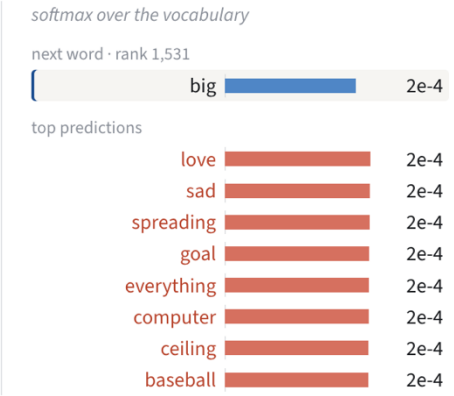
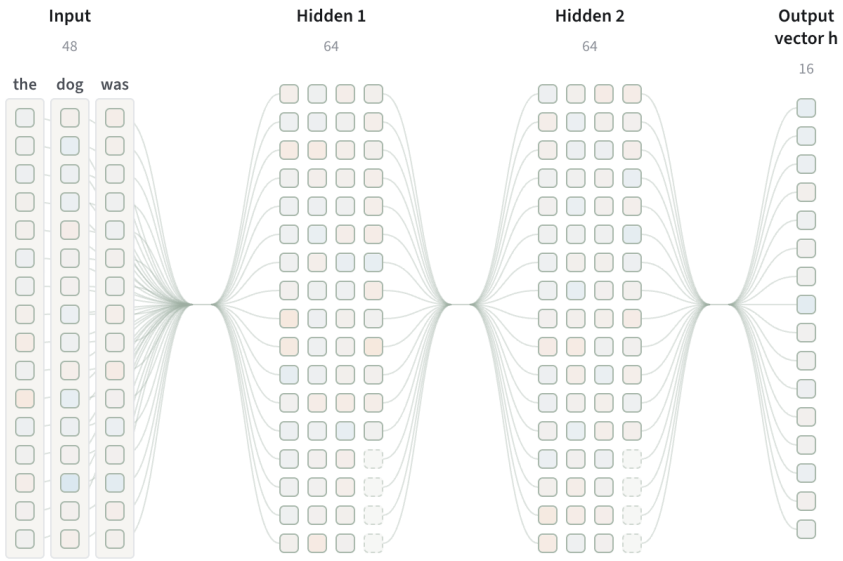
Transformers and Large Language Models

Recap: Next word prediction (*small* language model)





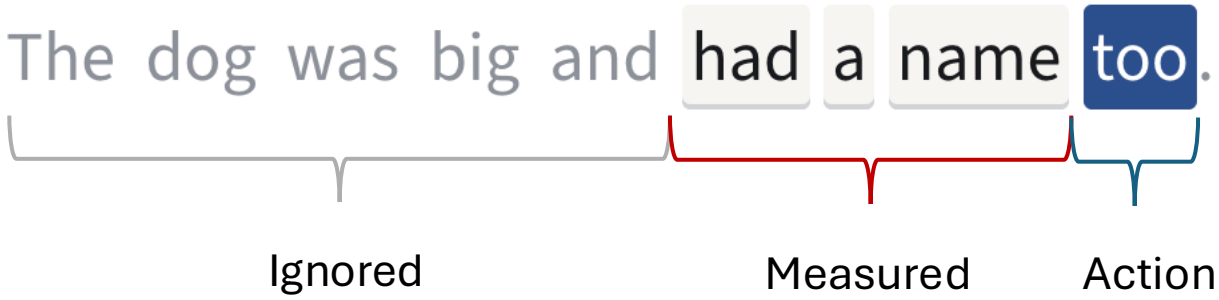




Neural networks assume state is represented by a ***fixed number*** of ***continuous measurements***

One solution to the fixed number restriction:

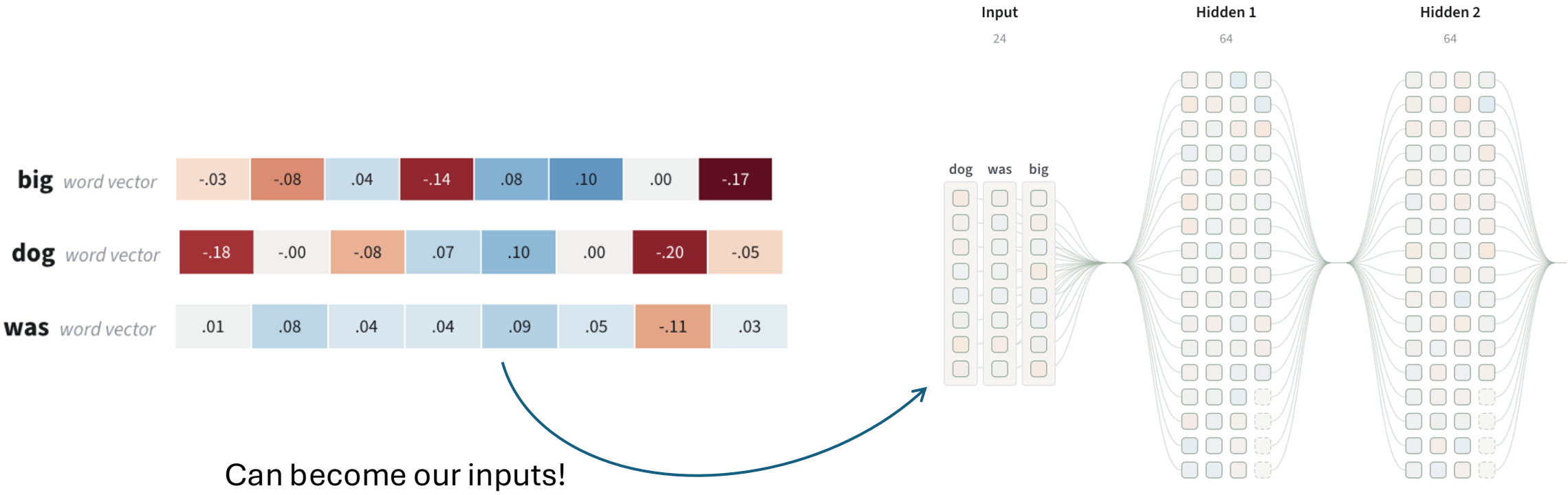
- Only measure the last N words



- *We'll see a better solution soon…*

Word embeddings

Idea: Represent each possible word as a set of continuous measurements (*embeddings)*



Byte - Pair embedding - GPT 3

Tokens	Characters
58	301

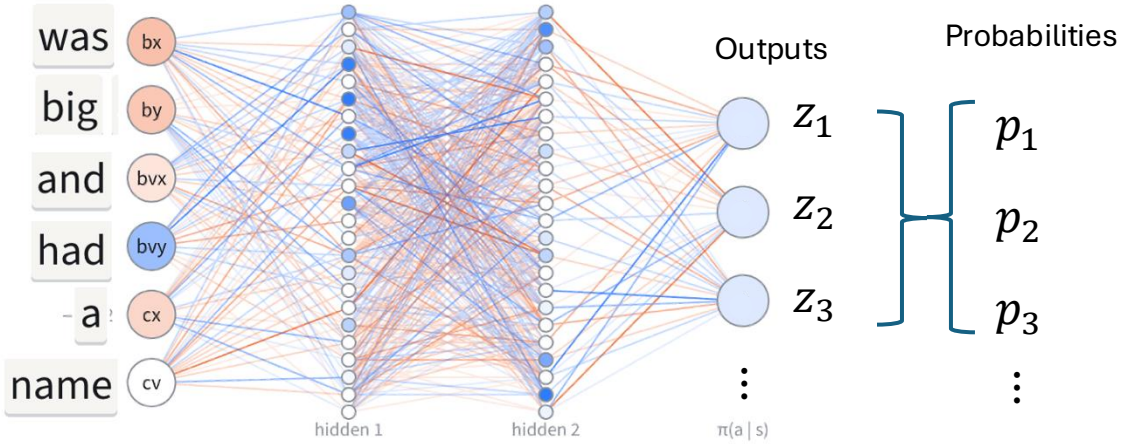
OpenAI's large language models (sometimes referred to as GPT's) process text using tokens, which are common sequences of characters found in a set of text. The models learn to understand the statistical relationships between these tokens, and excel at producing the next token in a sequence of tokens.

```
[5109, 15836, 596, 3544, 4221, 4211, 320, 57753, 14183, 311, 439, 480,
2898, 596, 8, 1920, 1495, 1701, 11460, 11, 902, 527, 4279, 24630, 315,
5885, 1766, 304, 264, 743, 315, 1495, 13, 578, 4211, 4048, 311, 3619,
279, 29564, 12135, 1990, 1521, 11460, 11, 323, 25555, 520, 17843, 279,
1828, 4037, 304, 264, 8668, 315, 11460, 13]
```

Review: ***Softmax*** activation

Special activation that maps a set of neurons to numbers that:

- Are positive
- Sum to 1



Positive

$$p_i = \frac{e^{z_i}}{\sum_{j=1}^{n} e^{z_j}}$$

Sum to 1

We want to scale up

Questions to answer:

- How do we (better) handle long context?
- Can we define a reward for RL?
- Even if *can* get this all to work, *should we?*

How well do you feel you understand how a LLM / transformer works?

A. Very well (*I could build one today*)

B. Pretty well (*I could explain the basics)*

C. Okay (*I'm confused on the details)*

D. Not very well (*I'm excited for today's lecture)*

The tipping point for language models (maybe)

Paper presented at the 2017 conference Neural Information Processing Systems

- Authors worked at Google
- Introduced the Transformer

Attention Is All You Need

Ashish Vaswani* Google Brain avaswani@google.com

Noam Shazeer* Google Brain noam@google.com

Niki Parmar* Google Research nikip@google.com

Jakob Uszkoreit* Google Research usz@google.com

Llion Jones* Google Research llion@google.com

Aidan N. Gomez* † University of Toronto aidan@cs.toronto.edu

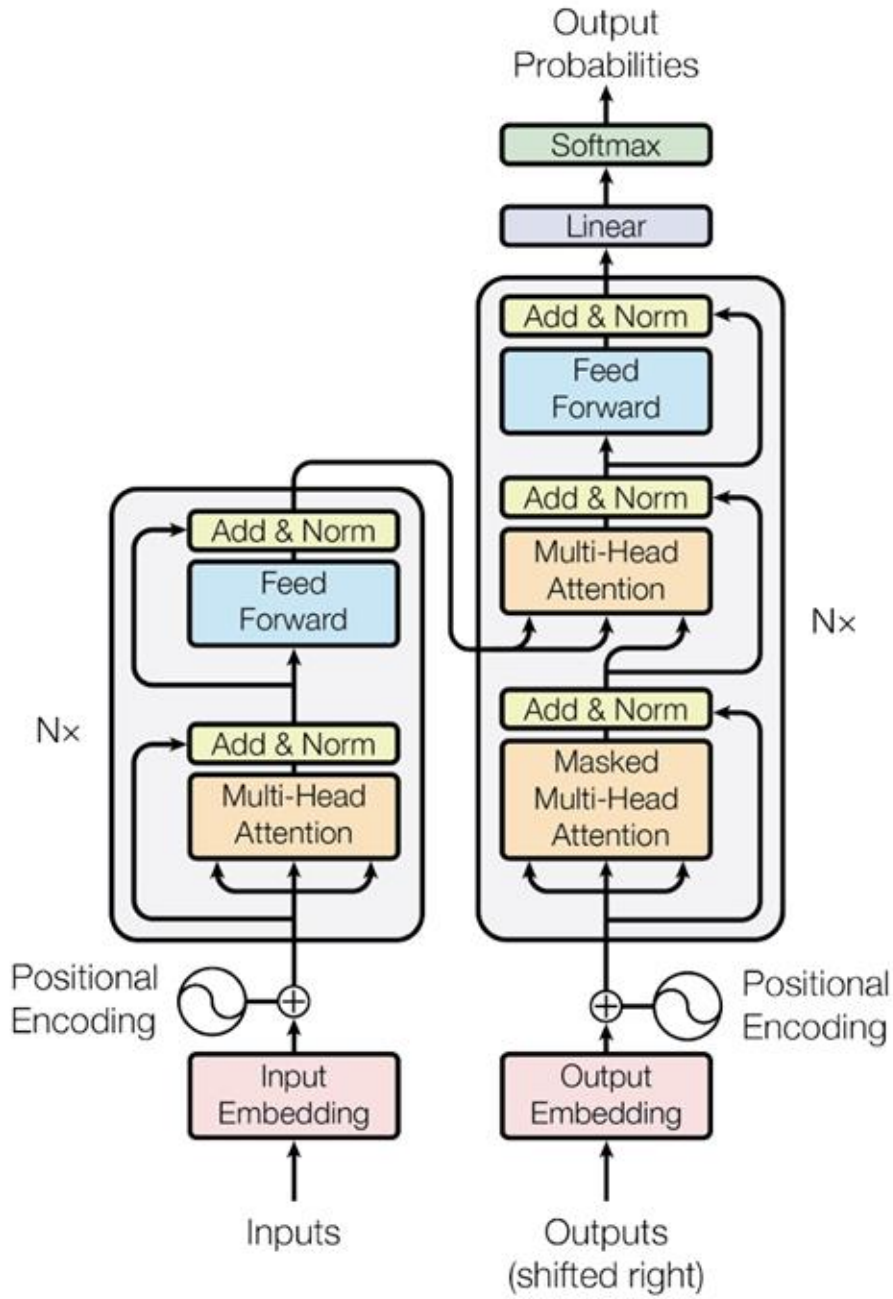
Łukasz Kaiser* Google Brain lukaszkaiser@google.com

