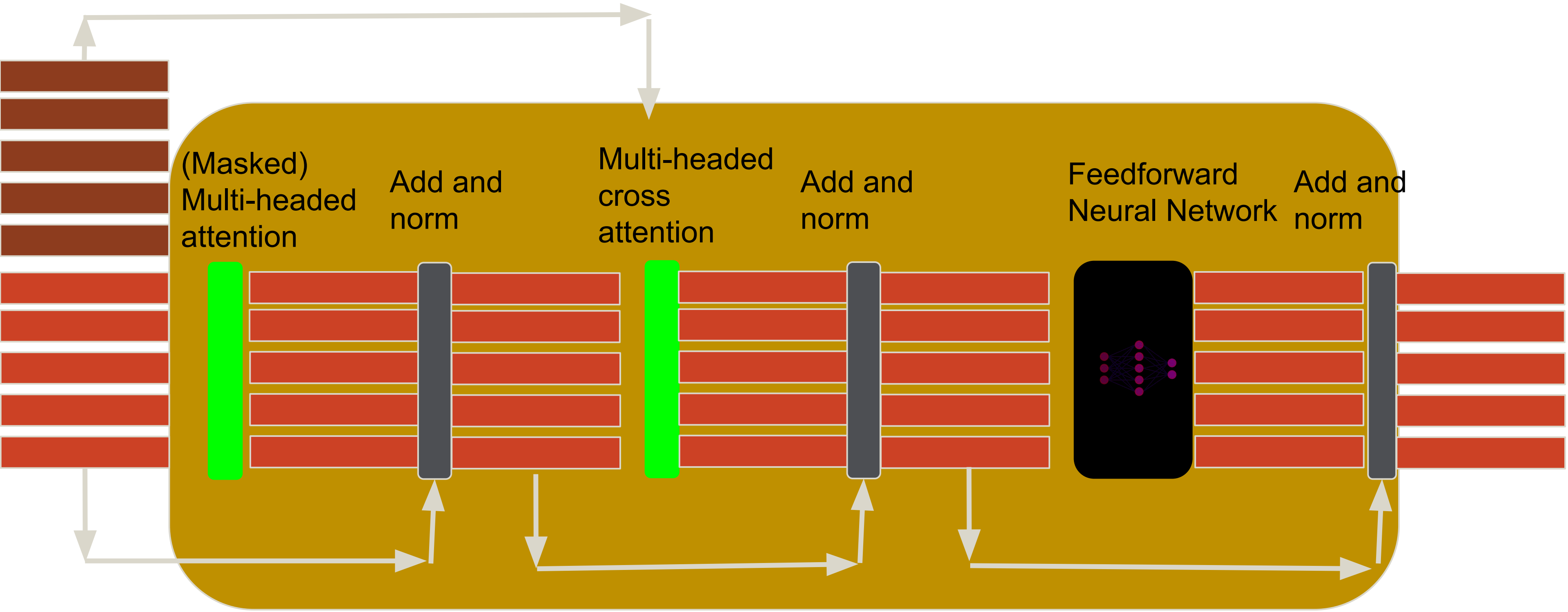
What about Multi-Headed Cross-Attention?

- Finally use encoder representation
  - Keys and values from encoder embeddings
  - Query from decoder embeddings

