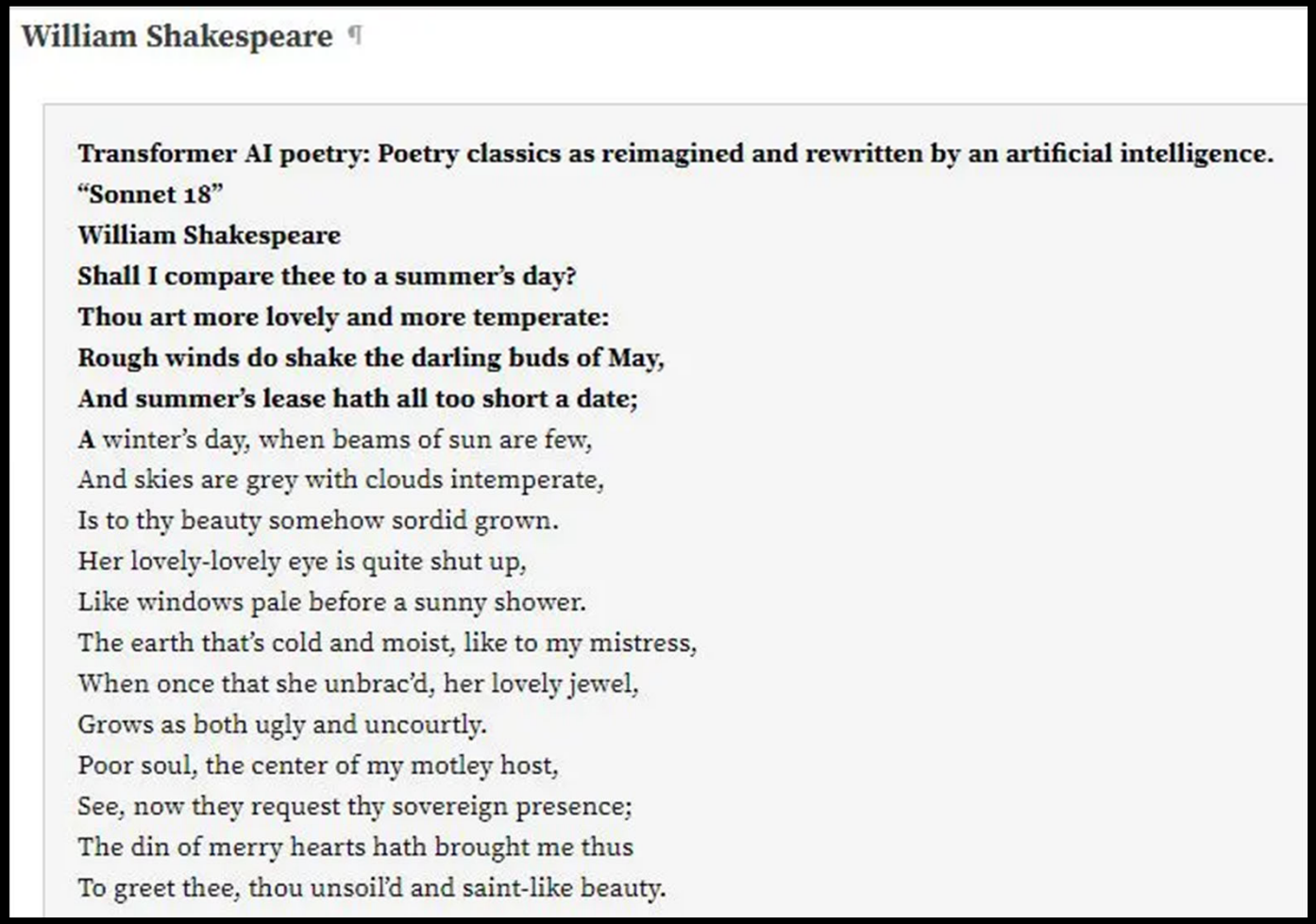
Transformers – GPT3



William Shakespeare ¶

Transformer AI poetry: Poetry classics as reimagined and rewritten by an artificial intelligence.
"Sonnet 18"
William Shakespeare
Shall I compare thee to a summer's day?
Thou art more lovely and more temperate:
Rough winds do shake the darling buds of May,
And summer's lease hath all too short a date;
A winter's day, when beams of sun are few,
And skies are grey with clouds intemperate,
Is to thy beauty somehow sordid grown.
Her lovely-lovely eye is quite shut up,
Like windows pale before a sunny shower.
The earth that's cold and moist, like to my mistress,
When once that she unbrac'd, her lovely jewel,
Grows as both ugly and uncourtly.
Poor soul, the center of my motley host,
See, now they request thy sovereign presence;
The din of merry hearts hath brought me thus
To greet thee, thou unsoil'd and saint-like beauty.