Illia Polosukhin* ‡ illia.polosukhin@gmail.com

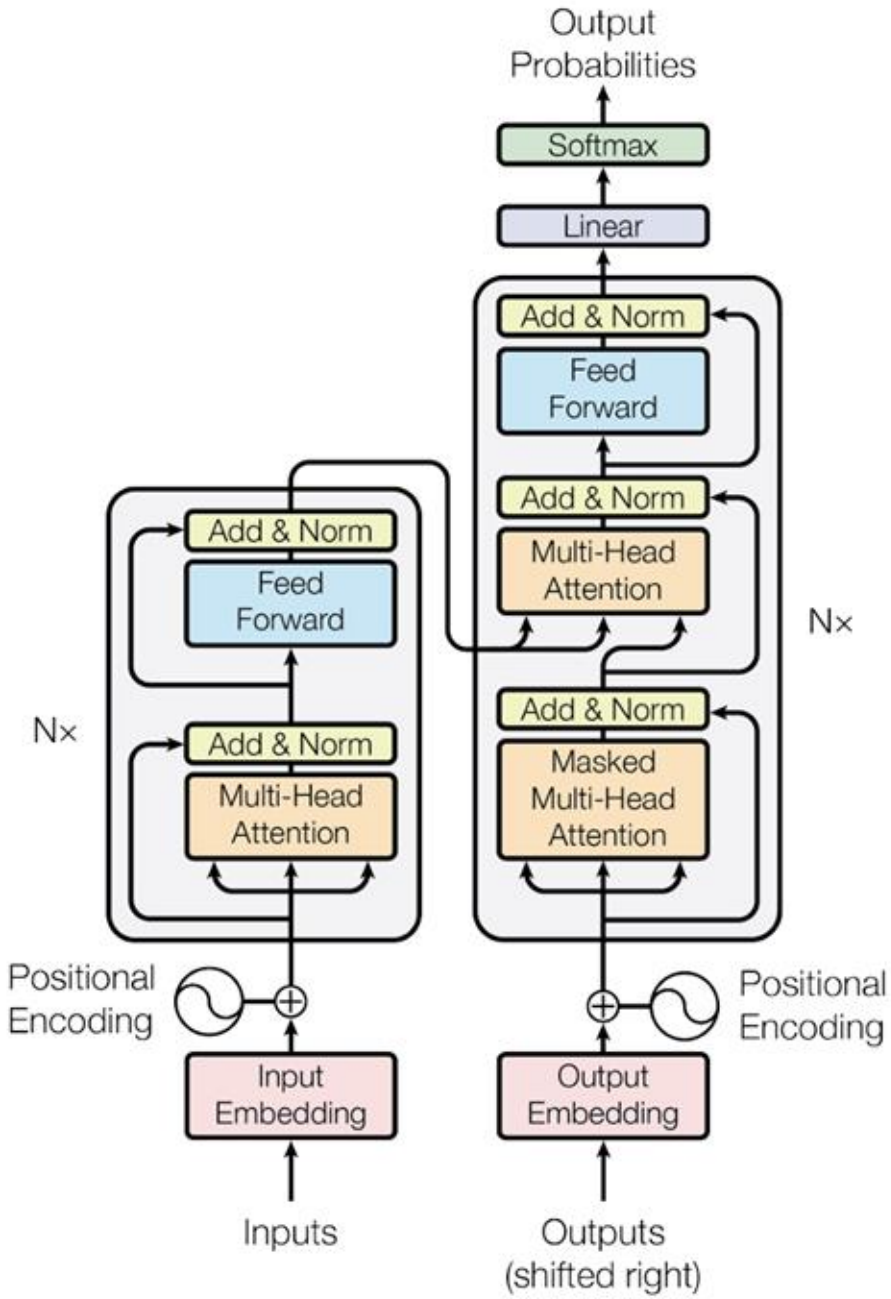
Abstract

The dominant sequence transduction models are based on complex recurrent or convolutional neural networks that include an encoder and a decoder. The best performing models also connect the encoder and decoder through an attention mechanism. We propose a new simple network architecture, the Transformer, based solely on attention mechanisms, dispensing with recurrence and convolutions entirely. Experiments on two machine translation tasks show these models to be superior in quality while being more parallelizable and requiring significantly less time to train. Our model achieves 28.4 BLEU on the WMT 2014 English-to-German translation task, improving over the existing best results, including ensembles, by over 2 BLEU. On the WMT 2014 English-to-French translation task, our model establishes a new single-model state-of-the-art BLEU score of 41.8 after training for 3.5 days on eight GPUs, a small fraction of the training costs of the best models from the literature. We show that the Transformer generalizes well to other tasks by applying it successfully to English constituency parsing both with large and limited training data.

