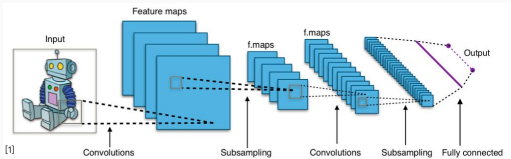
Transformer

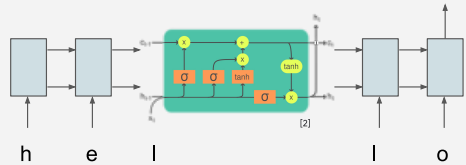
Computer Vision

Convolutional NNs (+ResNets)



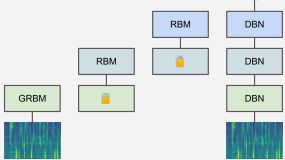
Natural Lang. Proc.

Recurrent NNs (+LSTMs)



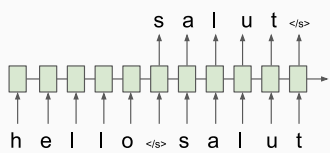
Speech

Deep Belief Nets (+non-DL)



Translation

Seq2Seq



RL

BC/GAIL

Algorithm 1 Generative adversarial imitation learning

1: **Input:** Expert trajectories $\pi_E \sim \pi_E$, initial policy and discriminator parameters θ_0, w_0
2: **for** $i = 0, 1, 2, \ldots$ **do**
3: Sample trajectories $\tau_i \sim \pi_{\theta_i}$
4: Update the discriminator parameters from w_i to w_{i+1} with the gradient

$$\hat{\mathbb{E}}_{\tau_i}[\nabla_w \log(D_w(s,a))] + \hat{\mathbb{E}}_{\pi_E}[\nabla_w \log(1 - D_w(s,a))] \tag{17}$$

5: Take a policy step from θ_i to θ_{i+1}, using the TRPO rule with cost function $\log(D_{w_{i+1}}(s,a))$. Specifically, take a KL-constrained natural gradient step with

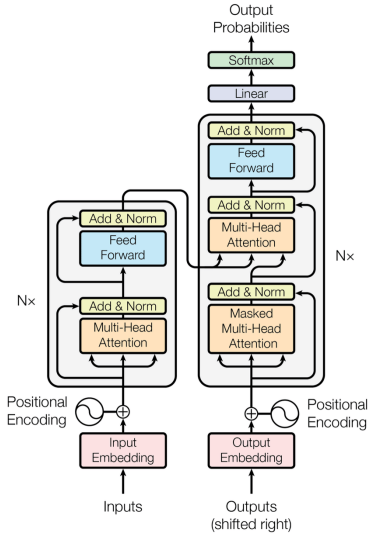
$$\hat{\mathbb{E}}_{\tau_i}[\nabla_\theta \log \pi_\theta(a|s) Q(s,a)] - \lambda \nabla_\theta H(\pi_\theta), \quad \text{where } Q(\bar{s},\bar{a}) = \hat{\mathbb{E}}_{\pi_i}[\log(D_{w_{i+1}}(s,a)) \mid s_0 = \bar{s}, a_0 = \bar{a}] \tag{18}$$

6: **end for**

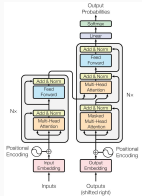
[1] CNN image CC-BY-SA by Aphex34 for Wikipedia https://commons.wikimedia.org/wiki/File:Typical_cnn.png
[2] RNN image CC-BY-SA by GChe for Wikipedia https://commons.wikimedia.org/wiki/File:The_LSTM_Cell.svg

Attention is All you Need

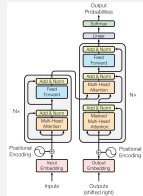
A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, I. Polosukhin (2017)



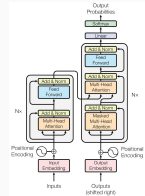
Computer Vision



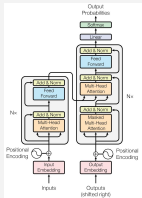
Natural Lang. Proc.