- Can also write computer code from a description, many other uses -- beta.openai.com

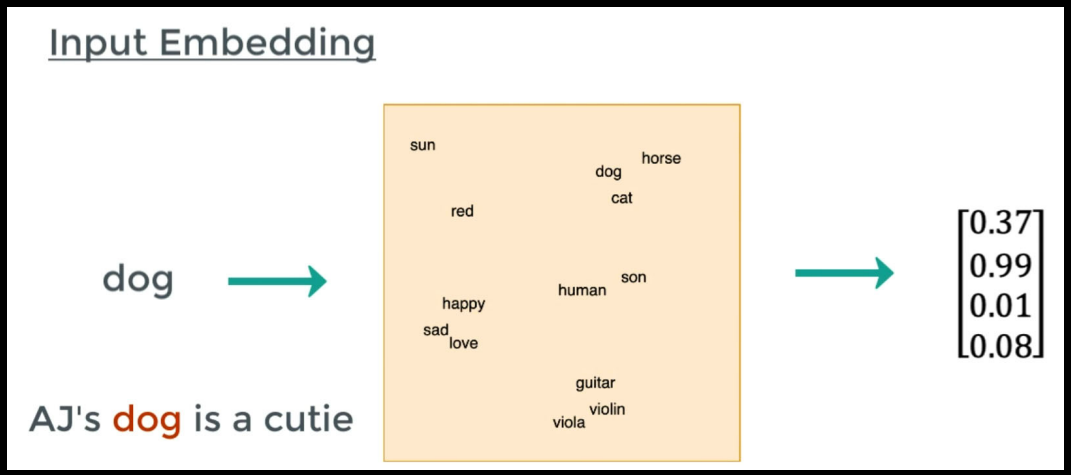
https://machinelearningknowledge.ai/openai-gpt-3-demos-to-convince-you-that-ai-threat-is-real-or-is-it/

Transformers

- How do they work? Several features
  - Encode words *and* their position in sentences
  - Keep track of which other words in general are important context for a given word (the “attention” to other elements in “attention is all you need)
  - Key-value queries: for a specific word, which other words are important
  - Plus lots of deep learning

Transformers – Vaswani et al

- Inputs – for Input embedding,



https://www.youtube.com/watch?v=TQQlZhbC5ps

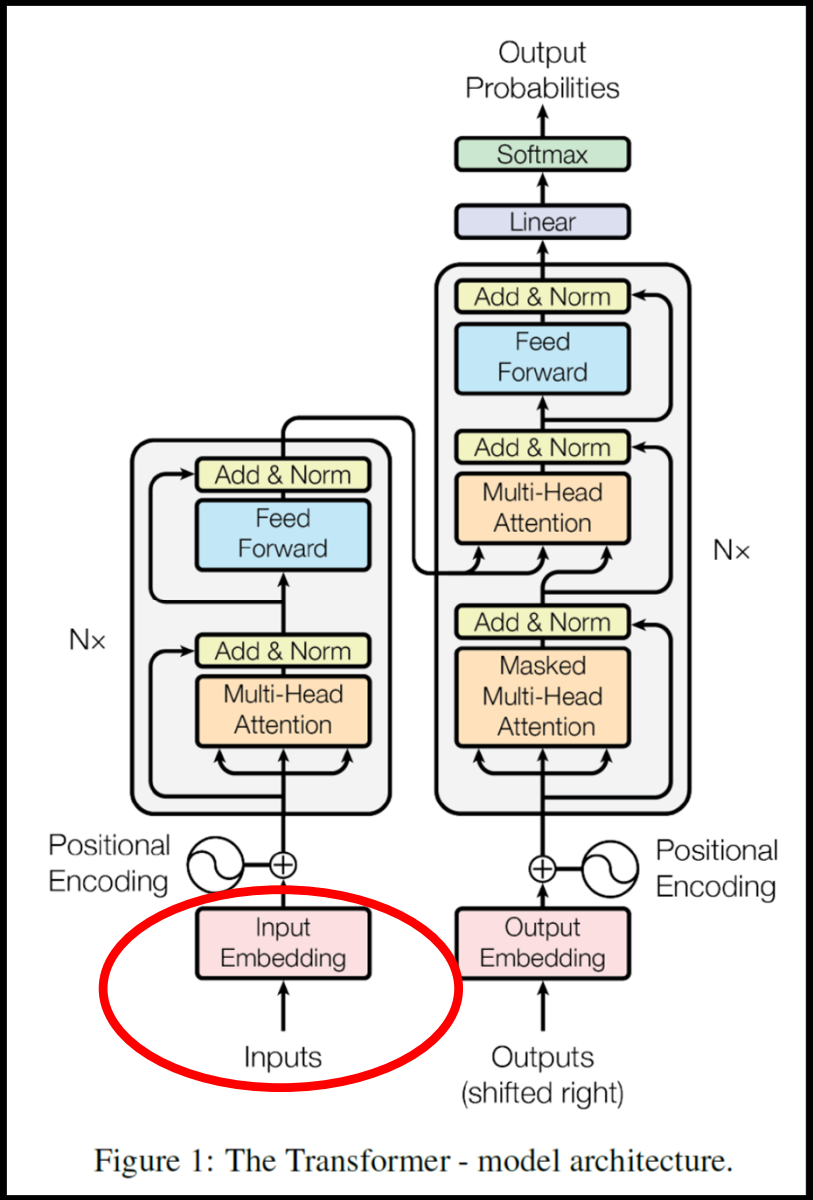
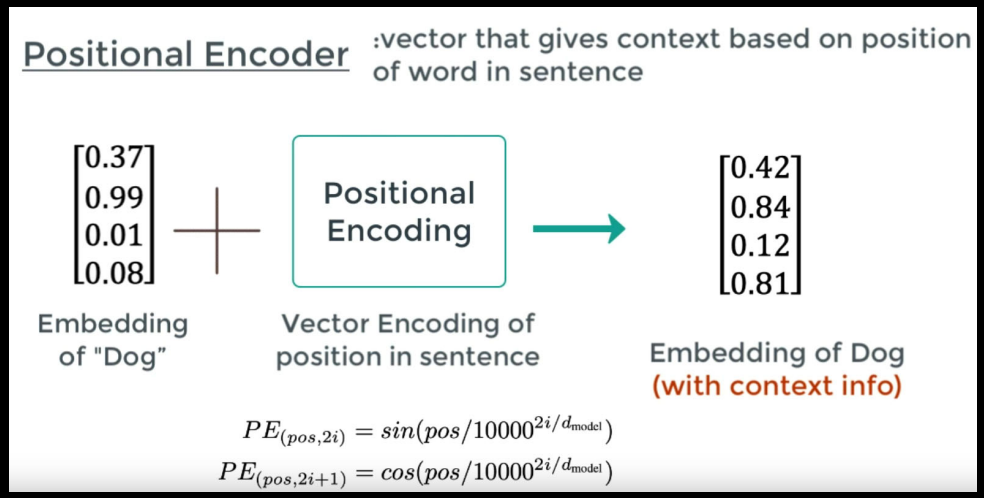