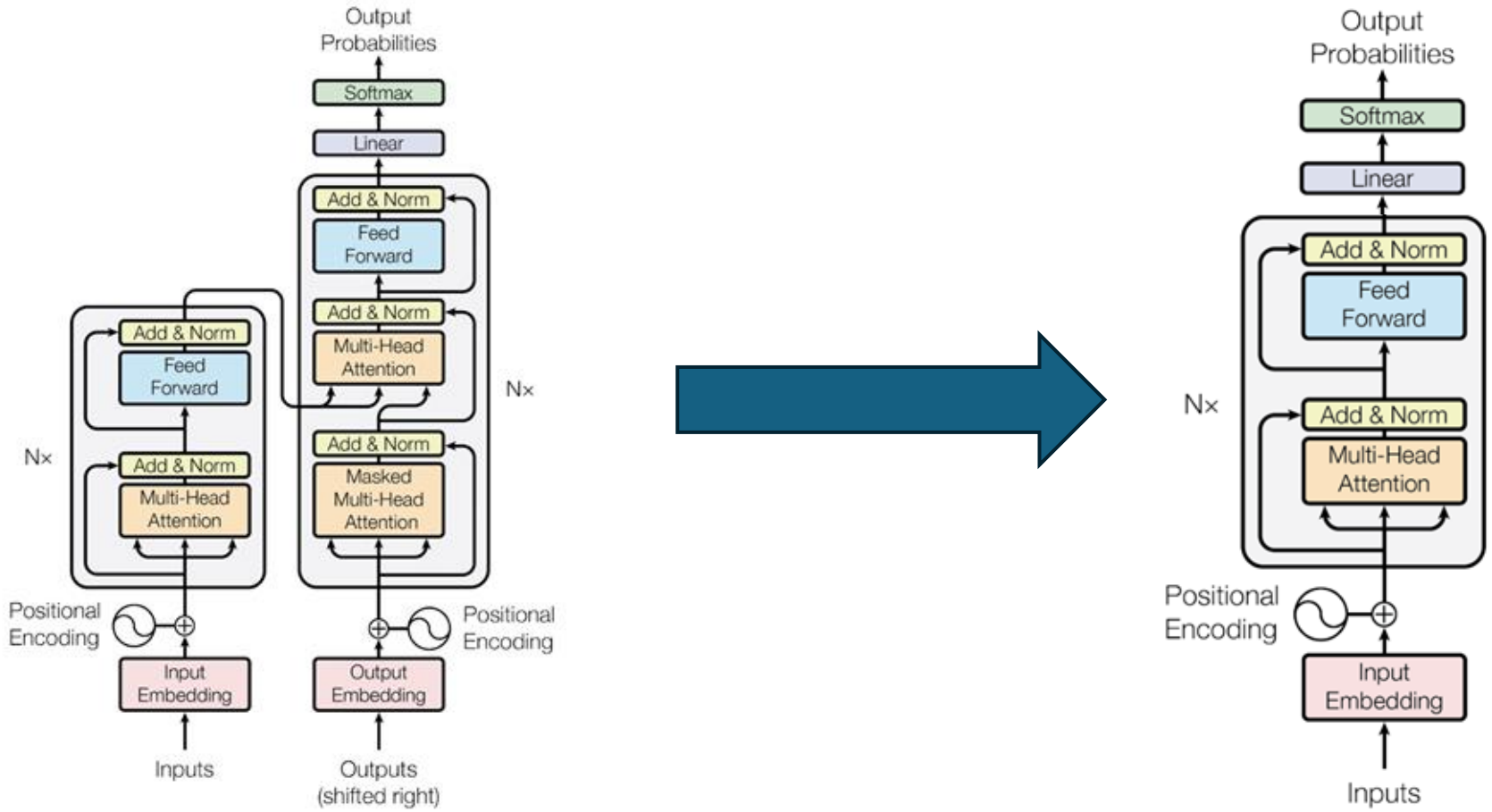
The Transformer architecture



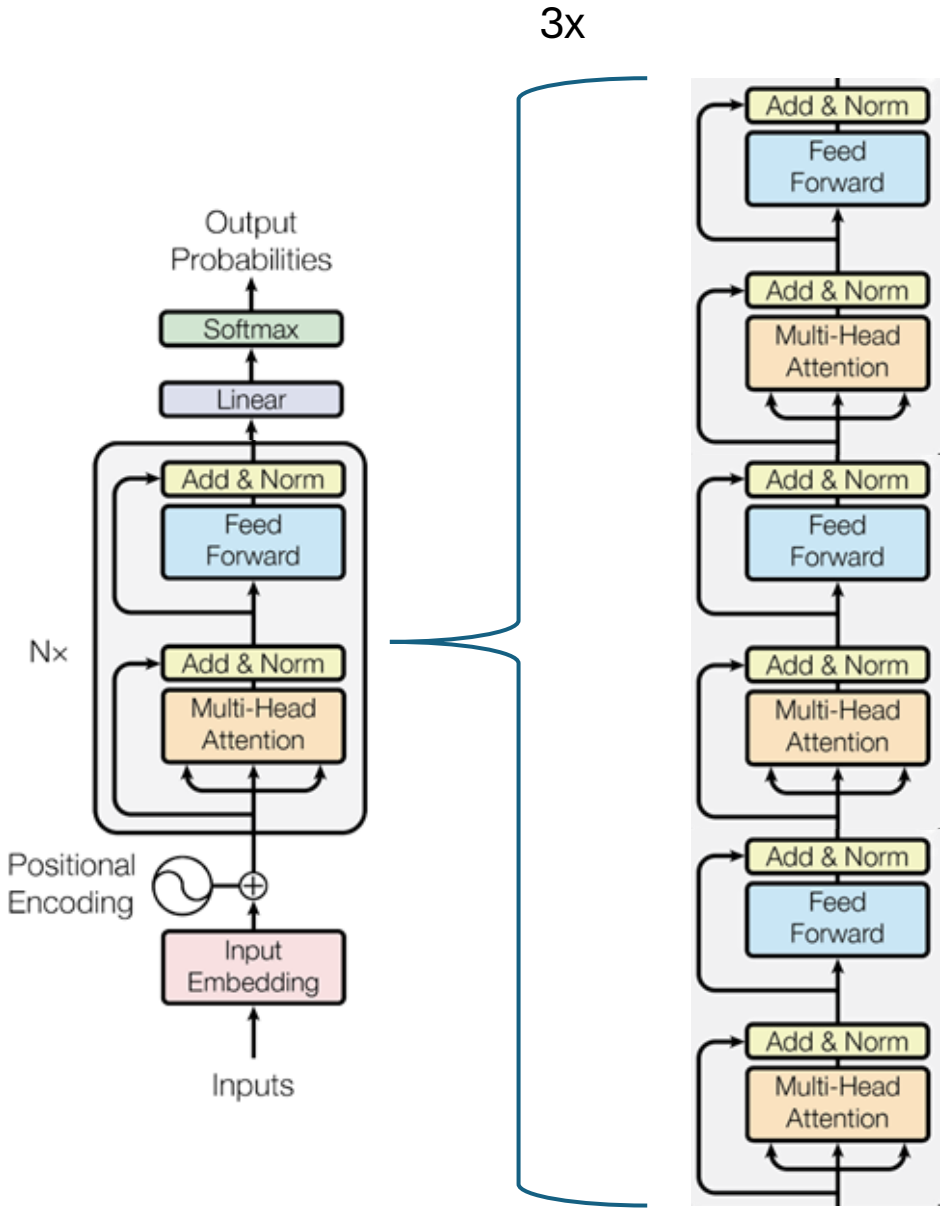
The encoder and decoder



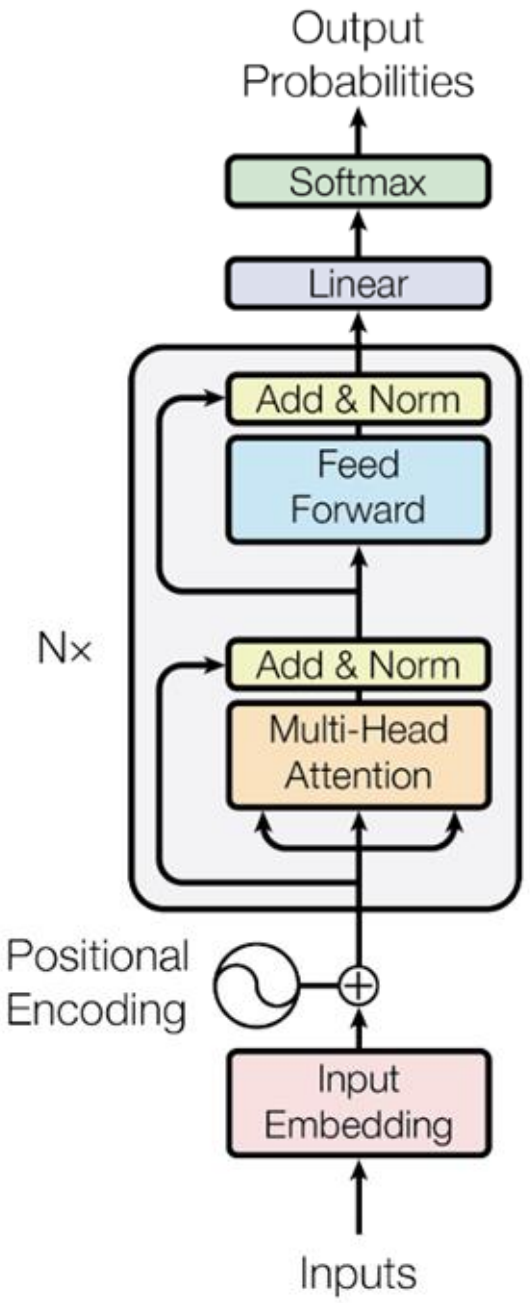
A simplified (encoder-only) model



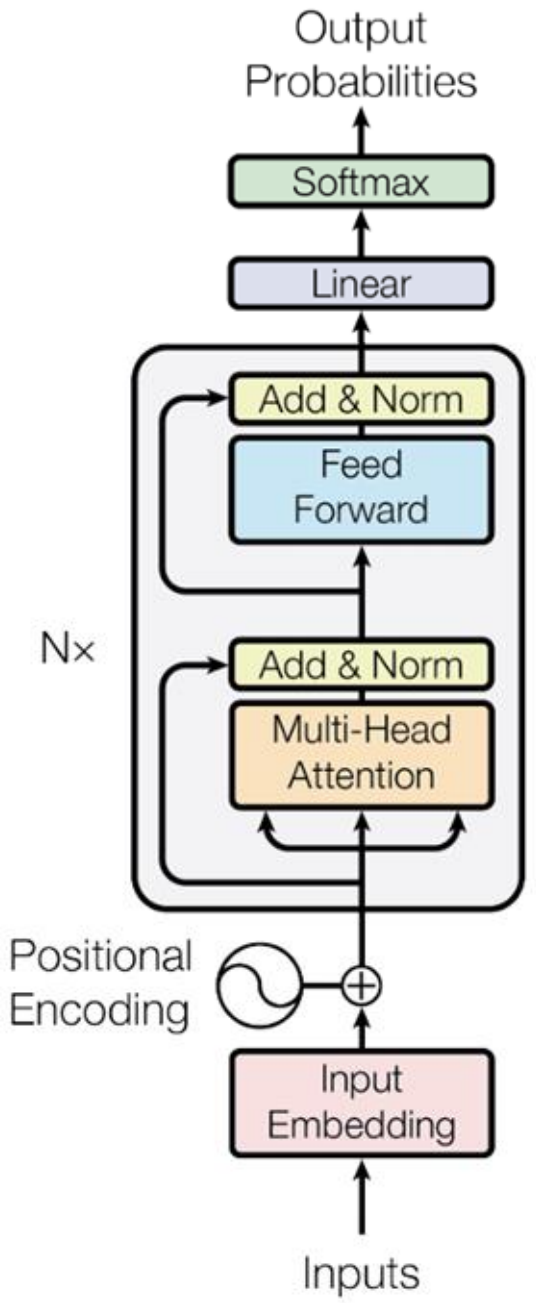
First mystery: N× ?



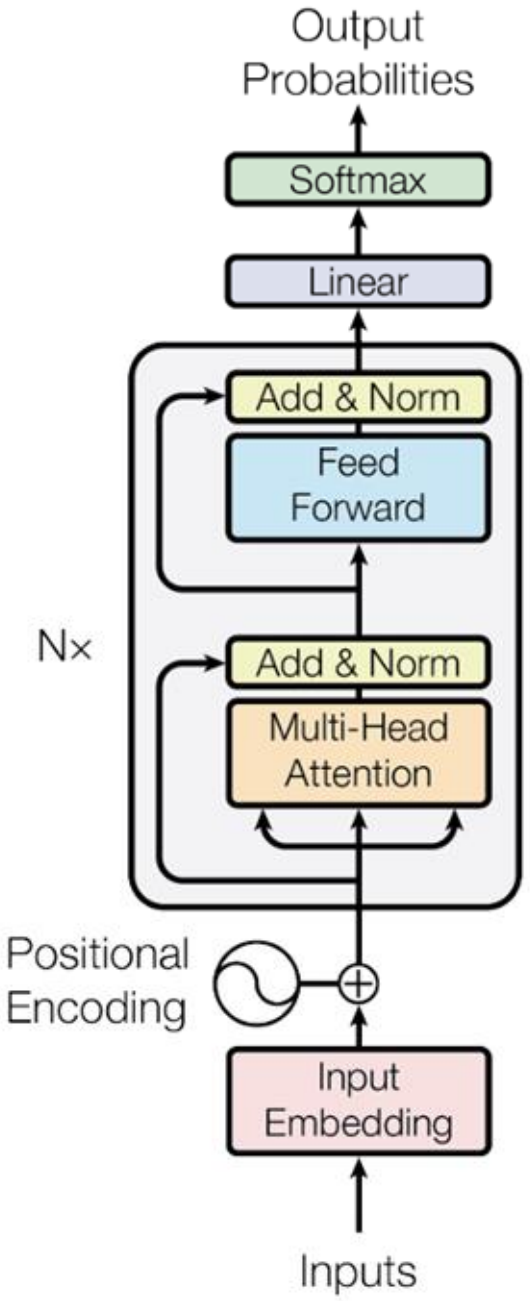
What do we already recognize?



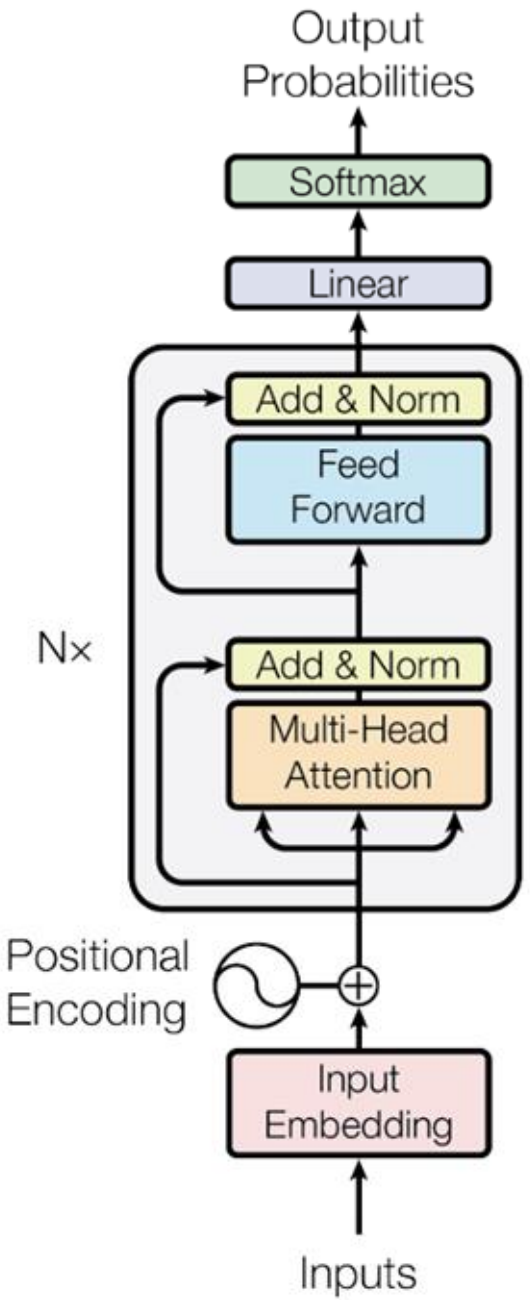
Input and embedding



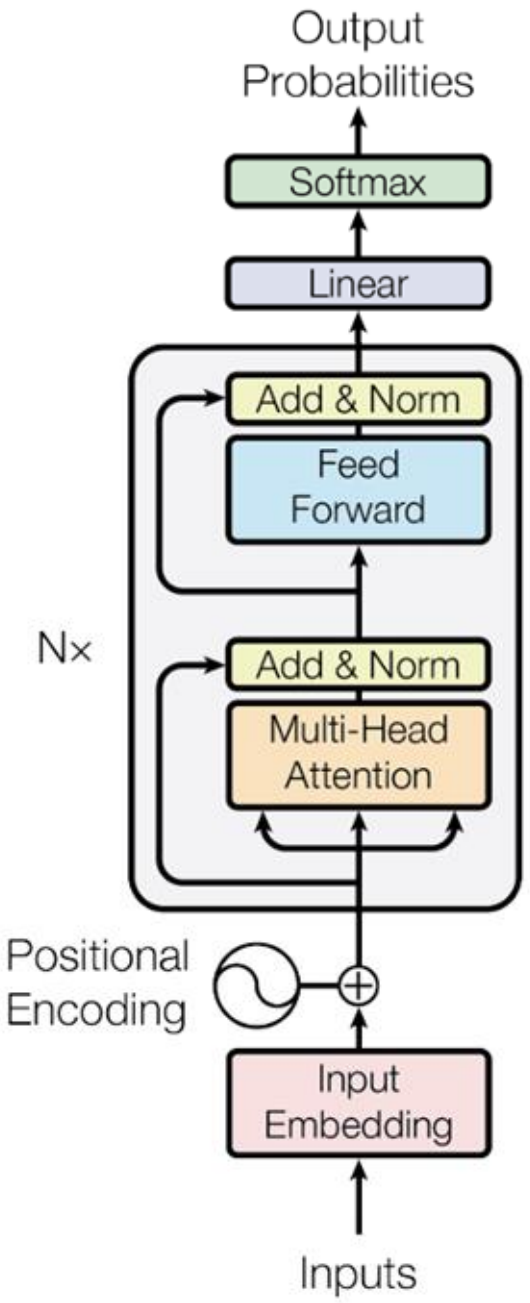
“Feed forward” layer



“Linear” and softmax layers



What do we need to understand?



“Multi-head Attention”

Attention Is All You Need

Ashish Vaswani* Google Brain avaswani@google.com

Noam Shazeer* Google Brain noam@google.com

Niki Parmar* Google Research nikip@google.com

Jakob Uszkoreit* Google Research usz@google.com

Llion Jones* Google Research llion@google.com

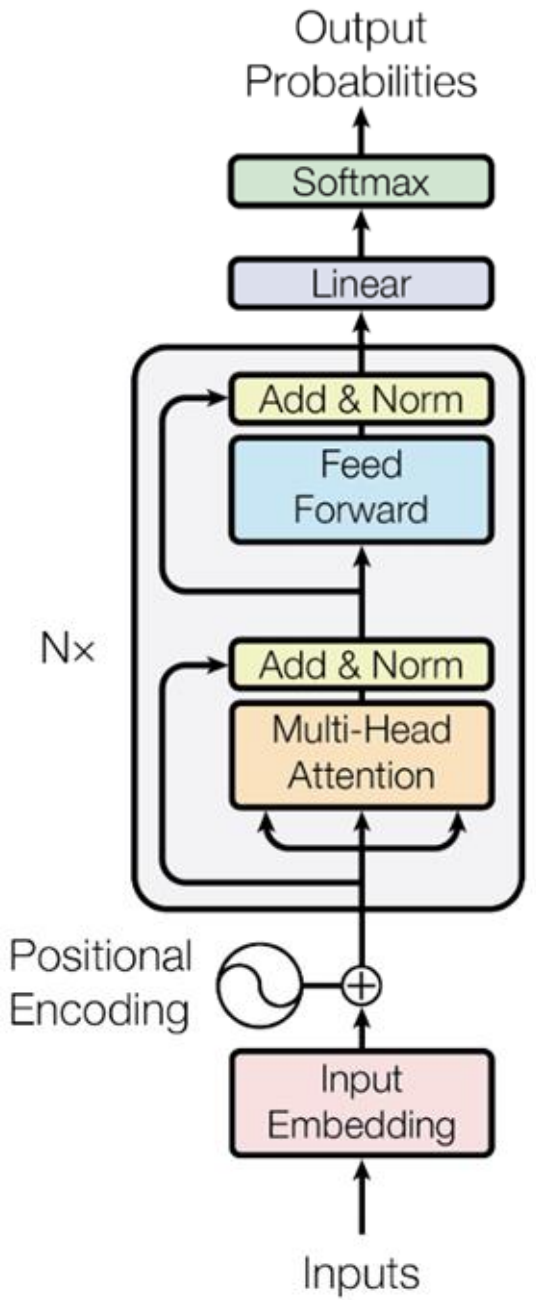
Aidan N. Gomez* † University of Toronto aidan@cs.toronto.edu

Łukasz Kaiser* Google Brain lukaszkaiser@google.com

Illia Polosukhin* ‡ illia.polosukhin@gmail.com

Abstract

The dominant sequence transduction models are based on complex recurrent or convolutional neural networks that include an encoder and a decoder. The best performing models also connect the encoder and decoder through an attention mechanism. We propose a new simple network architecture, the Transformer, based solely on attention mechanisms, dispensing with recurrence and convolutions entirely. Experiments on two machine translation tasks show these models to be superior in quality while being more parallelizable and requiring significantly less time to train. Our model achieves 28.4 BLEU on the WMT 2014 English-to-German translation task, improving over the existing best results, including ensembles, by over 2 BLEU. On the WMT 2014 English-to-French translation task, our model establishes a new single-model state-of-the-art BLEU score of 41.8 after training for 3.5 days on eight GPUs, a small fraction of the training costs of the best models from the literature. We show that the Transformer generalizes well to other tasks by applying it successfully to English constituency parsing both with large and limited training data.