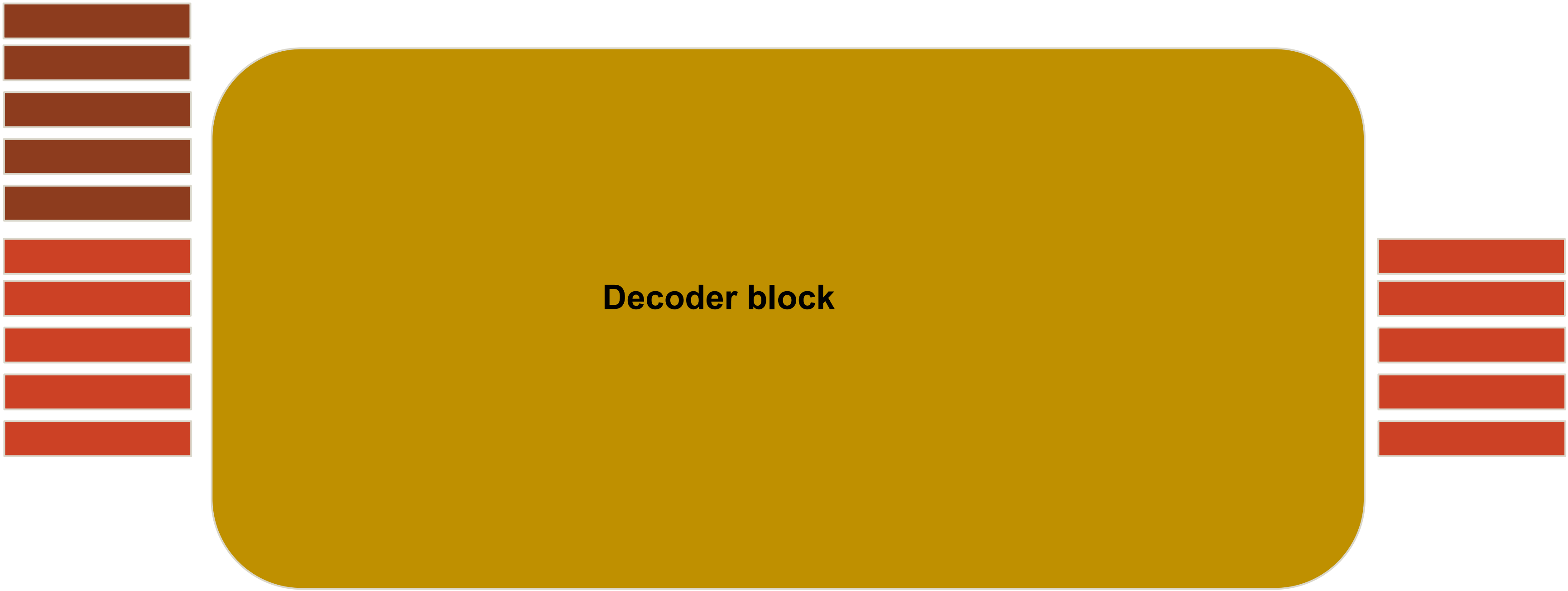
Decoder Block



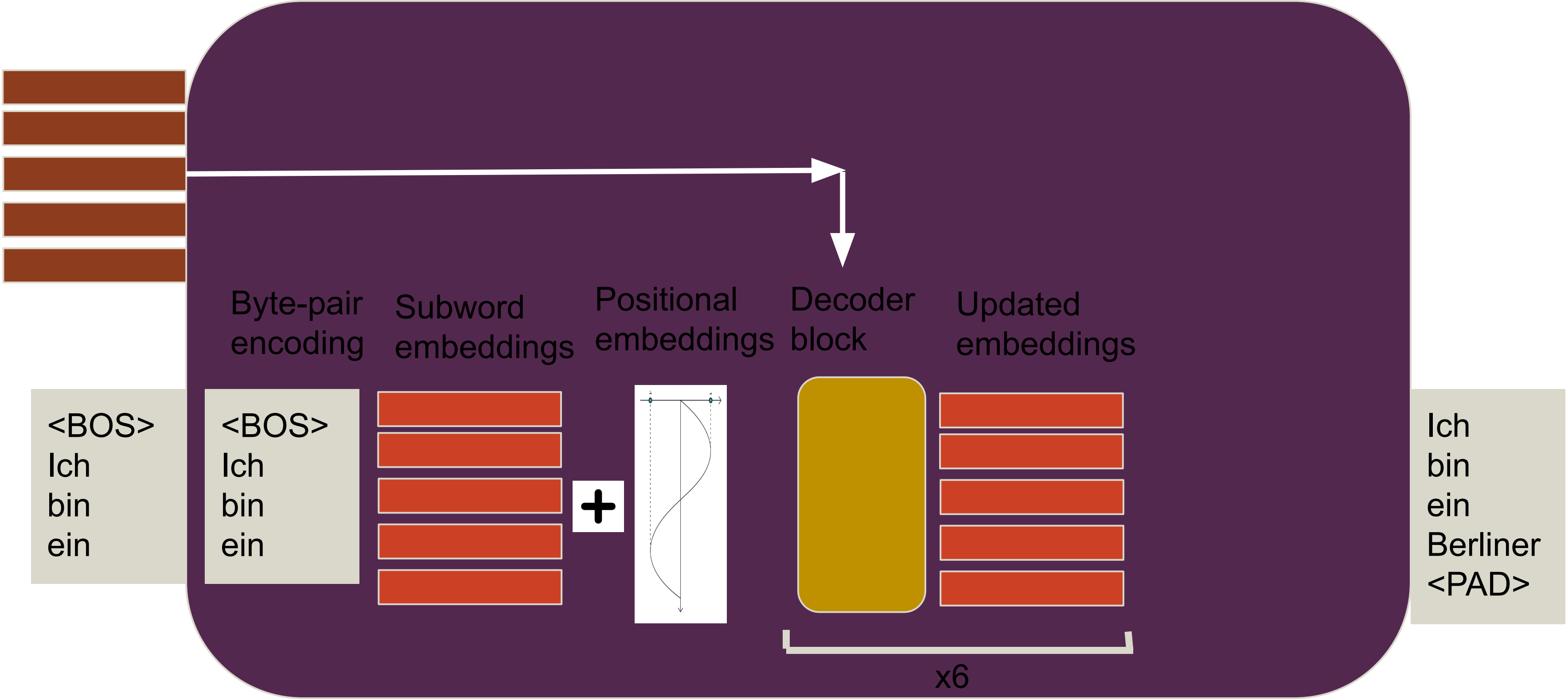