Figure 1: The Transformer - model architecture.

https://arxiv.org/abs/1706.03762

Transformers – Vaswani et al

- Positional encoding
  - i = vector element (e.g. here 1, 2, 3, or 4)
  - Pos = which word of the sentence?



https://www.youtube.com/watch?v=TQQlZhbC5ps

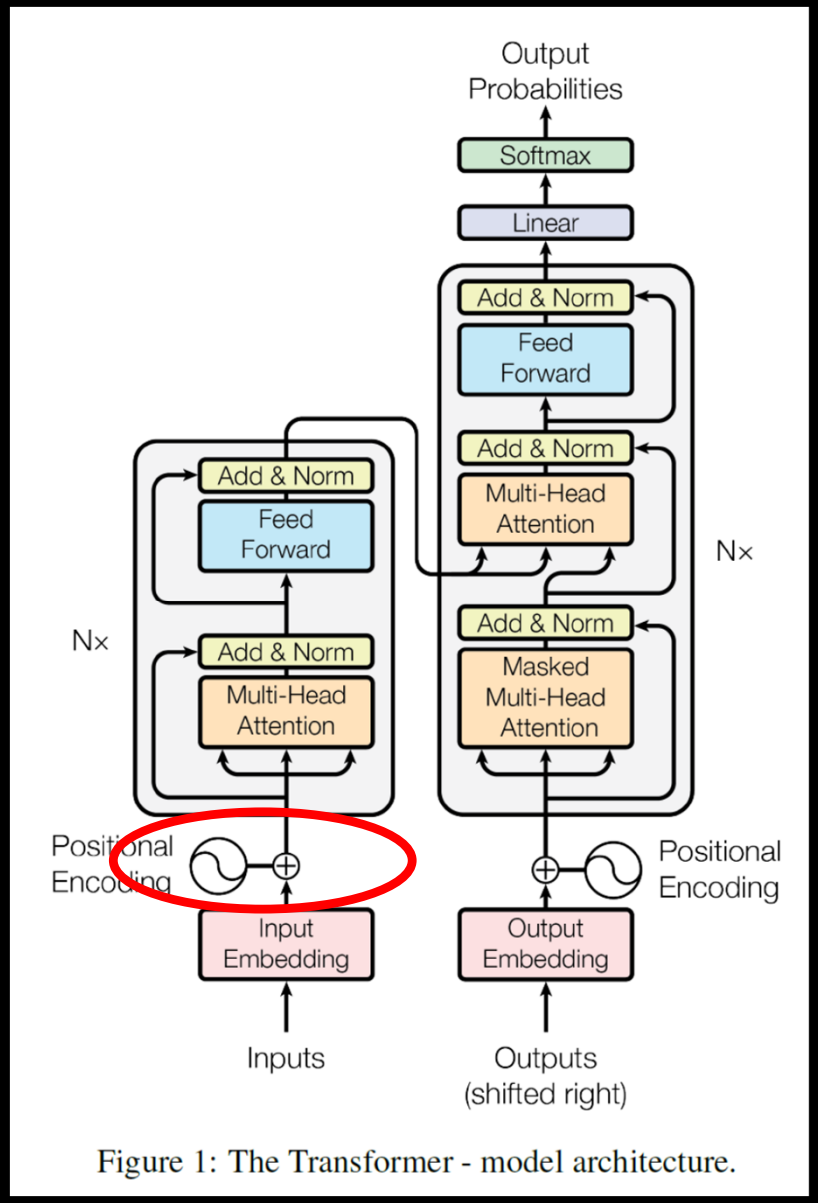
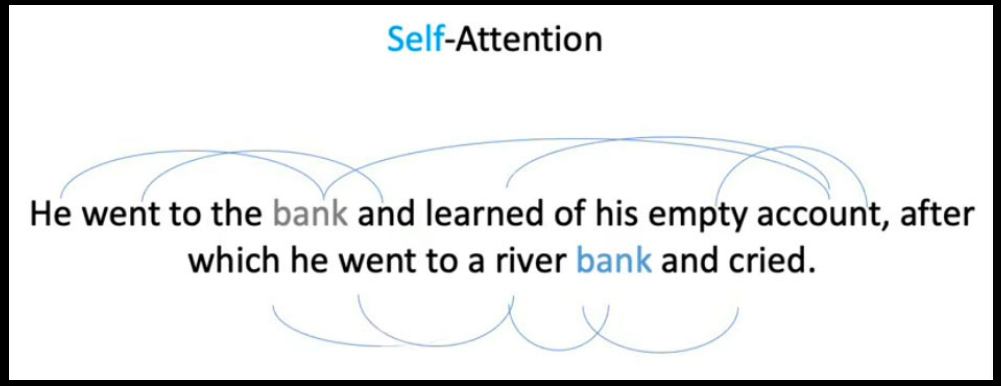