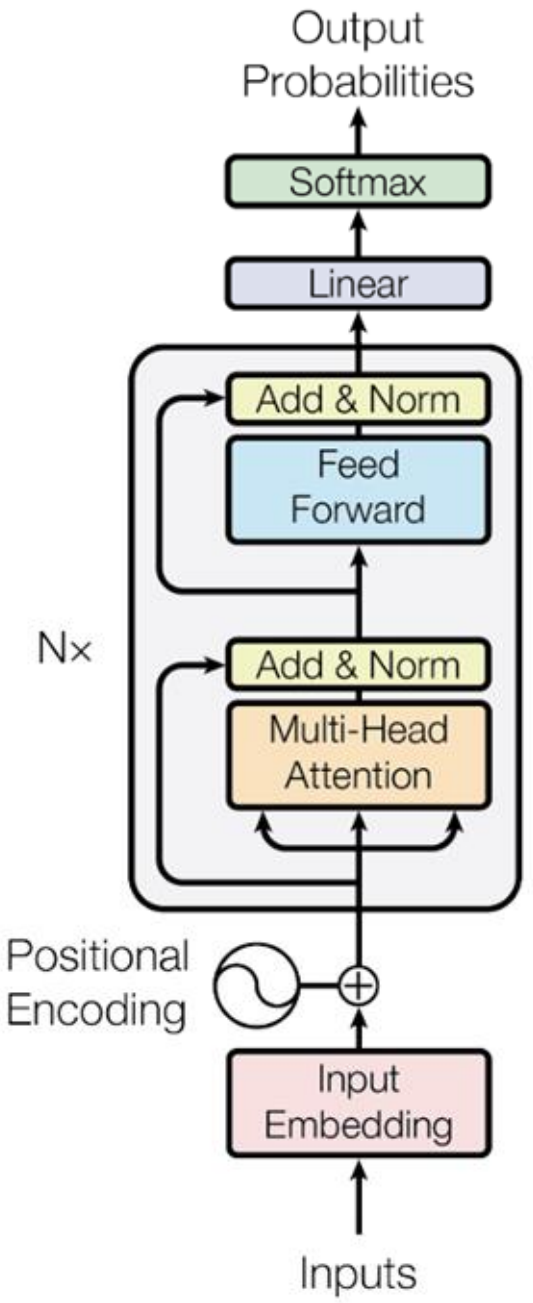


Attention Overview

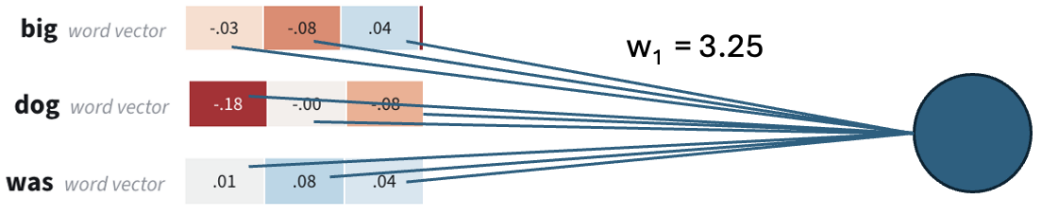
2 Big ideas to keep in mind:

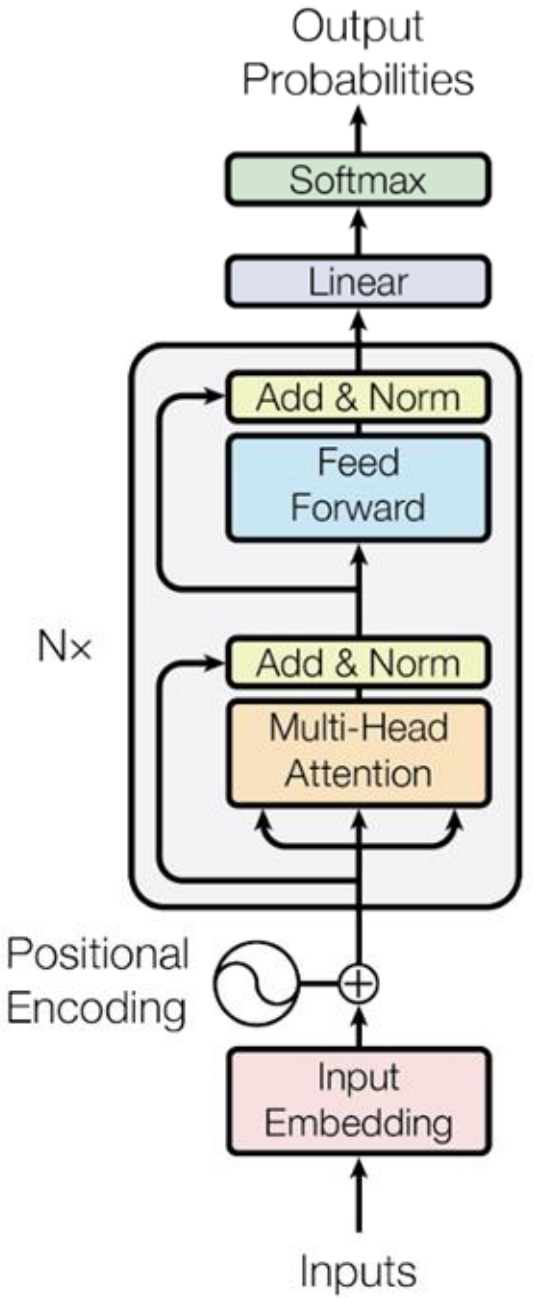
- A particular pattern of connections for neurons with weight sharing
  - *Similar to CNNs!*
- Weights that are computed from inputs
  - *We'll get to what this means!*