Decoder Block



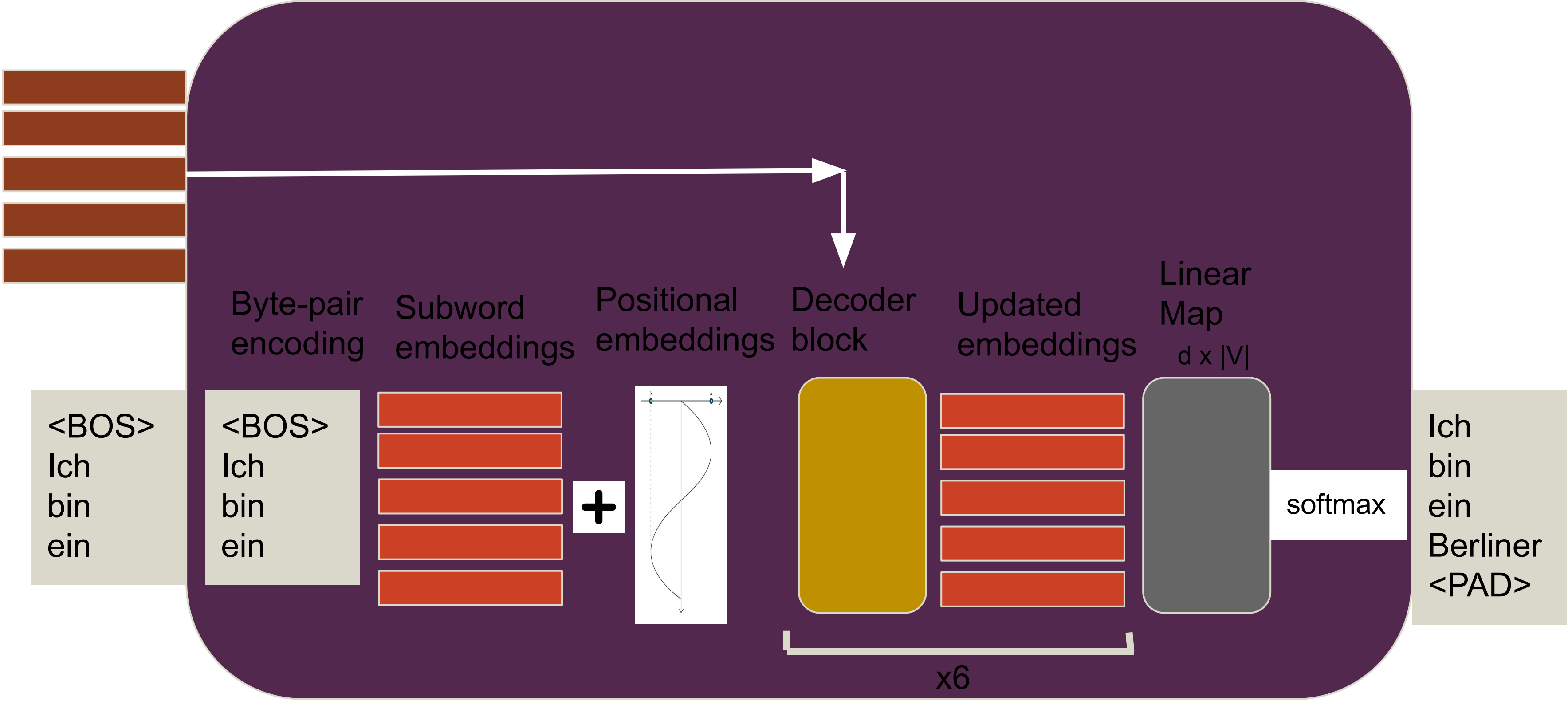