Figure 1: The Transformer - model architecture.

https://arxiv.org/abs/1706.03762

Transformers – Vaswani et al

- Attention
- If you want to understand a word's meaning, you need to process (i.e. attend to) both the word AND the words that are "related" to it
- E.g. "bank" related to "account", or "bank" related to "river"



https://www.youtube.com/watch?v=mMa2PmYJlCo

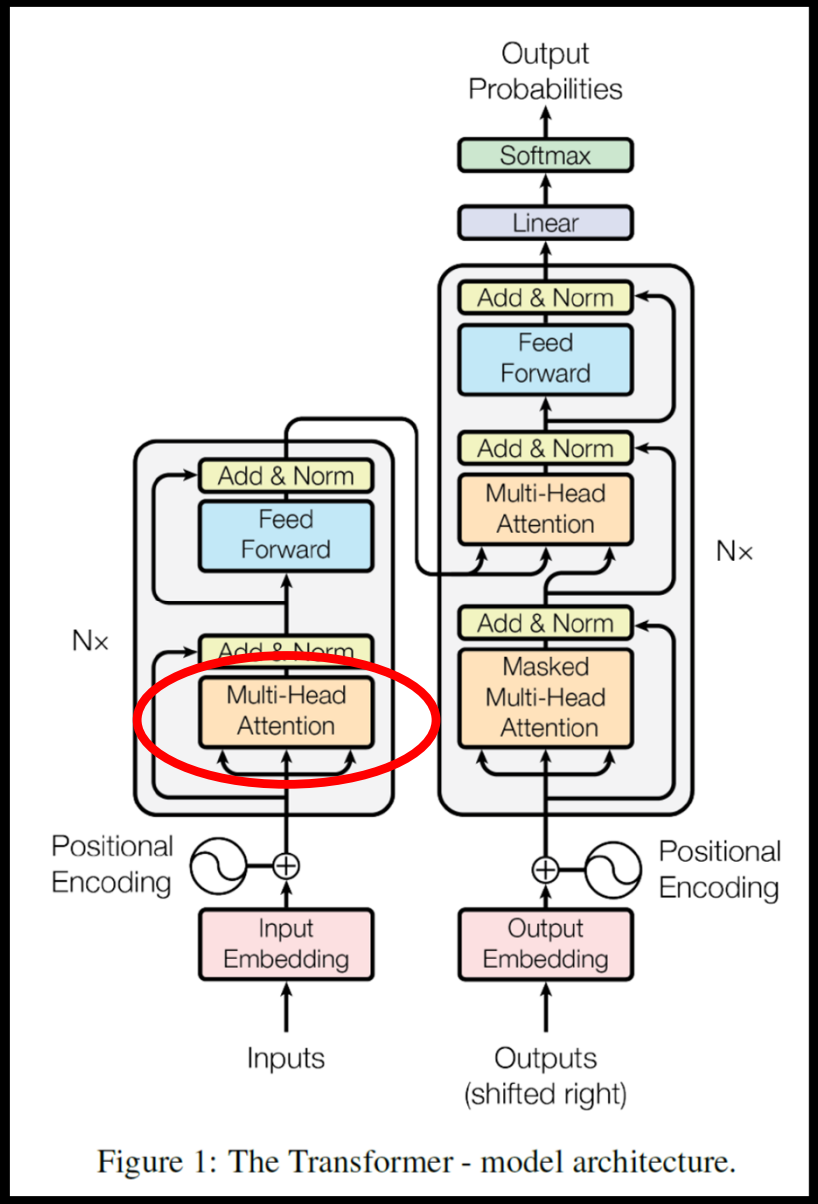


Figure 1: The Transformer - model architecture.

https://arxiv.org/abs/1706.03762

Transformers – Vaswani et al

- Attention
- Can be understood as query → key → value
- Example: "bank" is QUERY (Q), which matches word at position 11 (the KEY or K), whose VALUE (V) is "account"
- Intuition is that each attention head will extract the other inputs in the sequence that are most important and pass them forward as necessary context