+ a few more details

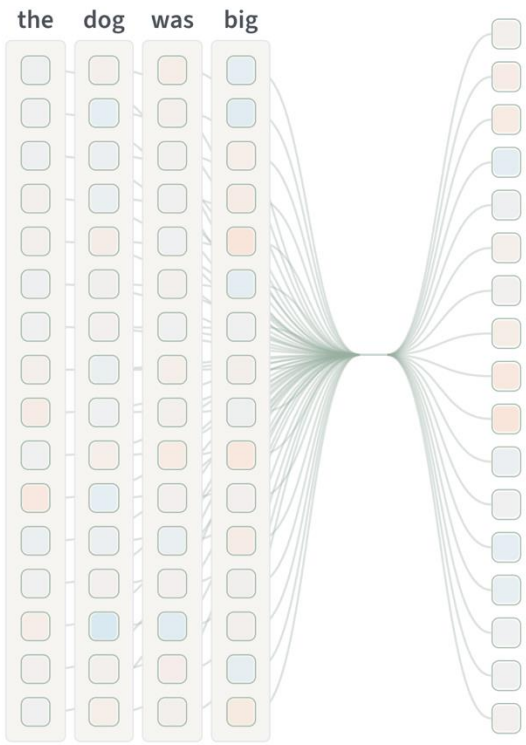


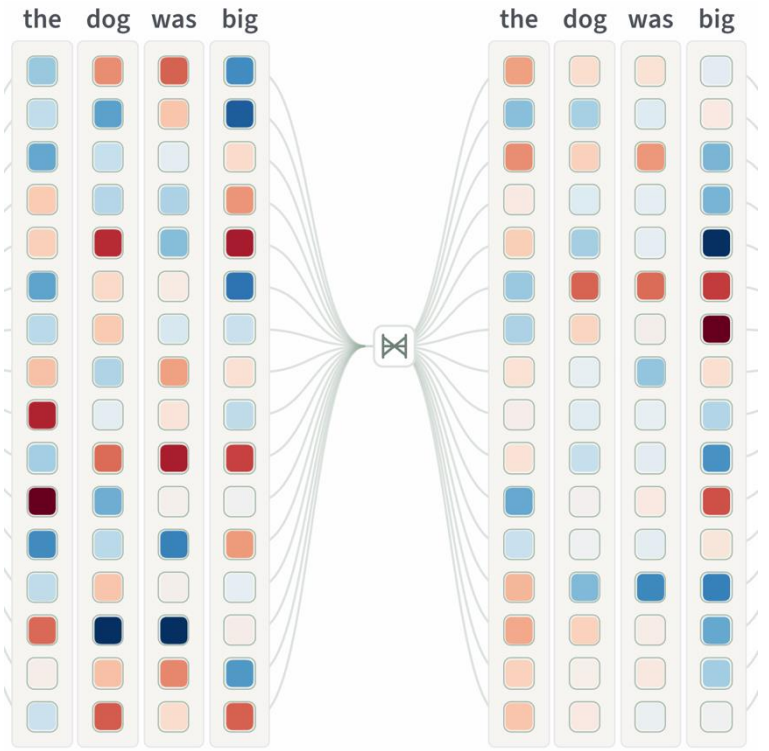
Motivation 1: Variable length inputs

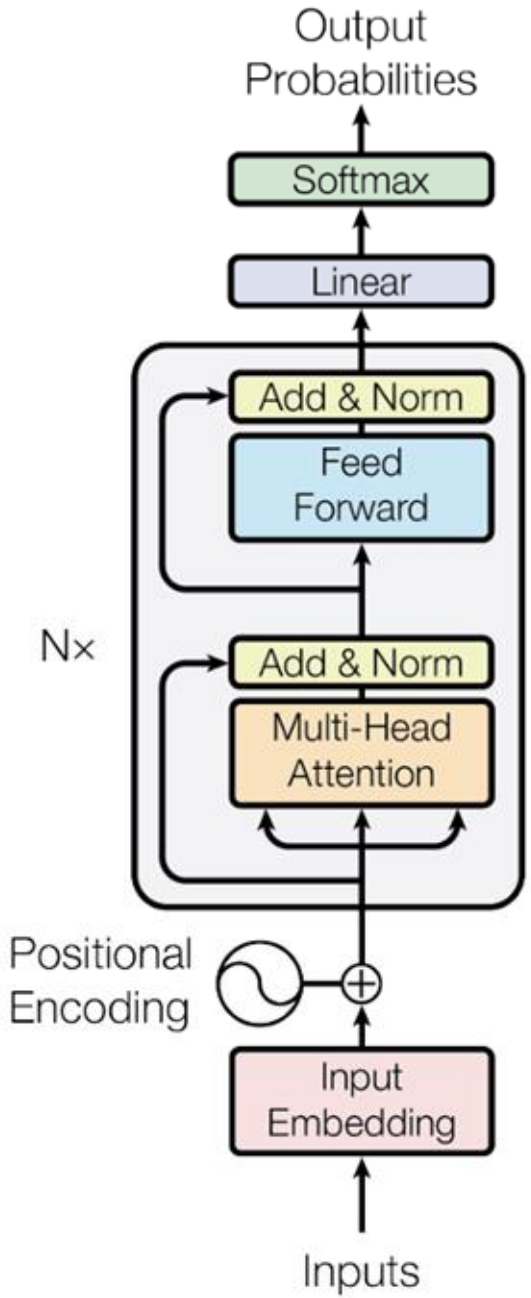




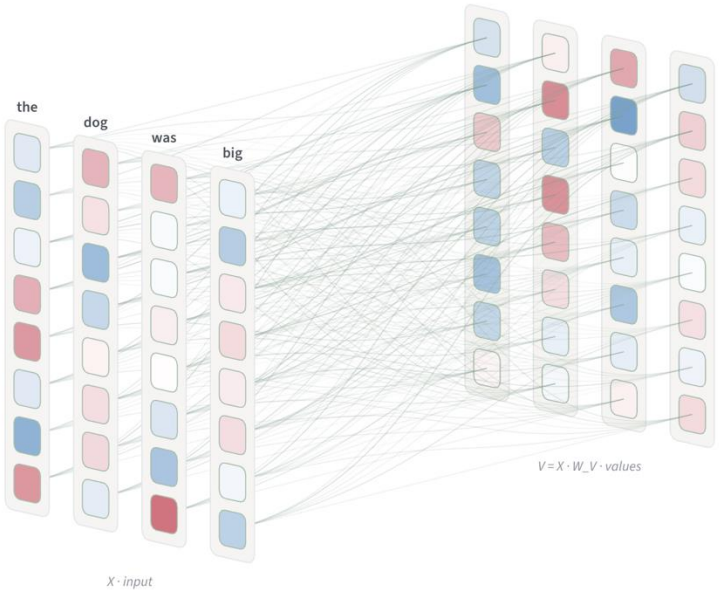
Organization by token

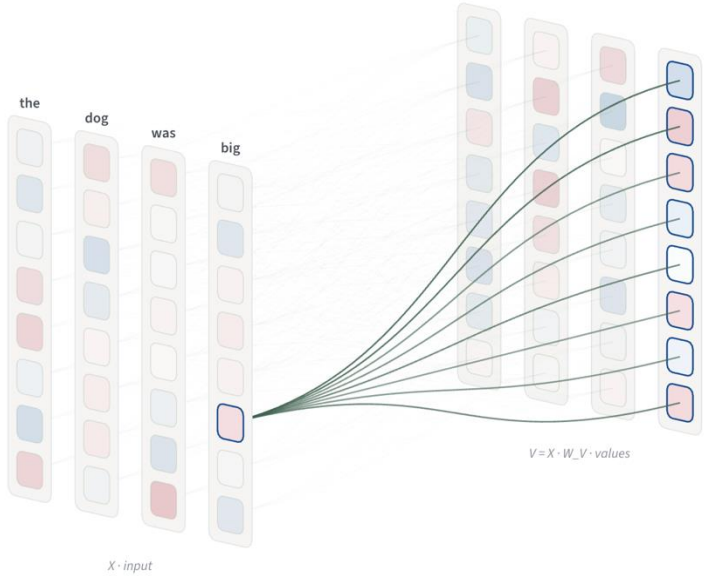


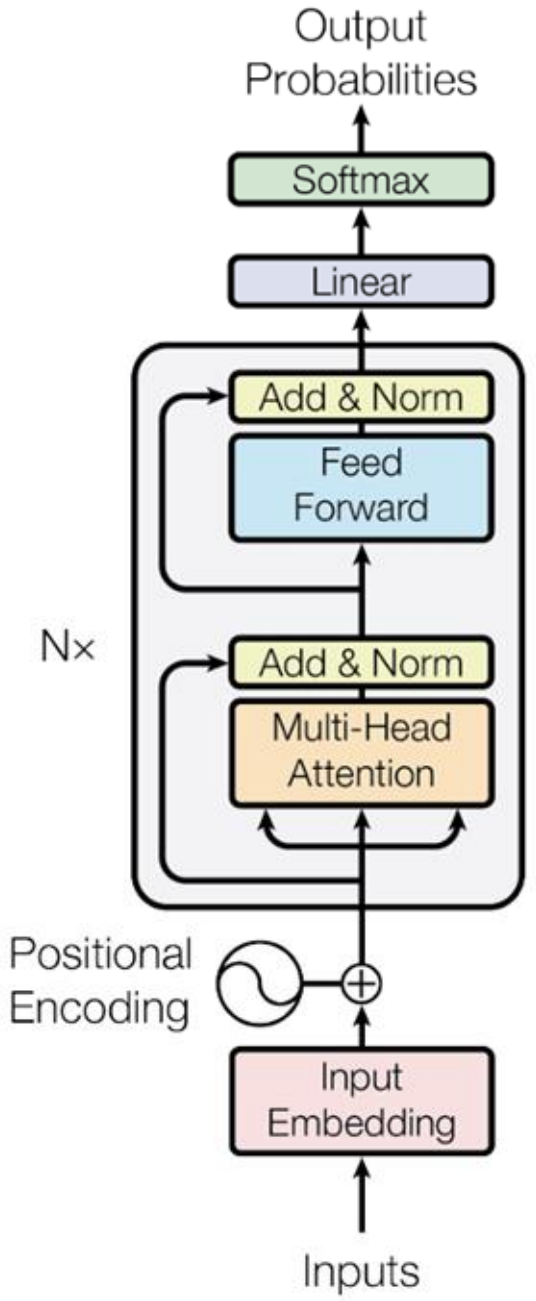




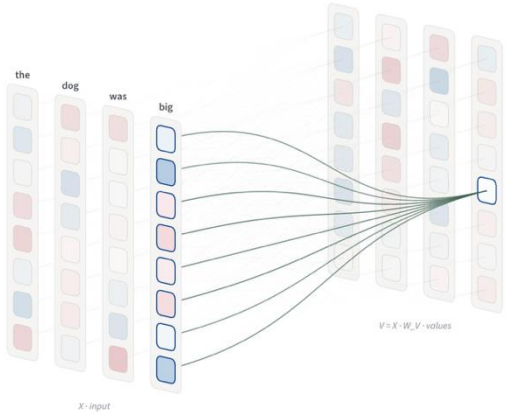
Organization by token

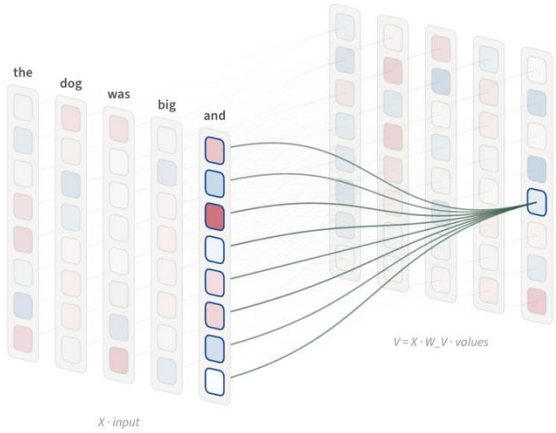


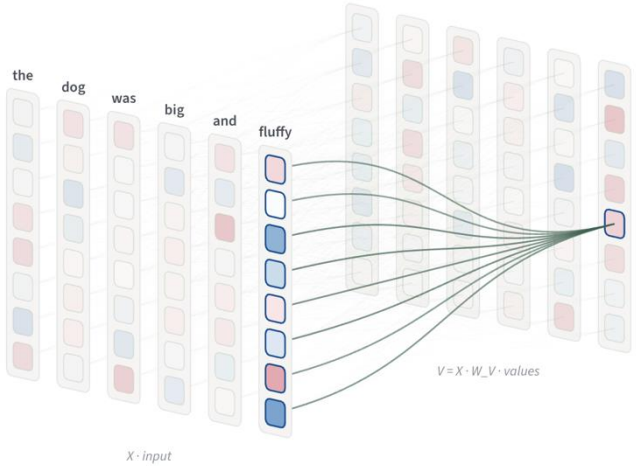