Decoder Block



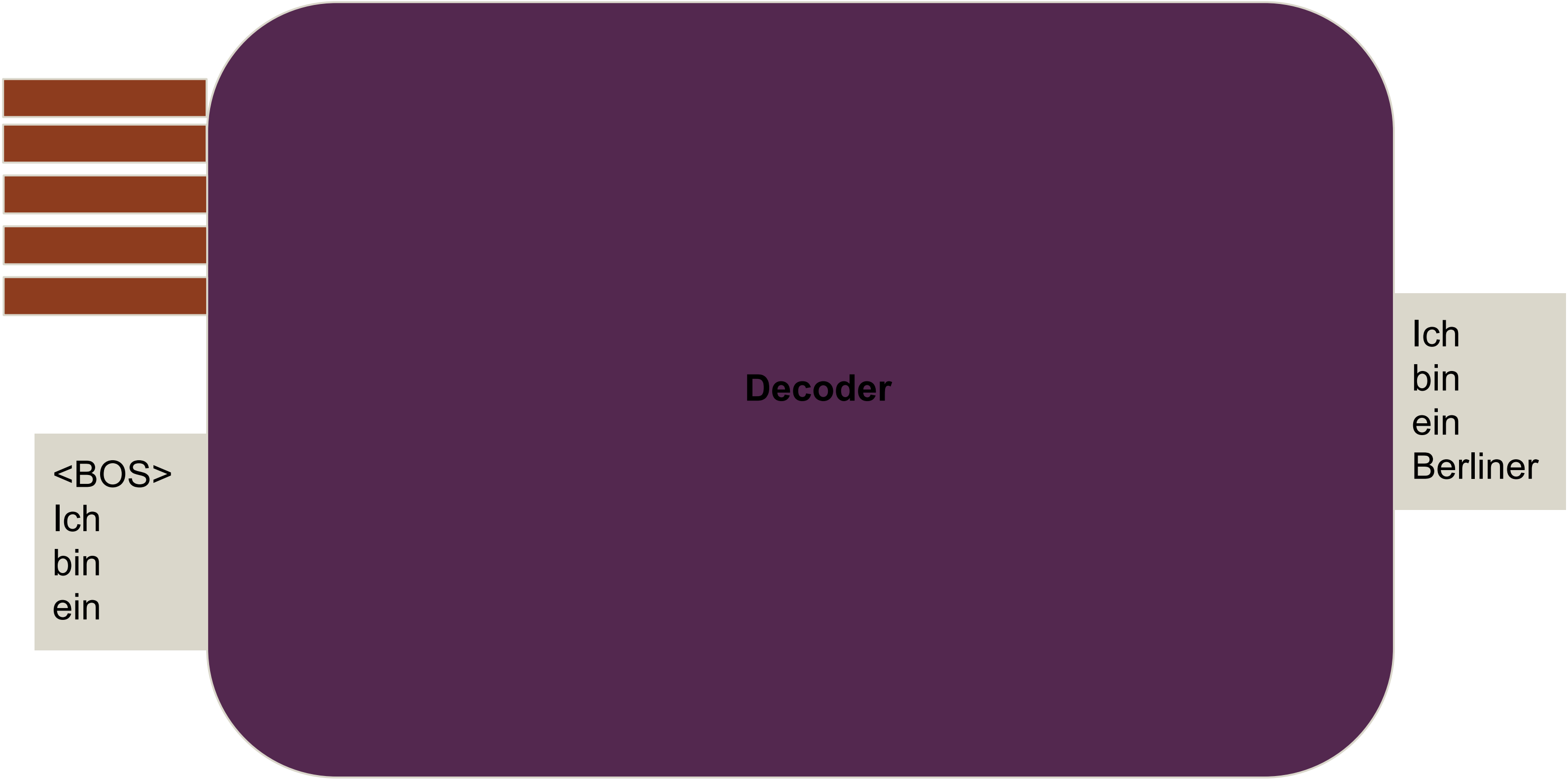
Decoder (Training Time)