- How??

https://www.youtube.com/watch?v=mMa2PmYJlCo

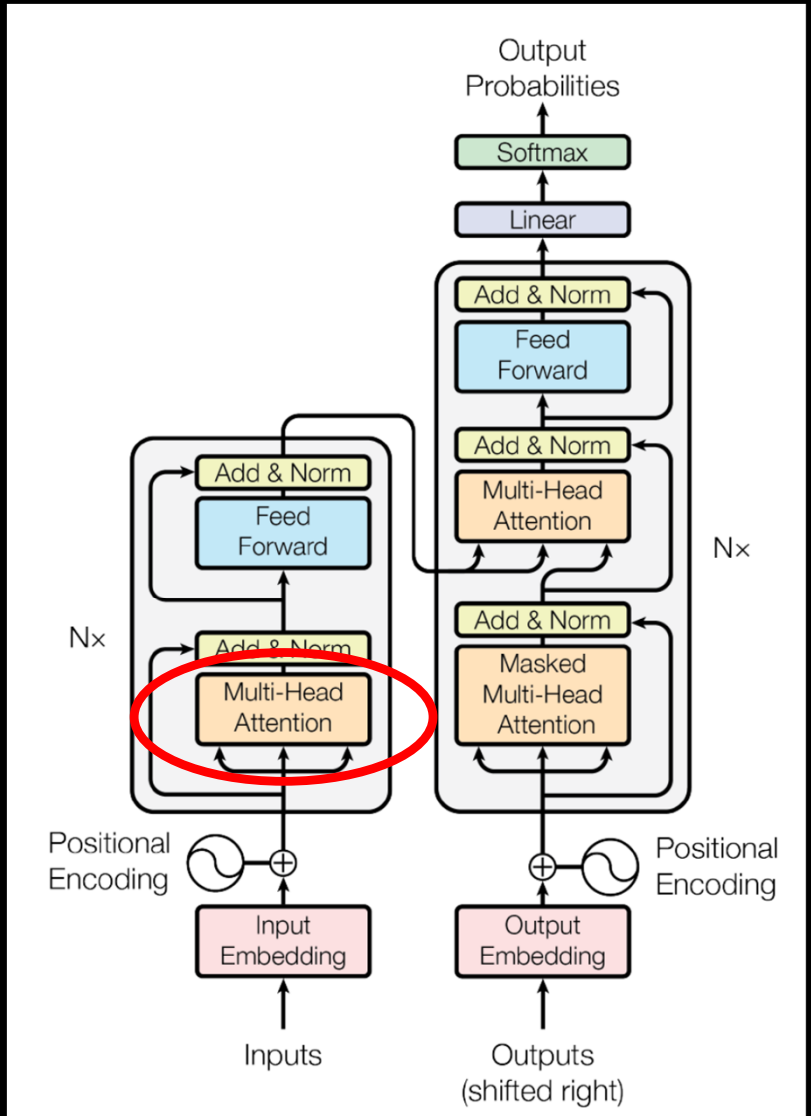


Figure 1: The Transformer - model architecture.

https://arxiv.org/abs/1706.03762

Transformers – Vaswani et al

- Attention
- First, each Query and Key are positionally encoded word vectors. If any two are similar, the angle between them will be small, so the cosine of that angle will be near 1
- → So we compute the cosine of the angle among all pairs with the dot product:

$$\text{similarity}(Q, K) = \frac{Q \cdot K^T}{scaling}$$

- Which yields an attention matrix, e.g.

7 x 7

	When	you	play	the	game	of	thrones
When	89	20	41	10	55	78	59
you	90	98	81	22	87	15	32
play	29	81	95	10	90	30	92
the	10	22	67	12	88	40	89
game	22	70	90	56	98	44	80
of	10	15	30	40	44	44	59
thrones	59	72	92	90	13	59	99