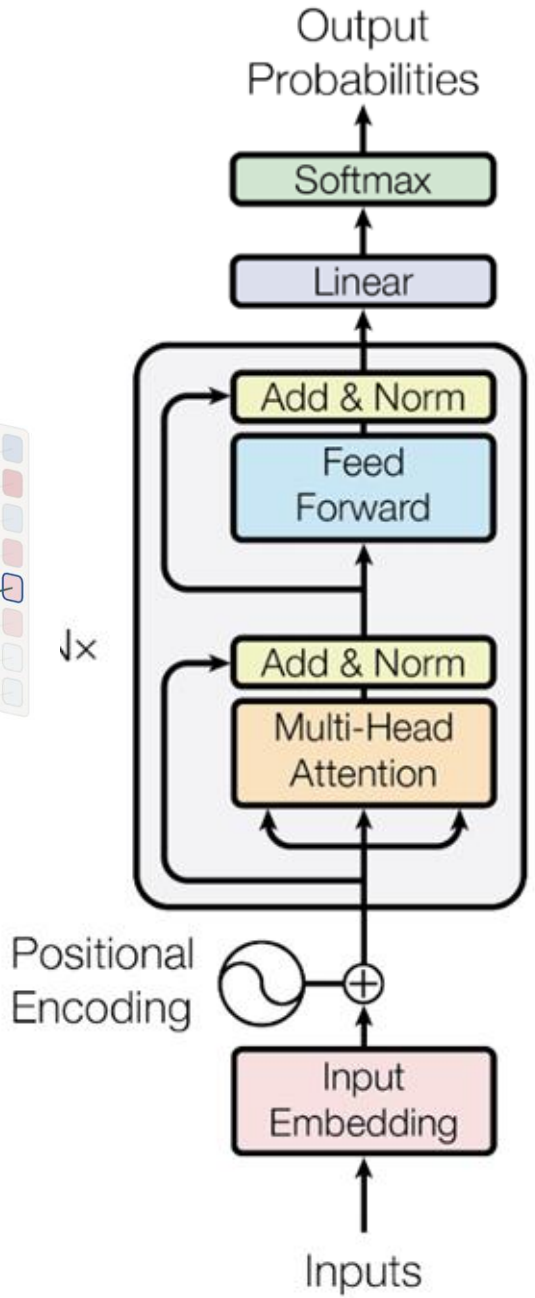


Organization by token

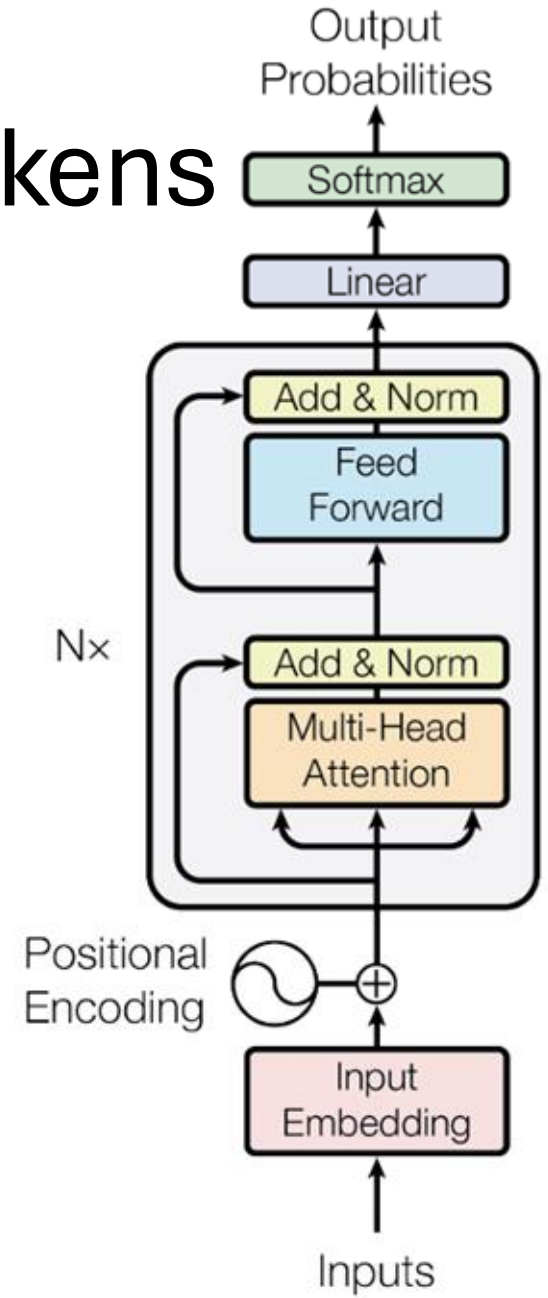




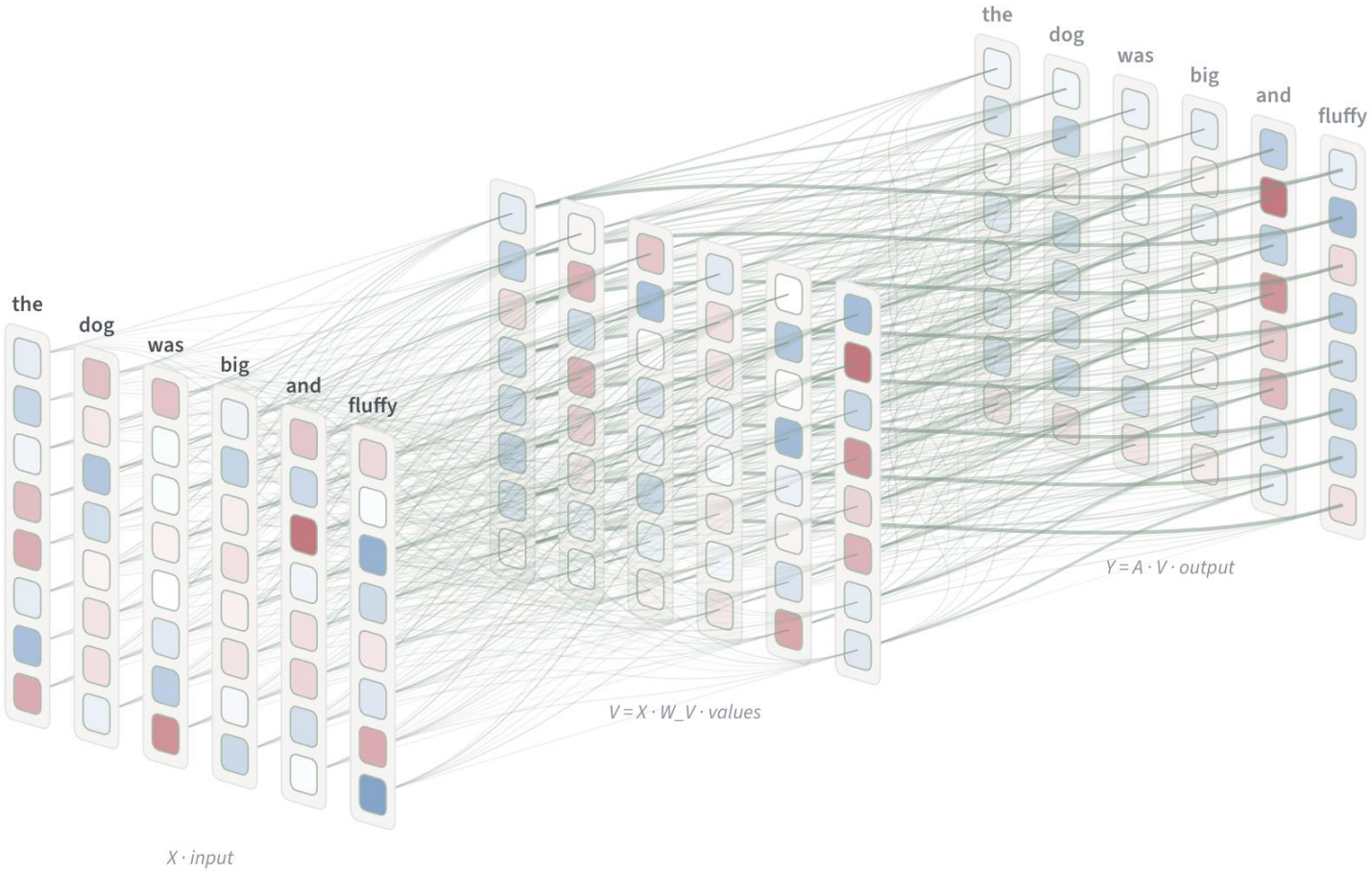


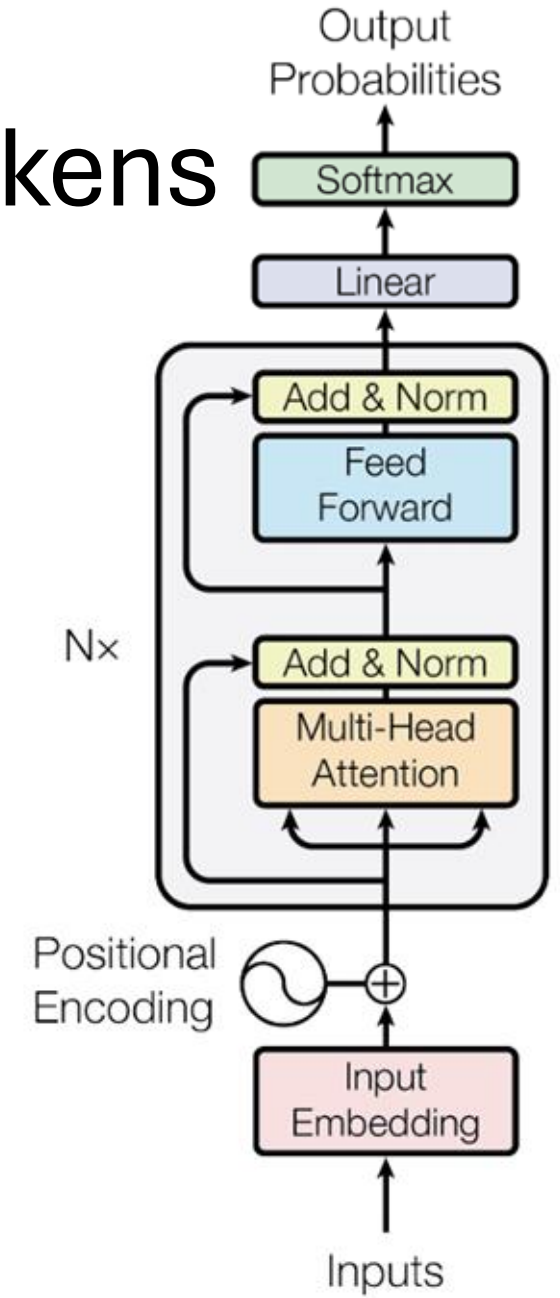


Attention: connect neurons across tokens

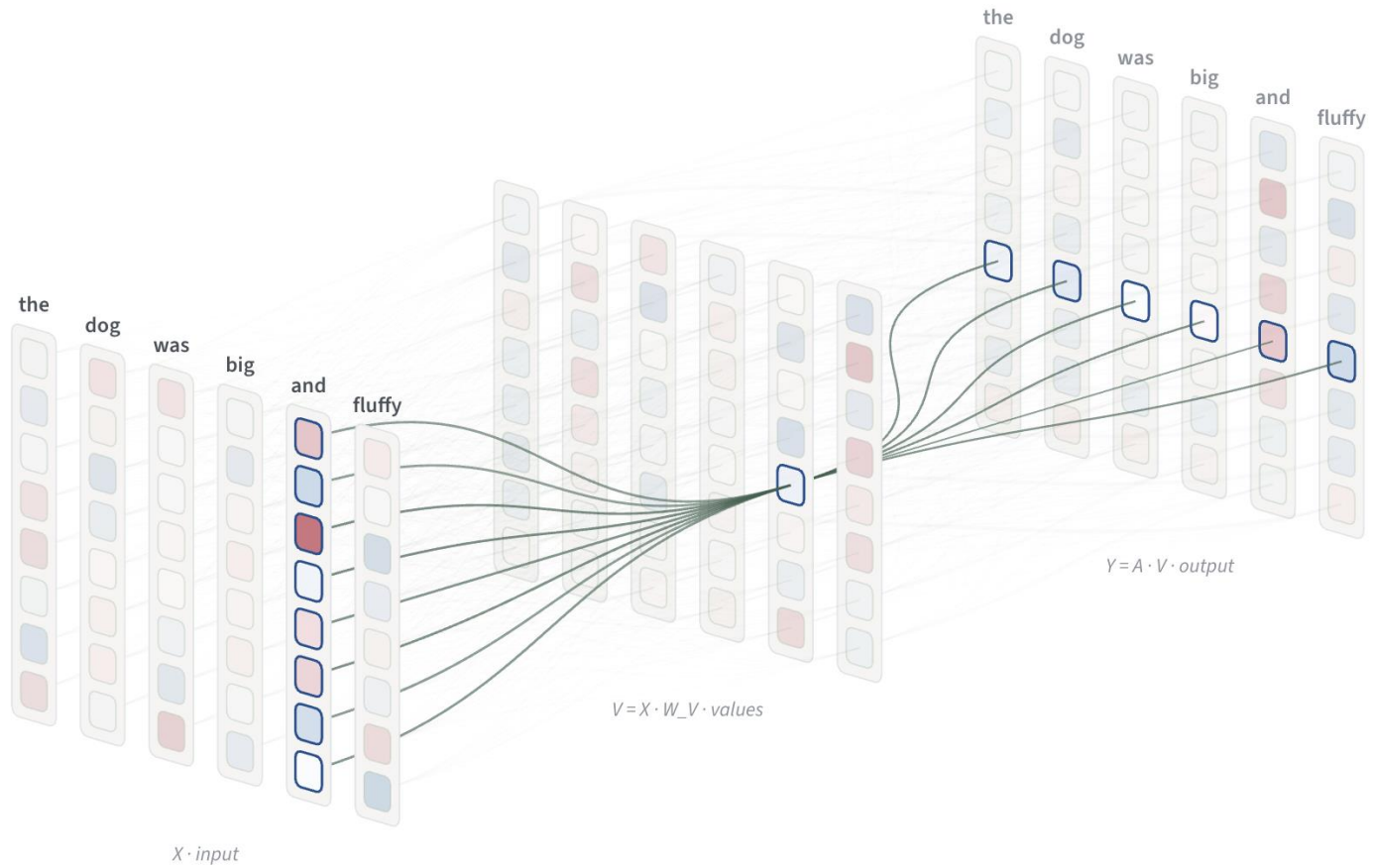


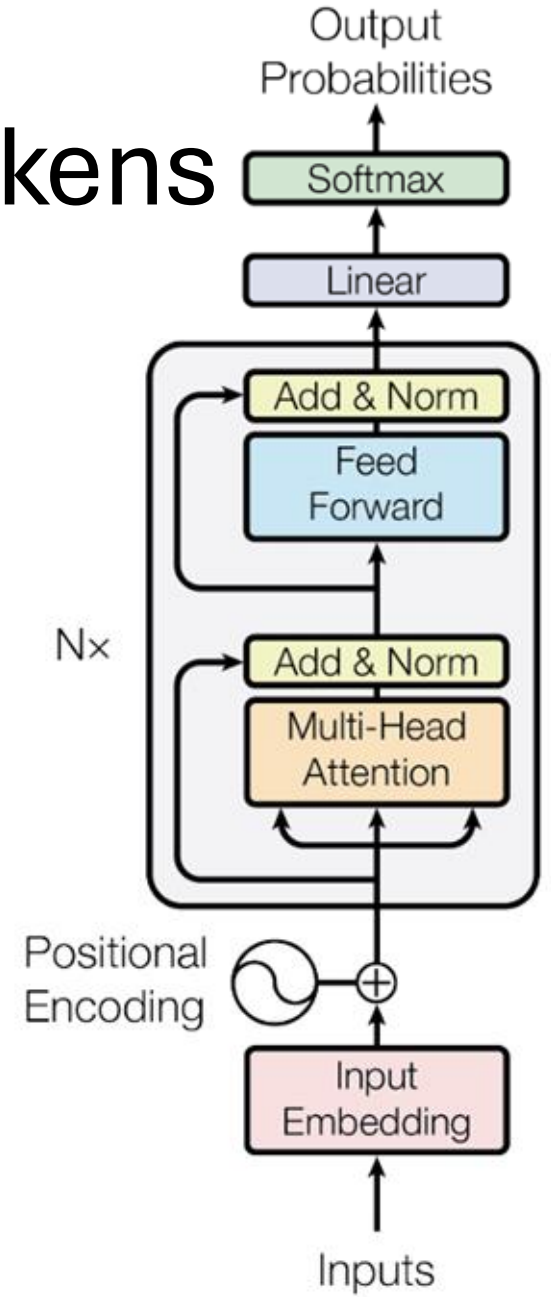
Attention: add connections across tokens



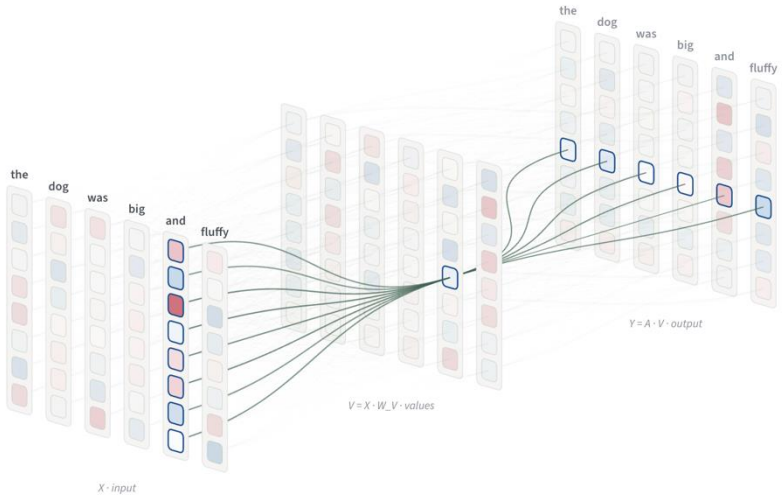


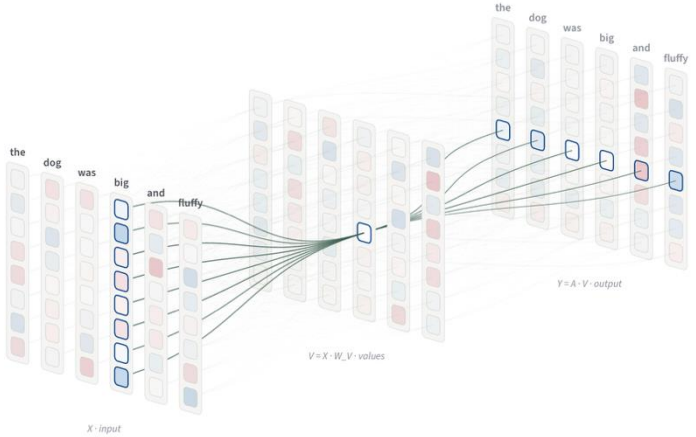
Attention: connect neurons across tokens