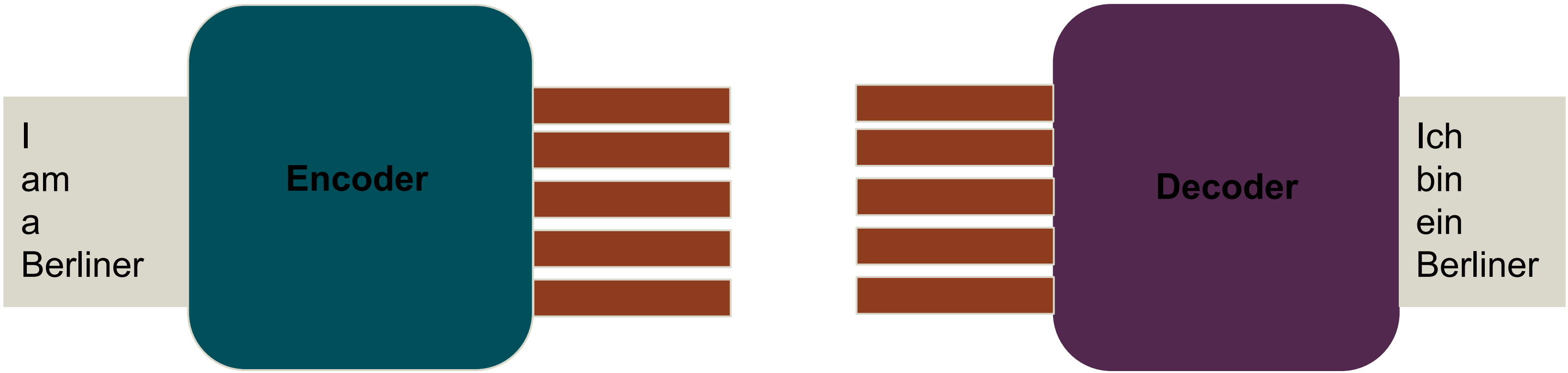
Decoder (Training Time)