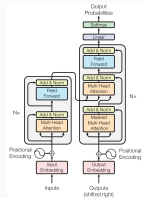
Reinf. Learning



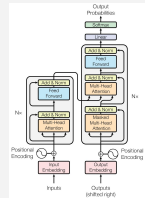
Speech



Translation

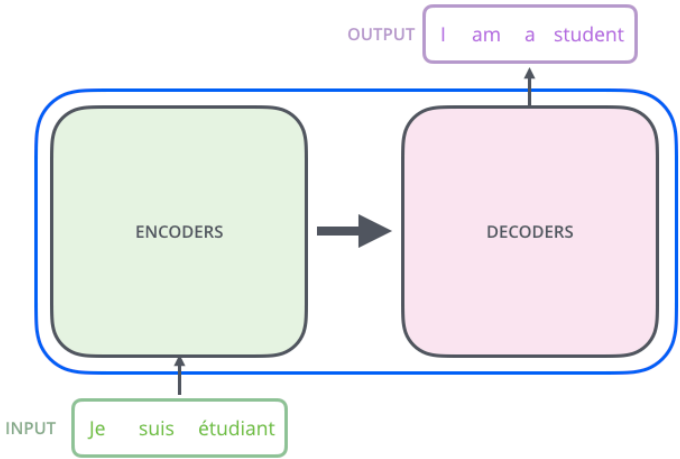


Graphs/Science

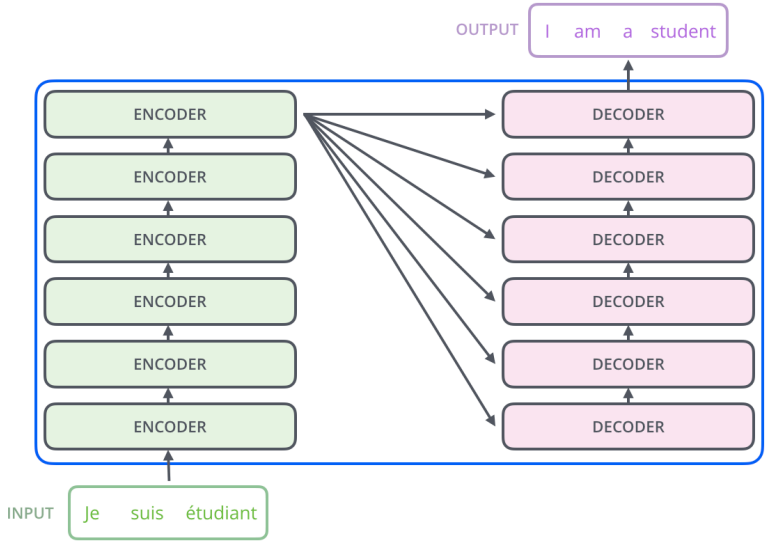


Transformer image source: "Attention Is All You Need" paper

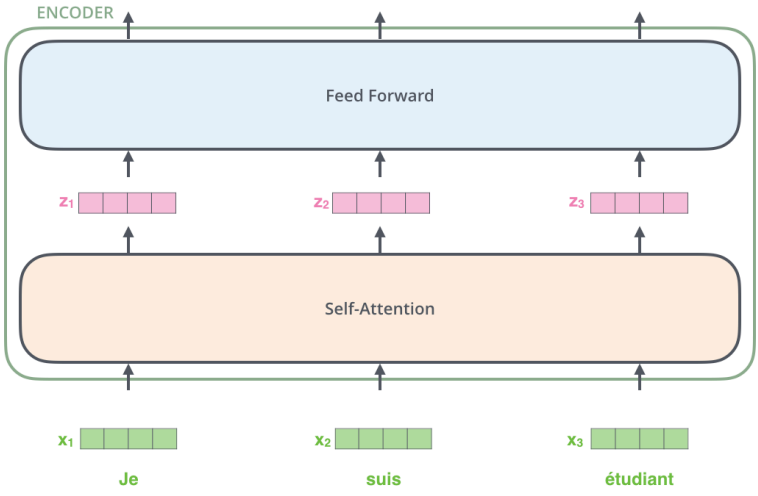
A big picture



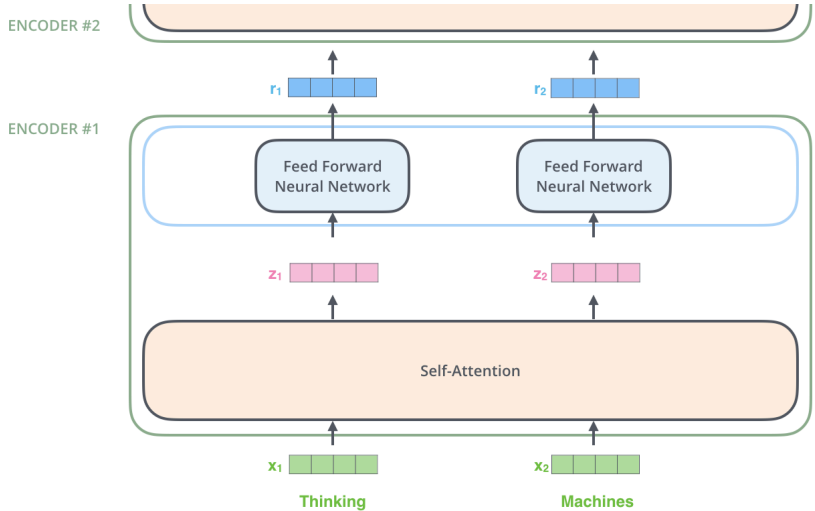
A smaller picture



Inside an Encoder



Zooming in on an Encoder



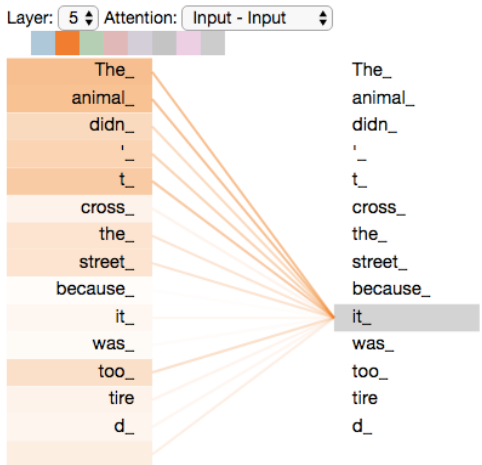
Idea behind Self-Attention

Consider an input sentence:

```