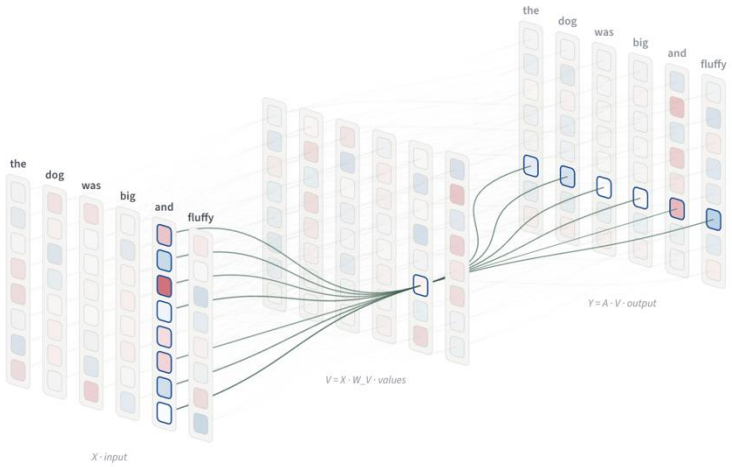


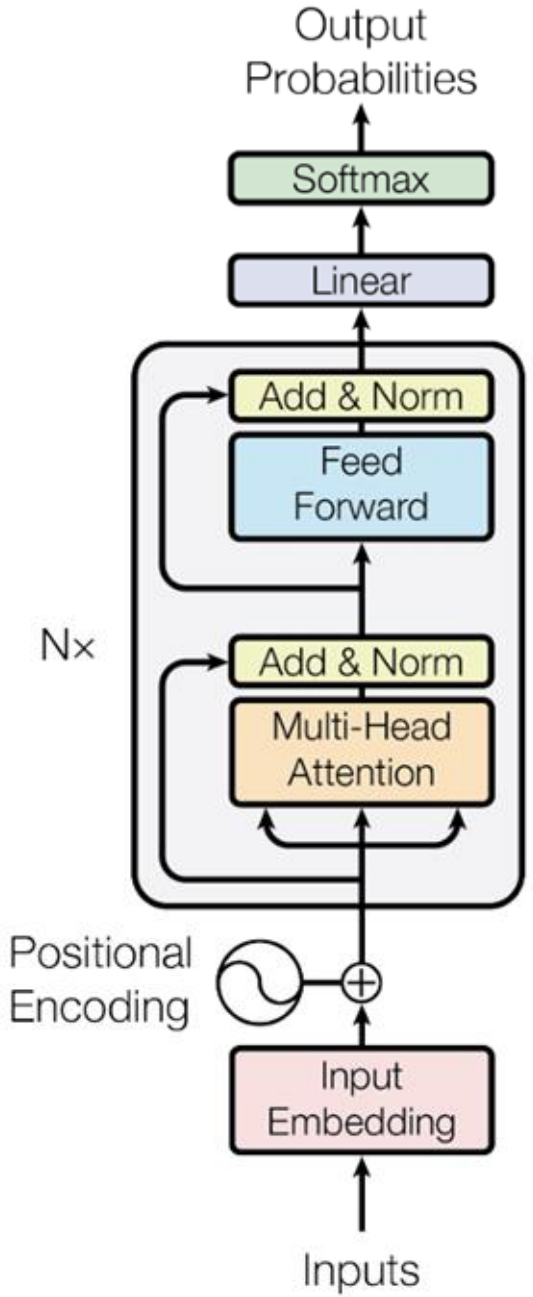


Attention: weight sharing

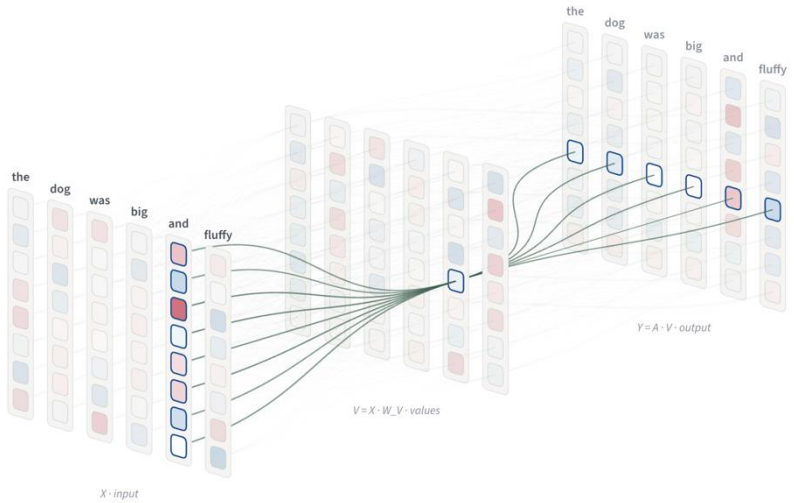


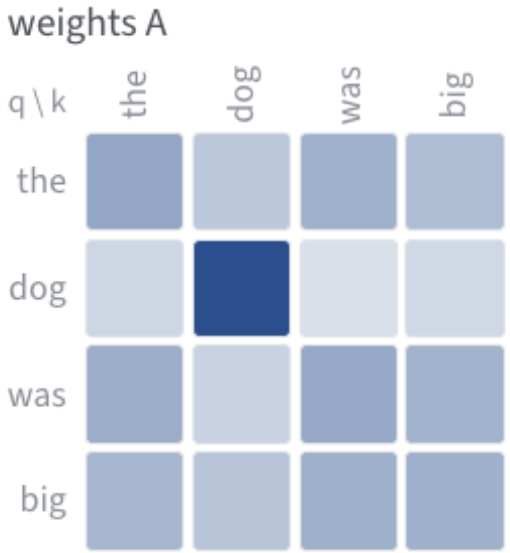


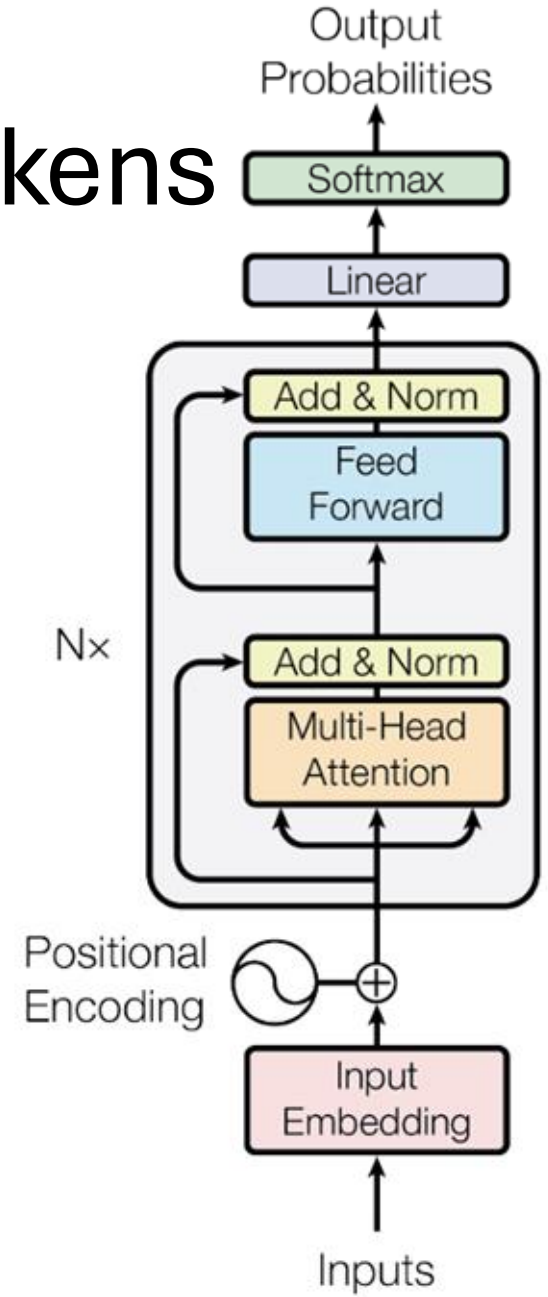




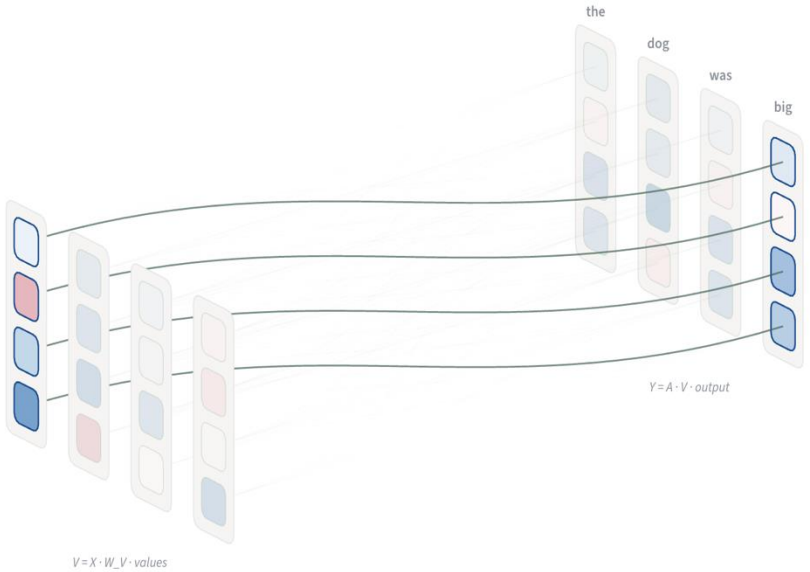
Attention: connect neurons across tokens

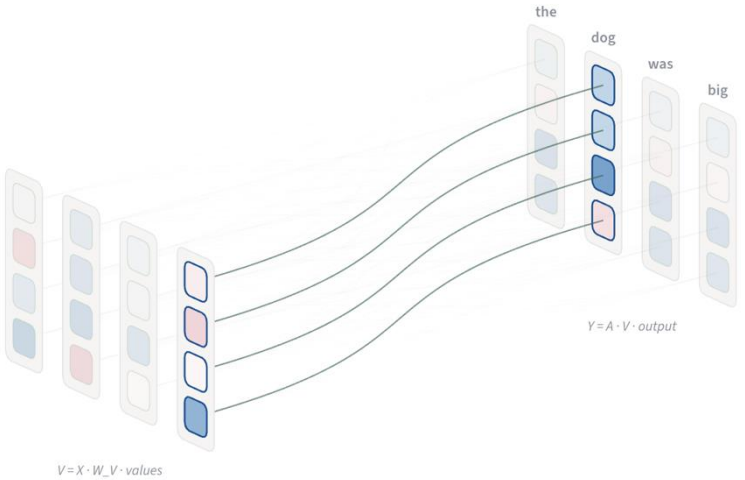


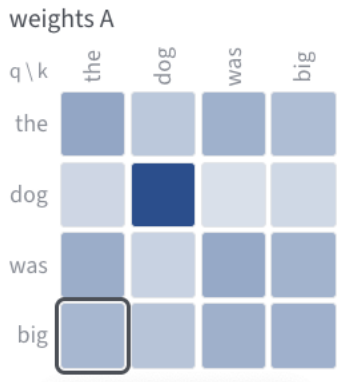


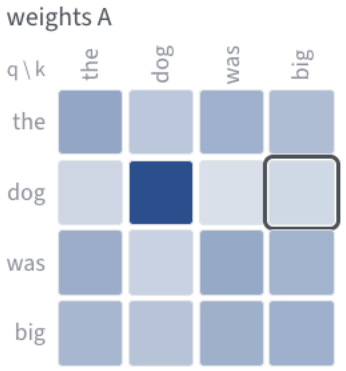


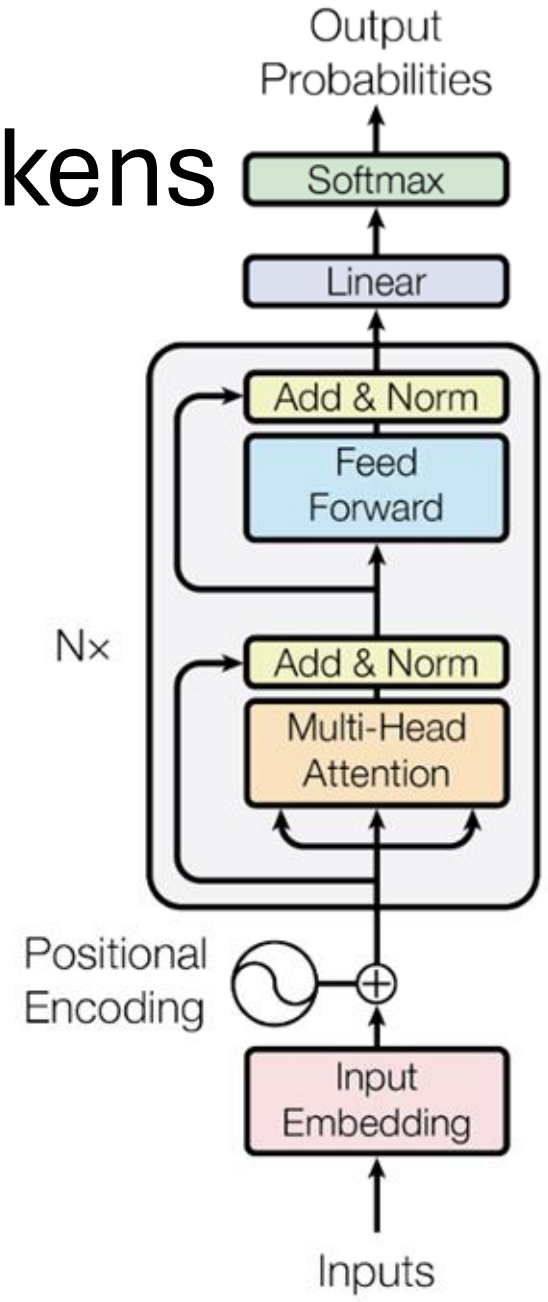
Attention: connect neurons across tokens