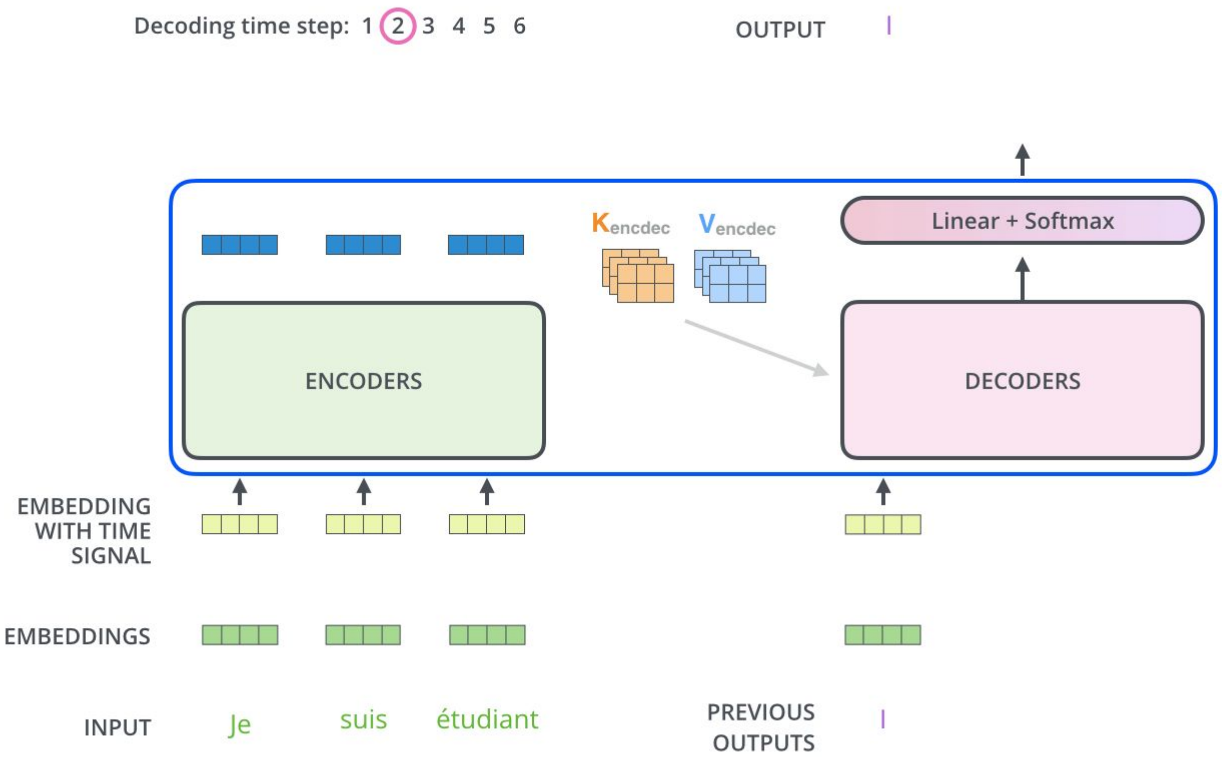
Decoder (Training Time)



Highest Level of Abstraction: Encoder/Decoder



Decoder (Test Time)