The animal didn't cross the street because it was too tired
```

What does “it” in this sentence refer to? The animal, the cross or the street?

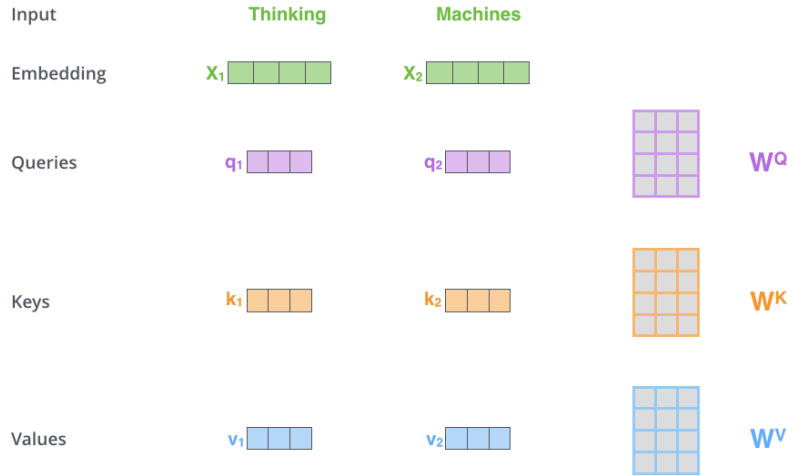
Self-Attention as finding relevant words



Try it yourself

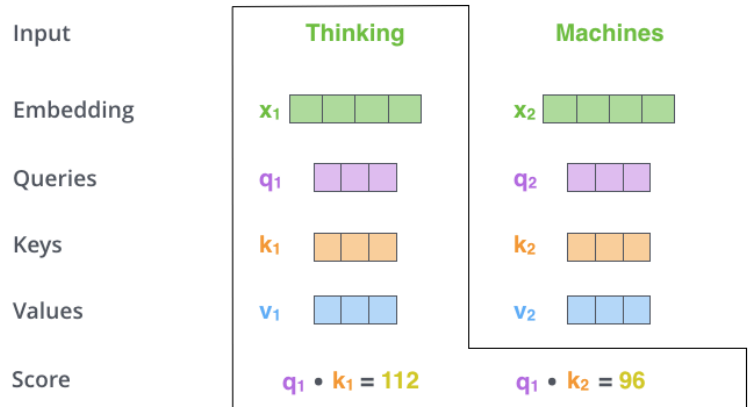
Calculating Self-Attention

Step 1: From an input vectors, create **Q**uery vector, **K**ey vector and **V**alue vector



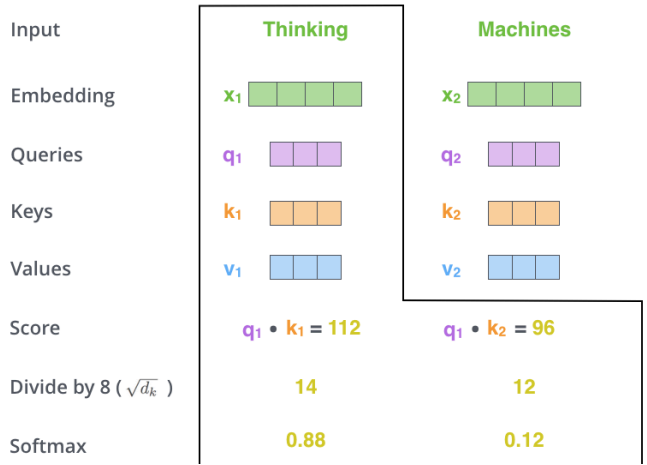
Calculating Self-Attention

Step 2: Compute the **scores** of each word by taking dot-product of its **query** and the **keys** of the all words