https://www.youtube.com/watch?v=mMa2PmYJlCo

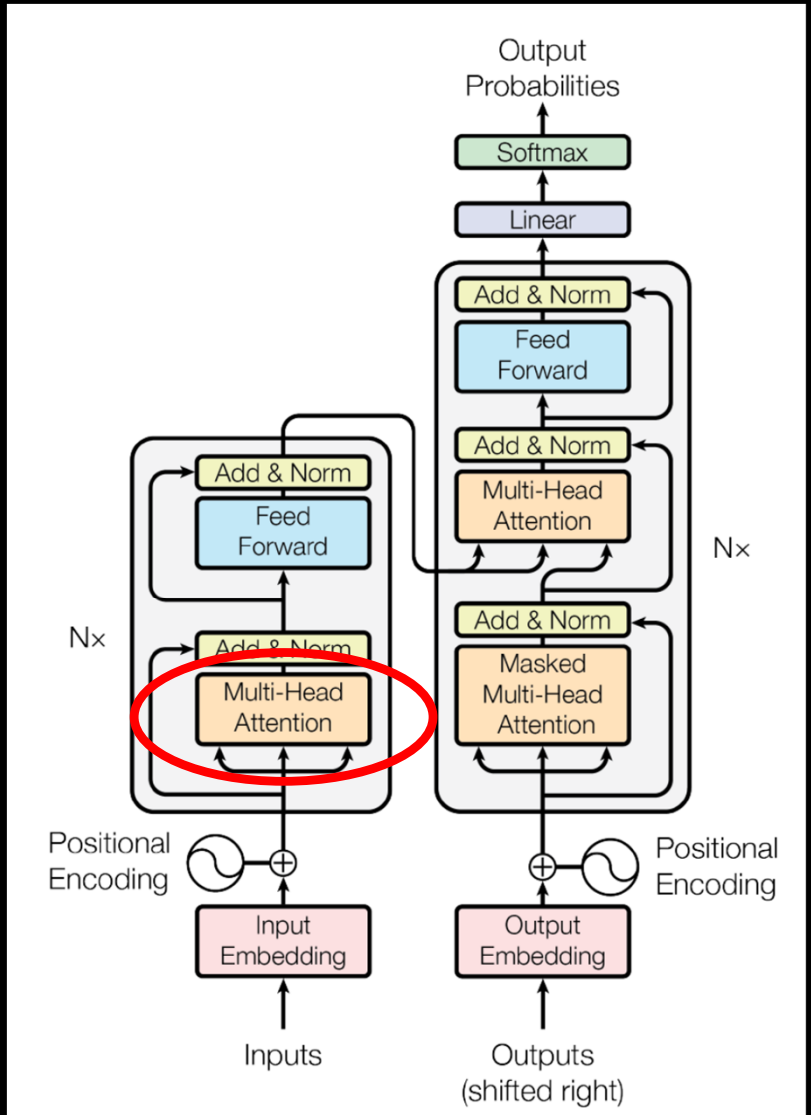
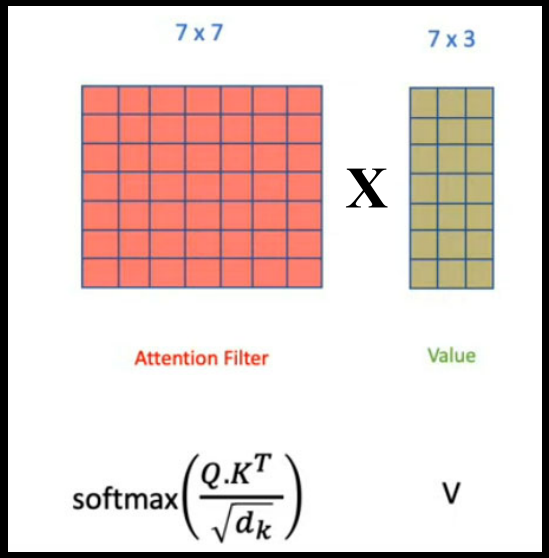


Figure 1: The Transformer - model architecture.

https://arxiv.org/abs/1706.03762

Transformers – Vaswani et al

- Attention
- Next, we matrix multiply the attention filter by the Value (V) for which we want to retrieve related items



- The result retrieves the related positionally encoded word vectors
- Note this is one attention "head", and there can be multiple "heads" trained and running in parallel – just vector concatenate the results
- All trained by backprop!