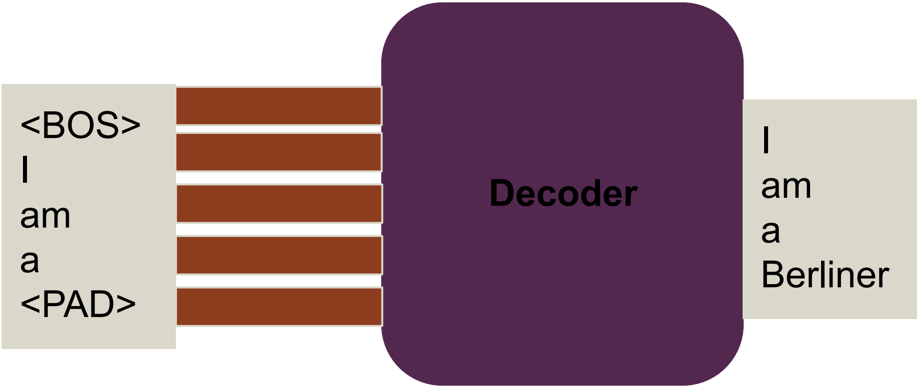


Training

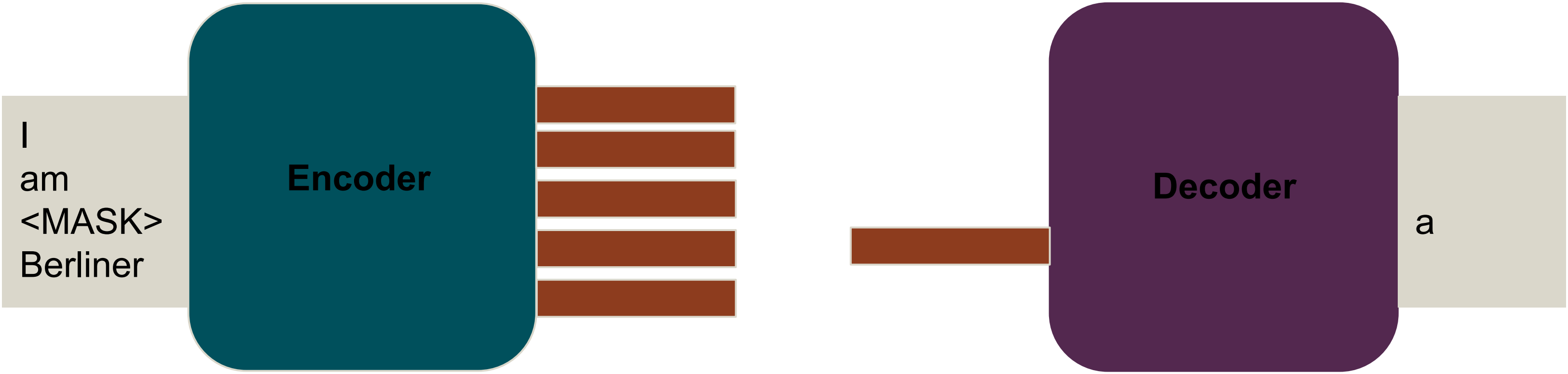
Self-Supervised Pre-training

- Idea: create tasks using the AMPLE unlabeled English text data we have available