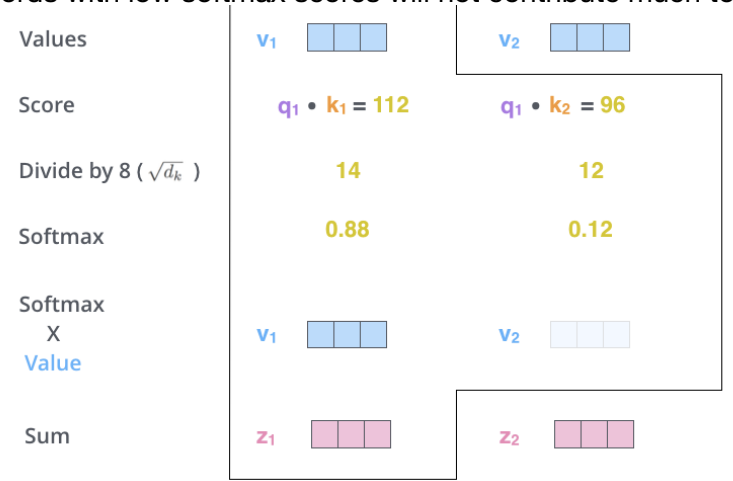
Calculating Self-Attention

Step 3: Divide by 8 (or the square root of the dimension of the key vectors) and apply Softmax



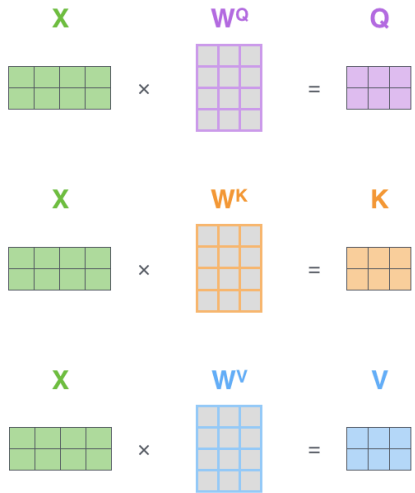
Calculating Self-Attention

Step 4: Multiply each value vector by the softmax score, then sum the vectors. Irrelevant words with low softmax scores will not contribute much to the sum