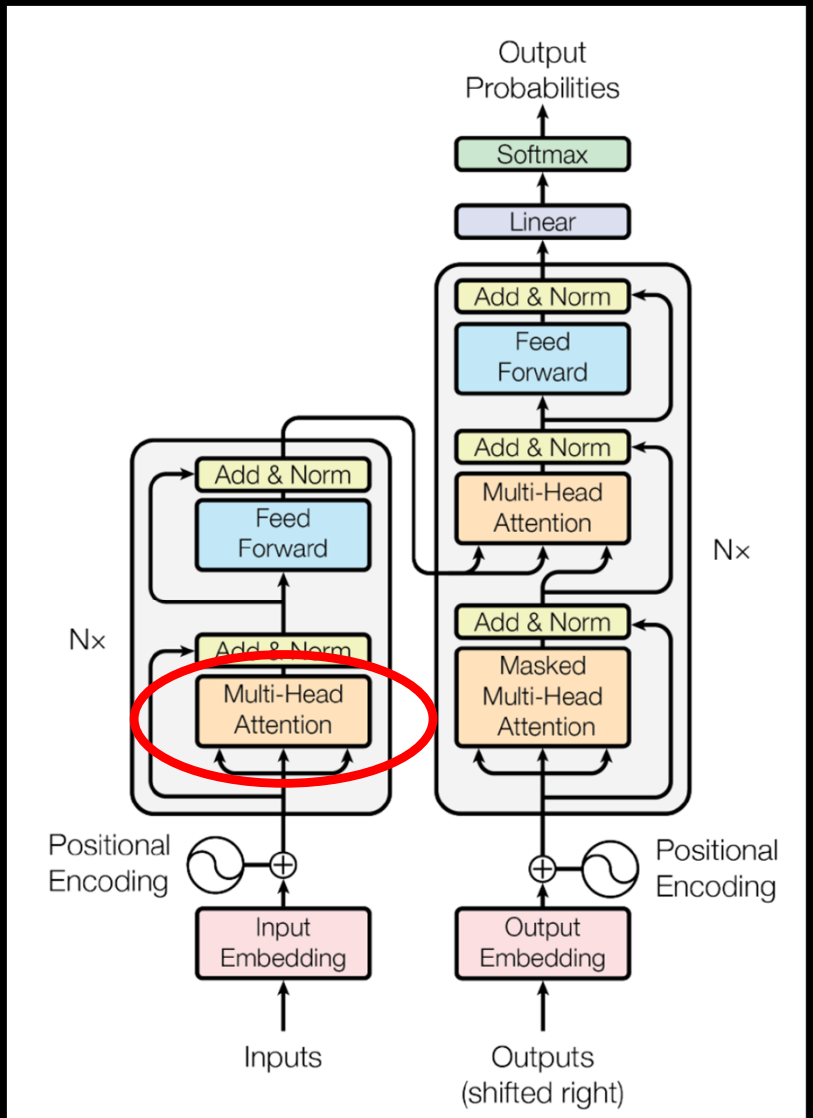


Figure 1: The Transformer - model architecture.

https://www.youtube.com/watch?v=mMa2PmYJlCo

https://arxiv.org/abs/1706.03762

Transformers – Vaswani et al

- Attention
- Next we need to combine the input embedding with the output embedding
- The output is a function of the inputs and the previous outputs (example is translation: you have access to the source language text and all words *so far* that you have translated to the destination language)
- Note the “Output probabilities” (which are compared against desired outputs) at time t become the “Outputs” fed into the bottom right at time t+1
- Note the “Masked” multi-head attention on the output stack only has access to the outputs up to time t. All future outputs are masked out, i.e. set to zero, otherwise no learning would occur
- Same positional encoding and multi-head attention processes the output embedding

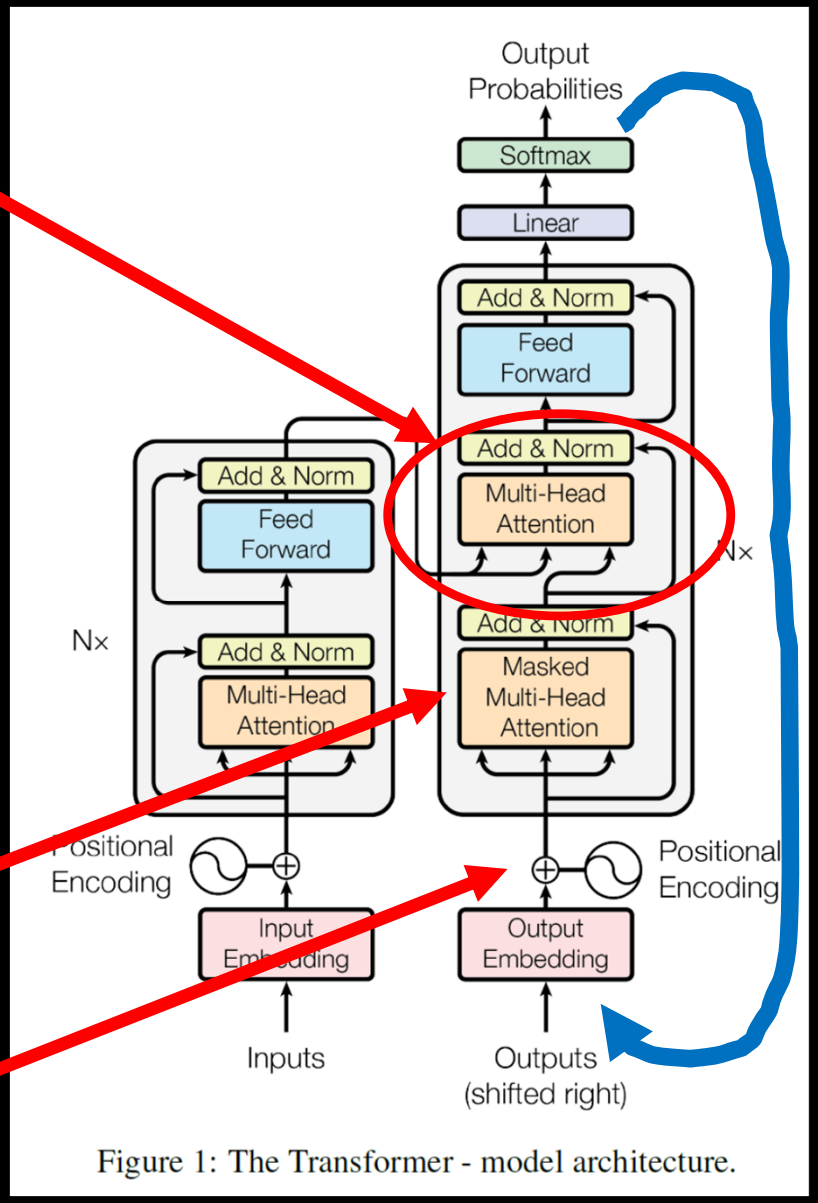


Figure 1: The Transformer - model architecture.