Decoder Pre-training: Next Word Prediction



Encoder Pre-training: Masked Language Modeling



Encoder Pre-training: Masked Language Modeling

Encoder Pre-training: Masked Language Modeling

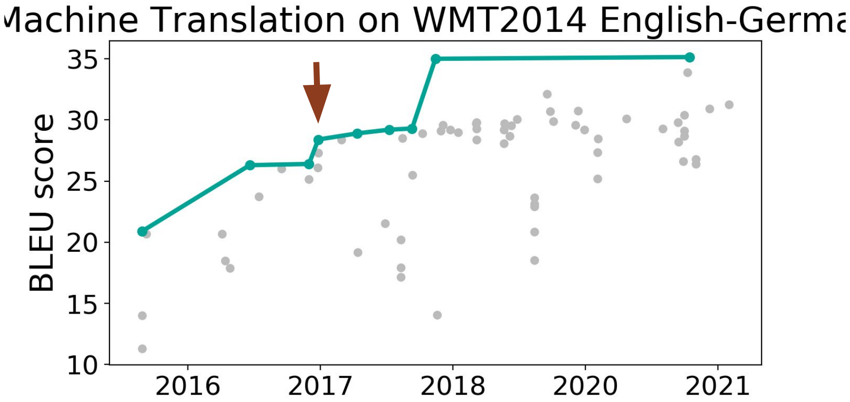
Results

A New SOTA

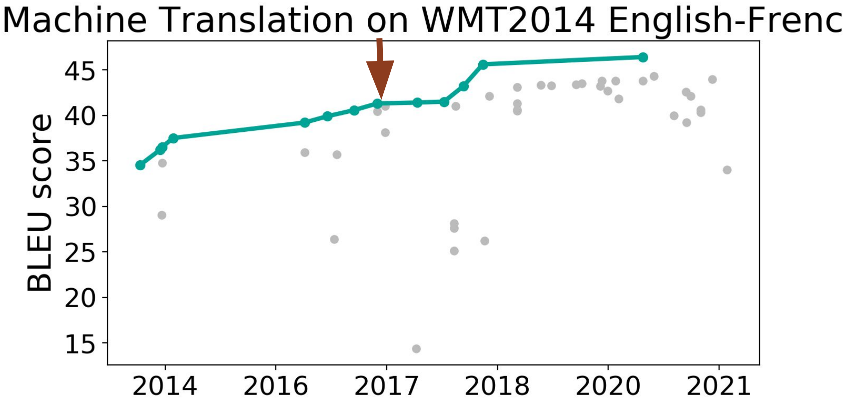
Table 2: The Transformer achieves better BLEU scores than previous state-of-the-art models on the English-to-German and English-to-French newstest2014 tests at a fraction of the training cost.

Model	BLEU		Training Cost (FLOPs)	
	EN-DE	EN-FR	EN-DE	EN-FR
ByteNet [18]	23.75			
Deep-Att + PosUnk [39]		39.2		$1.0 \cdot 10^{20}$
GNMT + RL [38]	24.6	39.92	$2.3 \cdot 10^{19}$	$1.4 \cdot 10^{20}$
ConvS2S [9]	25.16	40.46	$9.6 \cdot 10^{18}$	$1.5 \cdot 10^{20}$
MoE [32]	26.03	40.56	$2.0 \cdot 10^{19}$	$1.2 \cdot 10^{20}$
Deep-Att + PosUnk Ensemble [39]		40.4		$8.0 \cdot 10^{20}$
GNMT + RL Ensemble [38]	26.30	41.16	$1.8 \cdot 10^{20}$	$1.1 \cdot 10^{21}$
ConvS2S Ensemble [9]	26.36	**41.29**	$7.7 \cdot 10^{19}$	$1.2 \cdot 10^{21}$
Transformer (base model)	27.3	38.1	$\mathbf{3.3 \cdot 10^{18}}$	
Transformer (big)	**28.4**	**41.8**	$2.3 \cdot 10^{19}$	

A New SOTA



Machine Translation on WMT2014 English-Germa
BLEU score
35
30
25
20
15
10
2016
2017
2018
2020
2021



Machine Translation on WMT2014 English-Frenc
BLEU score
45
40
35
30
25
20
15
2014
2016
2017
2018
2020
2021

Strengths and Limitations

TRANSFORMERS
MORE THAN MEETS THE EYE

Desiderata:

Desiderata:

1. Low computational complexity per layer

Desiderata:

1. Low computational complexity per layer
    a. o(n^2 d)

Desiderata:

1. Low computational complexity per layer
   a. o(n^2 d)

Desiderata:

1. Low computational complexity per layer
    a. o(n^2 d)
2. Parallelizability

Desiderata:

1. Low computational complexity per layer
    a. o(n^2 d)
2. Parallelizability
    a. Don’t have to wait for previous tokens to be processed = BIG DATA

Desiderata:

1. Low computational complexity per layer
   a. o(n^2 d)
2. Parallelizability
   a. Don’t have to wait for previous tokens to be processed = BIG DATA
3. Low path length between tokens

Desiderata:

1. Low computational complexity per layer
   a. o(n^2 d)
2. Parallelizability
   a. Don’t have to wait for previous tokens to be processed = BIG DATA
3. Low path length between tokens
   a. O(1)

Limitations

1. Fixed window size

Limitations

1. Fixed window size
2. Computation scales quadratically with window size