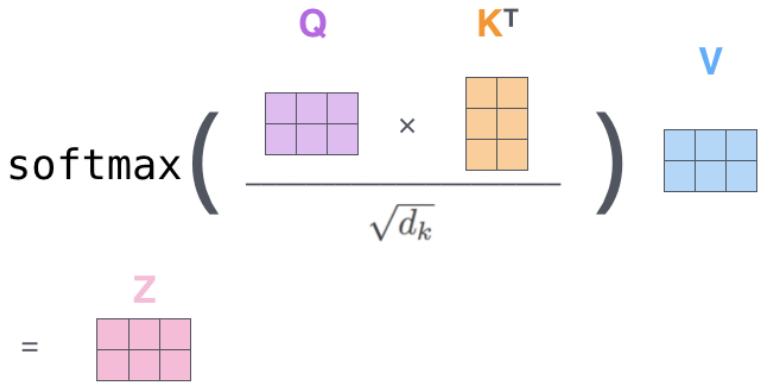


Self-Attention with Multiple inputs

We can write multiple dot-products as a matrix multiplication



Self-Attention in one equation

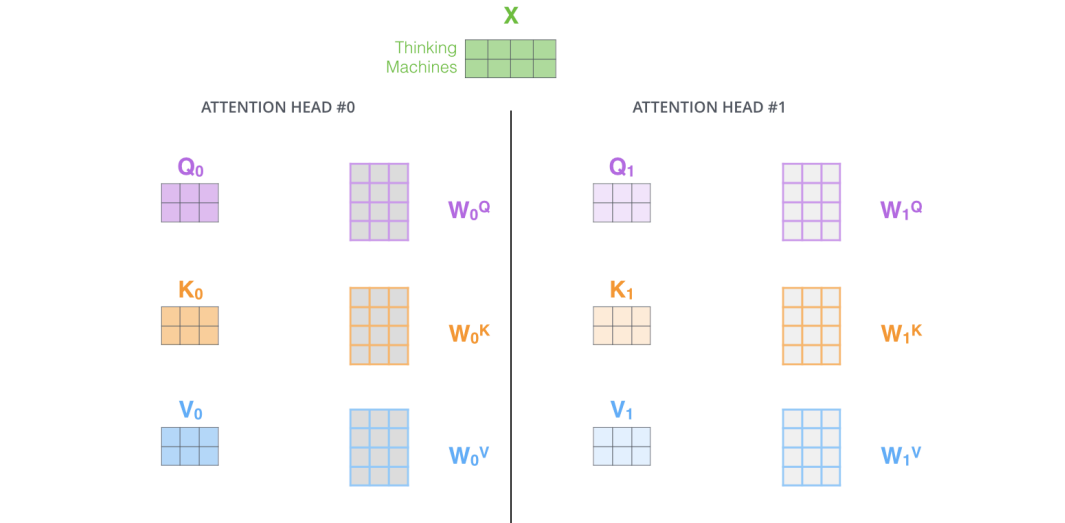


Multi-head Attention

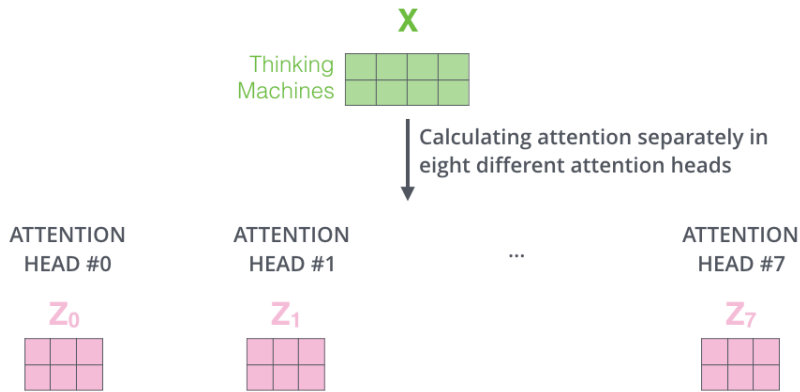
Motivation: want multiple Query/Key/Value combinations

Multi-head Attention

Motivation: want multiple Query/Key/Value combinations



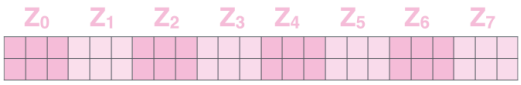
Multi-head Attention



but the feed-forward layer only takes a single matrix. How do we combine these into a single matrix?

Combining matrices

1) Concatenate all the attention heads

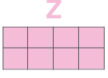


2) Multiply with a weight matrix W^O that was trained jointly with the model

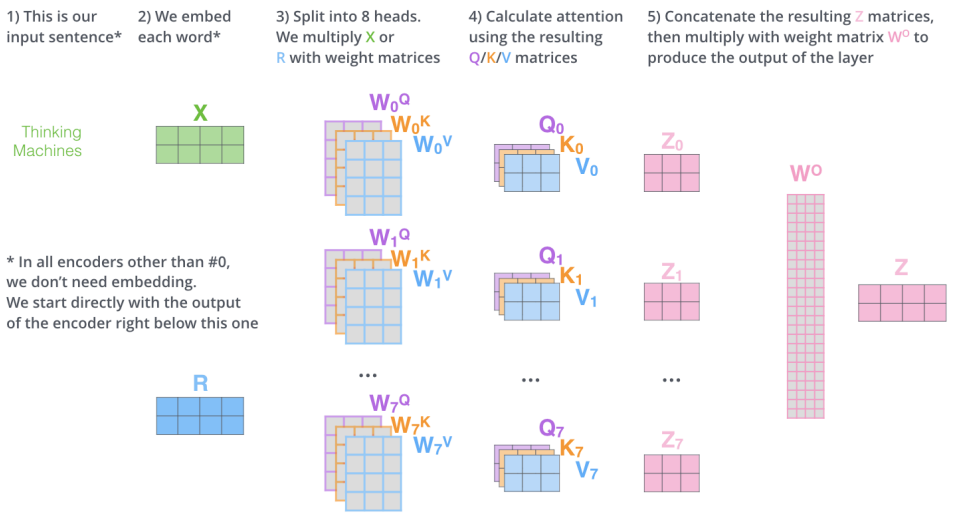
X

3) The result would be the Z matrix that captures information from all the attention heads. We can send this forward to the FFNN