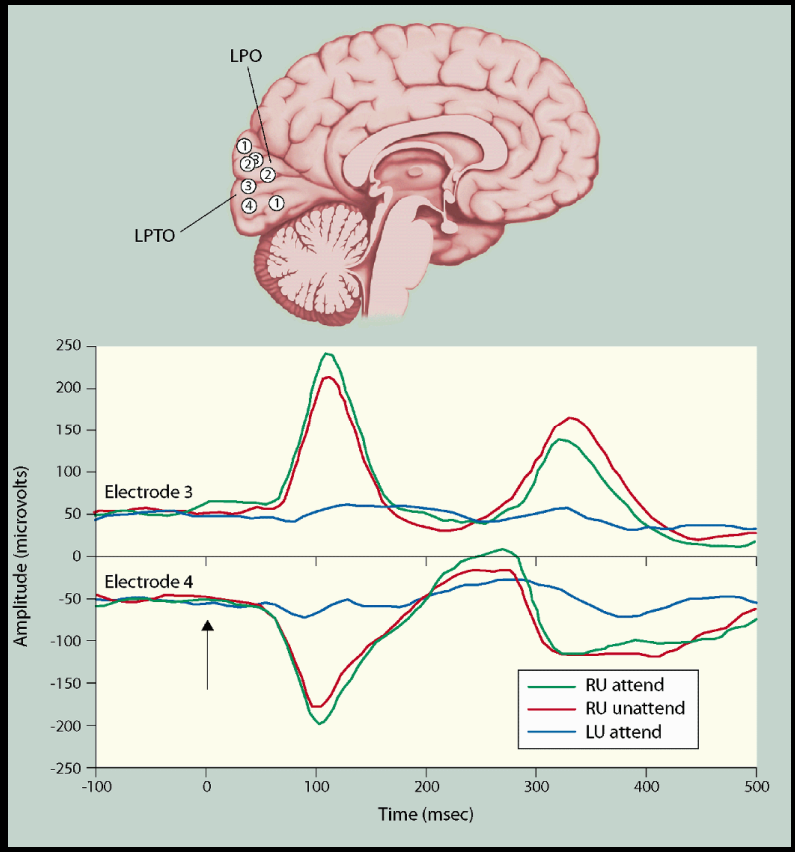
https://arxiv.org/abs/1706.03762

Transformers

- Could actual neurons do what transformers are doing?

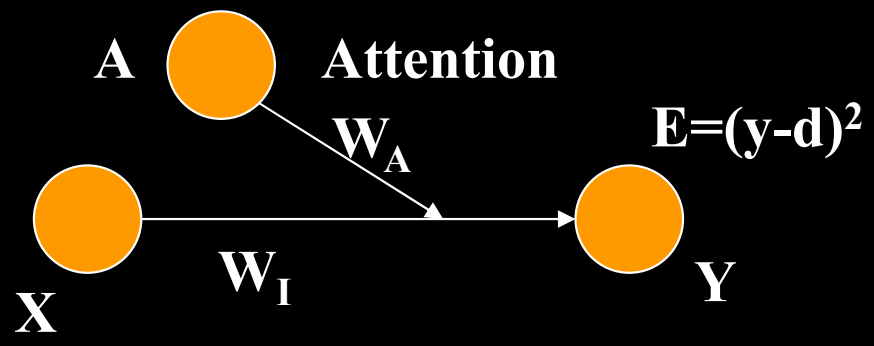
Transformers – biology?

- Could actual neurons do what transformers are doing?
- Input embedding, positional encoding seem plausible
- Attention is ubiquitous in the brain
- Simple multiplication of stimulus signal by attention signal > 1
- BUT how could backprop train neurons like a transformer??



Transformers – biology?

- BUT how could backprop train neurons like a transformer??
  - $\frac{\partial E}{\partial W}$ with multiplicative attentional signal requires product rule (same issue with LSTM)



- Here the delta rule must be multiplied by the attentional signal (potentially N time steps in the past!)

$$Y = (X * W_I) \bullet (A * W_A)$$

Chain rule:

$$\frac{\partial E}{\partial W_I} = 2E * \frac{\partial Y}{\partial W_I} = 2E * X * (A * W_A)$$

- Feedback alignment? R2N2? Perhaps – would need to *ALSO* reconstruct attentional inputs from previous time steps!!

$$\frac{\partial E}{\partial W_A} = 2E * \frac{\partial Y}{\partial W_A} = 2E * A * (X * W_I)$$

Transformers – GPT3

- Beta.openai.com
- Transformers: https://www.youtube.com/watch?v=TQQlZhbC5ps
- https://colab.research.google.com/github/pytorch/tutorials/blob/gh-pages/_downloads/transformer_tutorial.ipynb

Example – Stock market prediction

- Can we use RNNs, like LSTM, to predict the stock market and make $$$?
- Short answer: It's very hard
- Long answer: The efficient market hypothesis and random walk theory suggest that markets immediately react to new information and are otherwise very unpredictable, which makes it difficult to predict asset price changes (at least in short term)
- Longer answer: Garbage in = Garbage out. If you're going to make a prediction, on what basis? What input will you use? Previous asset price history? What else?

Example – Generative models

- Twitter @DeepDrumpf