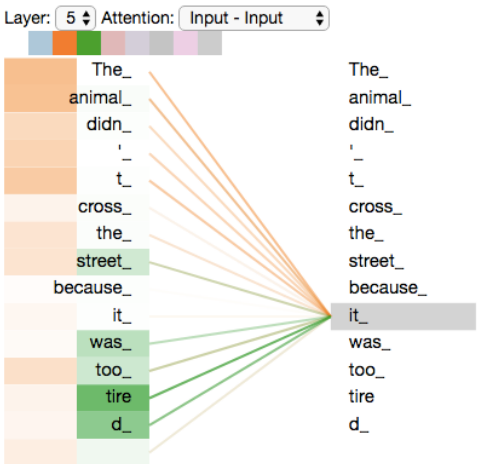
=



Summary of Multi-head Self-Attention



Revisit the Visualization



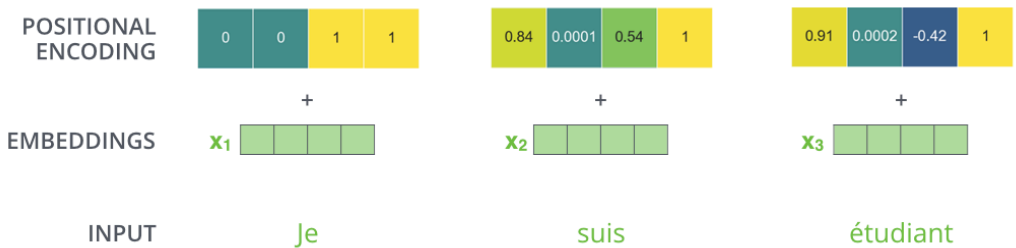
Positional Encoding

Issue: The model does not know the order of words

Positional Encoding

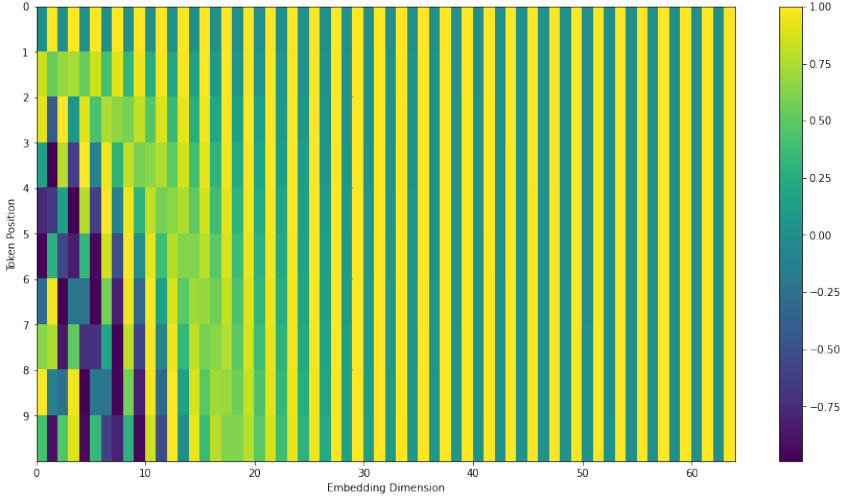
Issue: The model does not know the order of words

Solution: Add different vectors to the sequence of vectors (**positional encoding**)



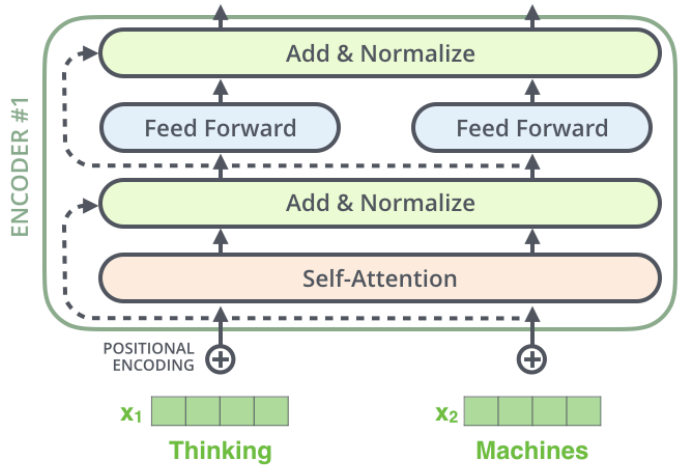
Positional Encodings of first 10 words

If each word is a 64-dim vector. Below are positional encodings of first 10 words

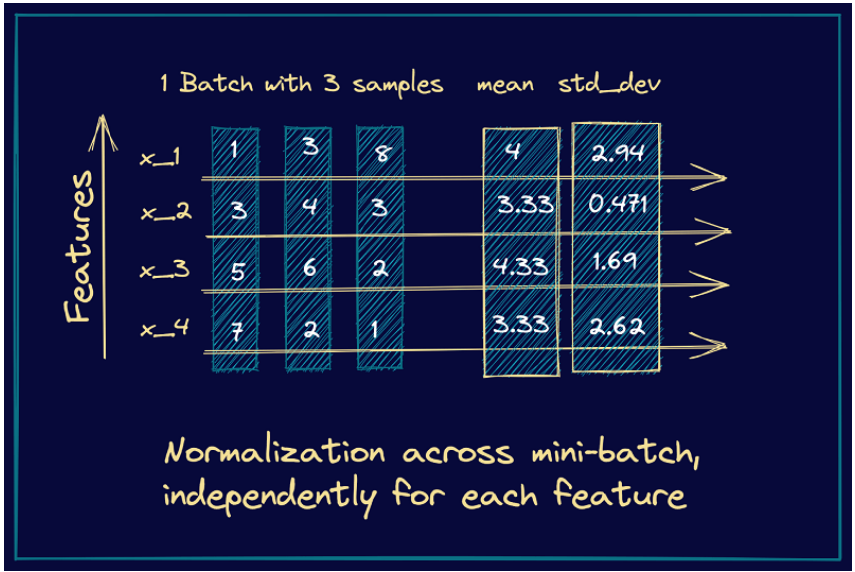


Missing details in the encoder

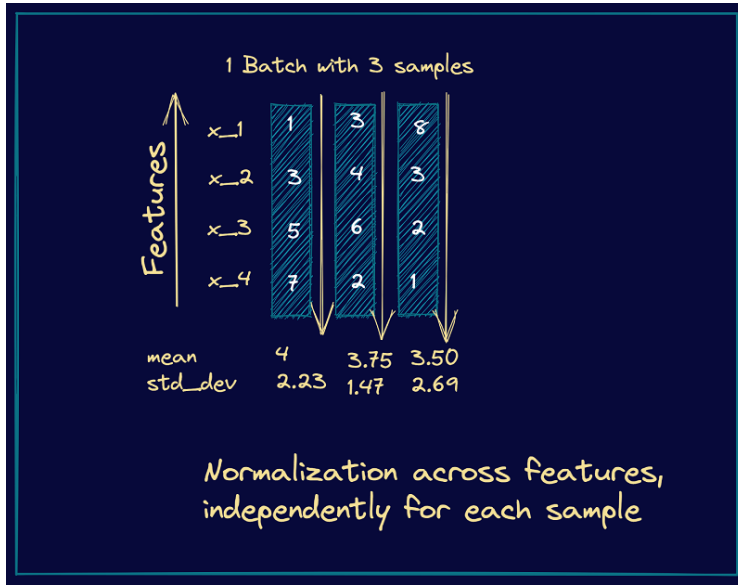
After Self-Attention and Feed-forward, apply **Residual Connection** and **Layer normalization**



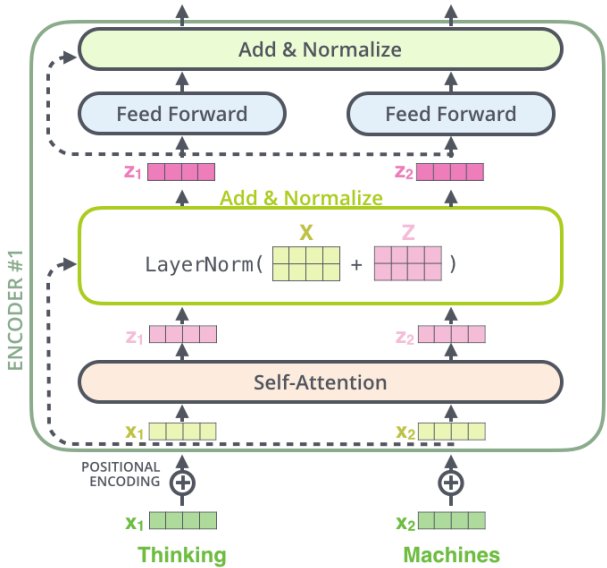
Review: Batch normalization



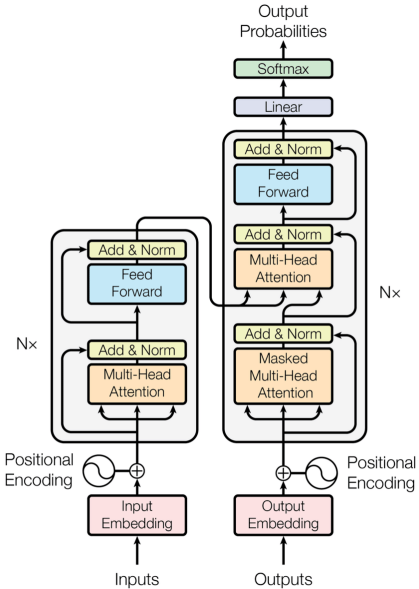
Layer normalization



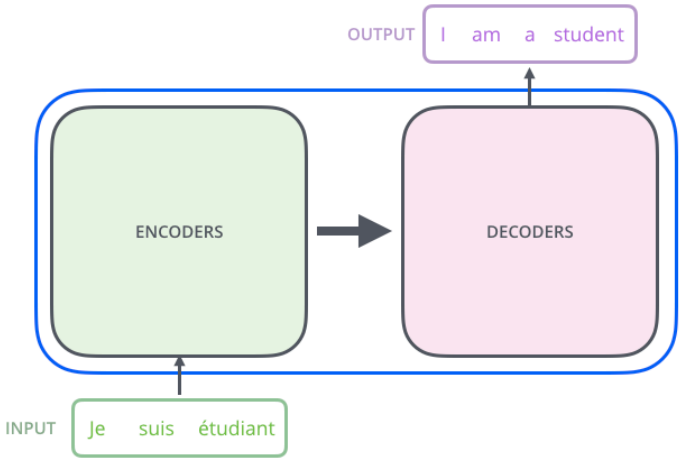
Inside the Encoder



The Decoder: Cross-Attention



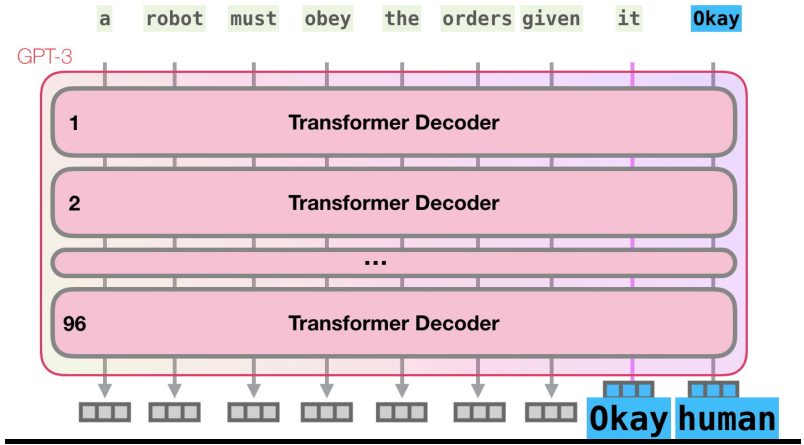
Decoder’s outputs



Transformer-based Language Models

GPT3

96 stacks of Transformer Decoder



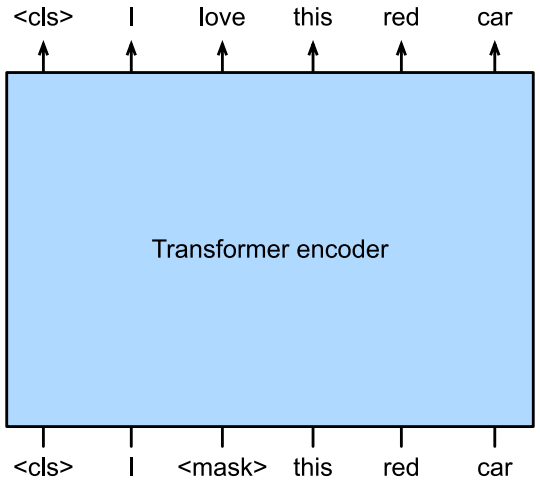
AI DUNGEON

You arrive at a small wooden store with sheet music in the window. The door is locked so you knock. The music suddenly stops and you hear footsteps coming to the door.
"Hello?" A voice says behind the door.
"I'm looking for the book of essence," You say.
"What do you want it for?" The voice answers, a bit of an edge to it.

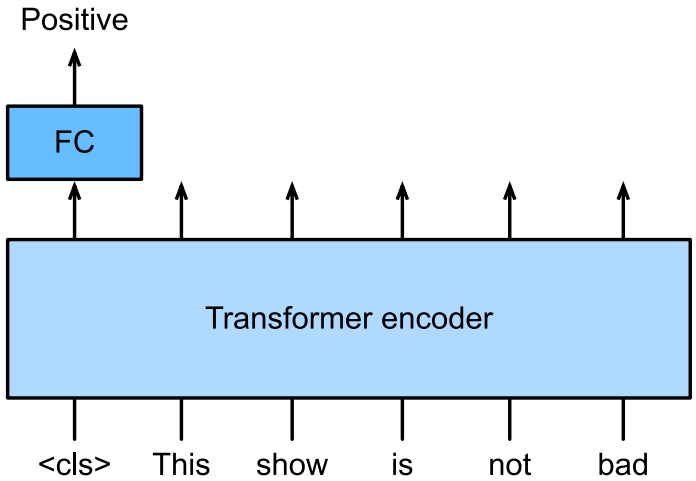
> You tell them you are a humble seeker of knowledge.
"I am a humble seeker of knowledge," You answer.
"Ah, I see. You seek to learn about the essence of reality, and the universe." The voice says. "I know of this book. I am the guard of this library. The only way you can get to it, is if you get past me. Are you up for the challenge?"

> You ask them what the challenge is.
"The challenge is, you have to win a game of chess against me. If you win, then you may try to take the book. I will get the board." The door unlocks and opens.

BERT


<cls>
I
love
this
red
car
Transformer encoder
<cls>
I
<mask>
this
red
car

Fine-tuning BERT



References

- Jay Alammar. The Illustrated Transformer.
  `https://jalammar.github.io/illustrated-transformer`
- Bala Priya. Build Better Deep Learning Models with Batch and Layer Normalization.
  `https://www.pinecone.io/learn/batch-layer-normalization`