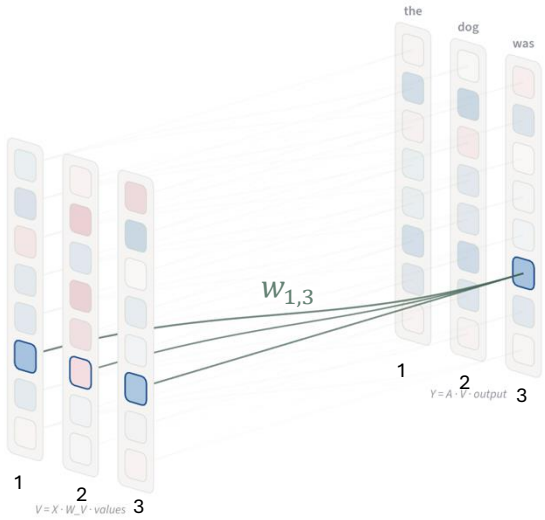


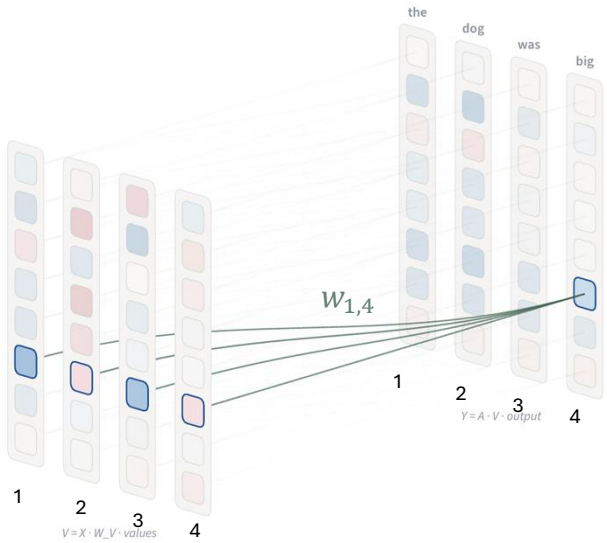


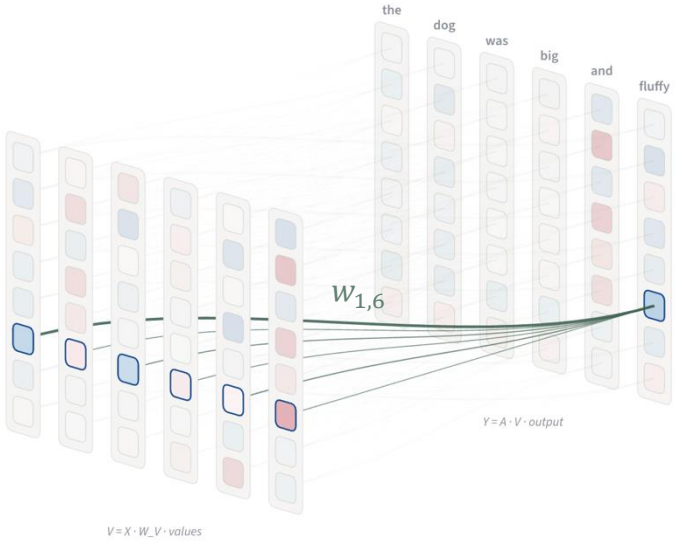




Handling variable-length inputs



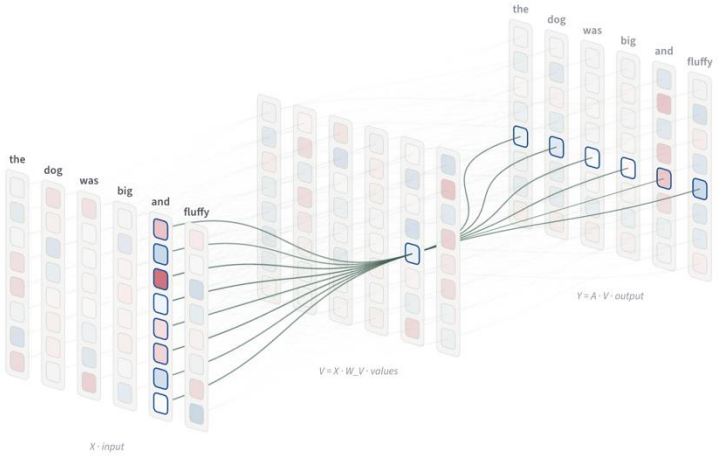


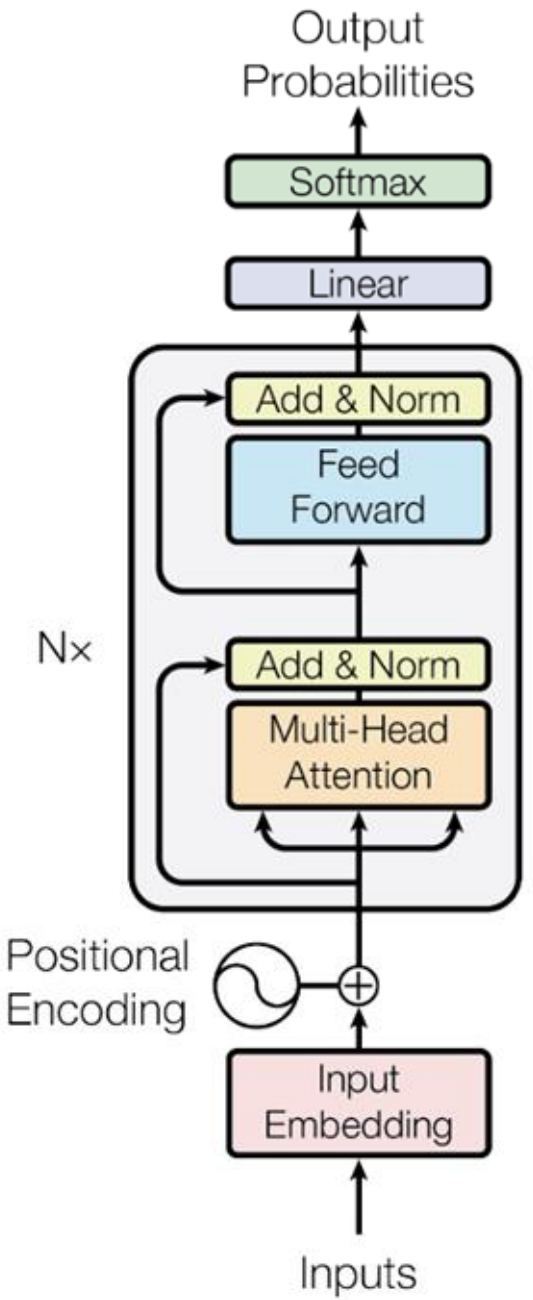


$w_{1,140}$

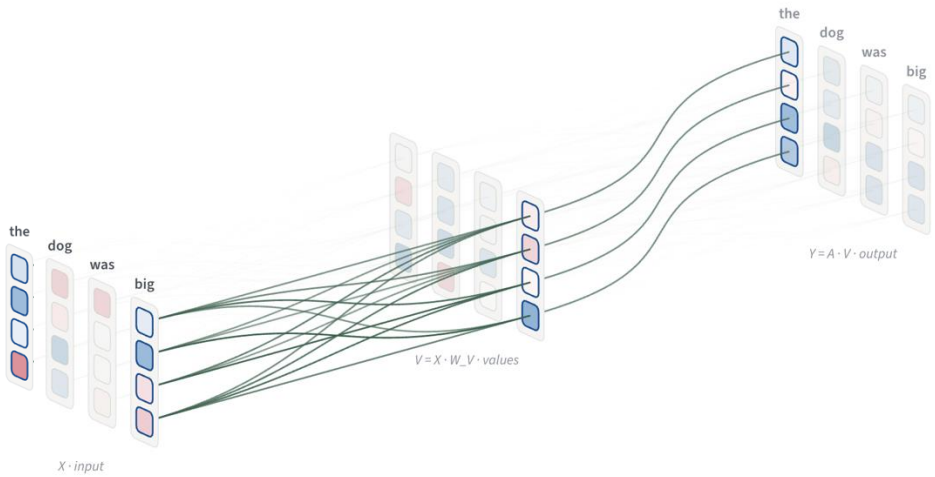
the dog was big and fluffy, and by the time Mara got him out of the rain, he weighed about as much as her washing machine. She spread a towel on the kitchen floor. He stepped neatly over it, climbed onto the couch, and let out the long, put-upon sigh of someone who'd been waiting all day for her to come home. Mara had never seen him before in her life.

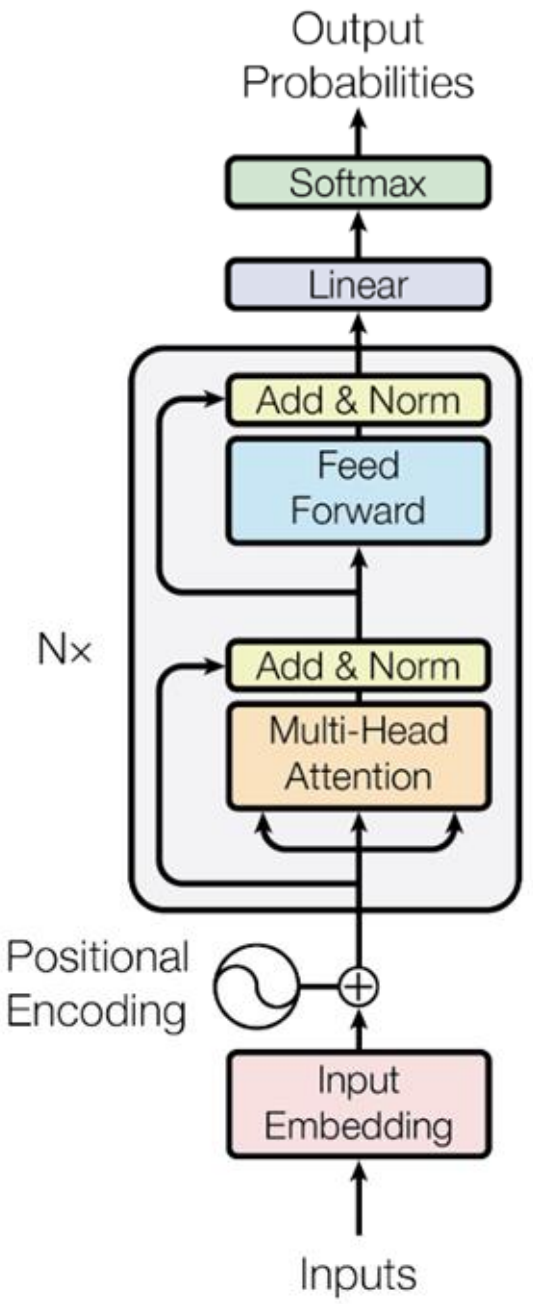
Computing weights from activations



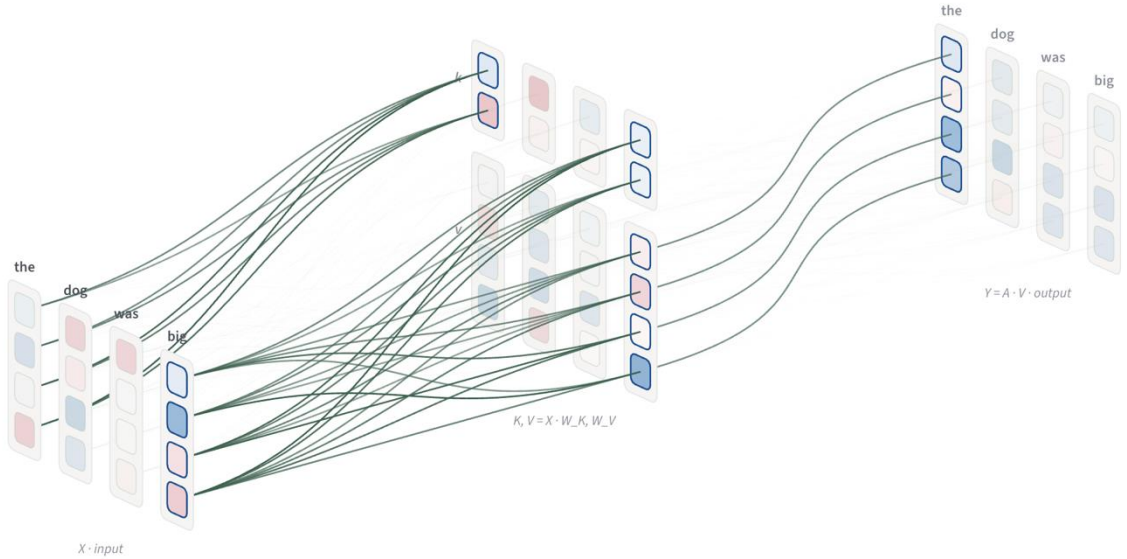


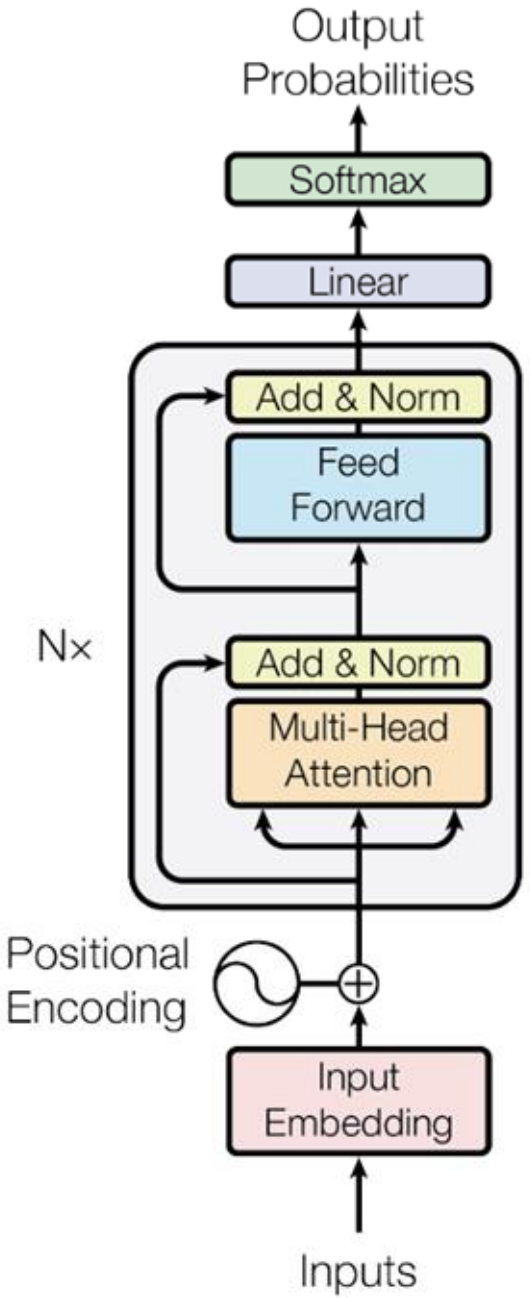
Dot product weights



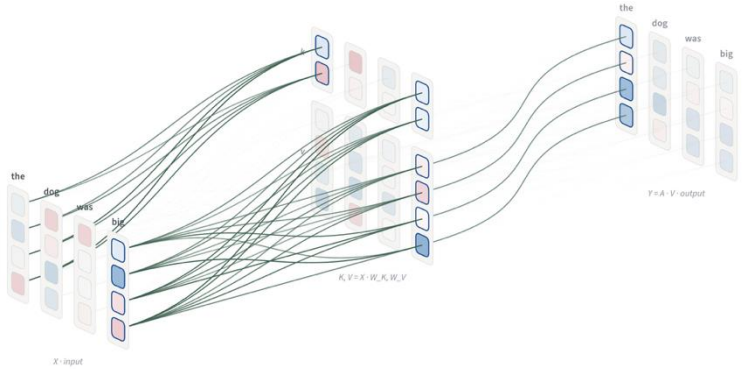


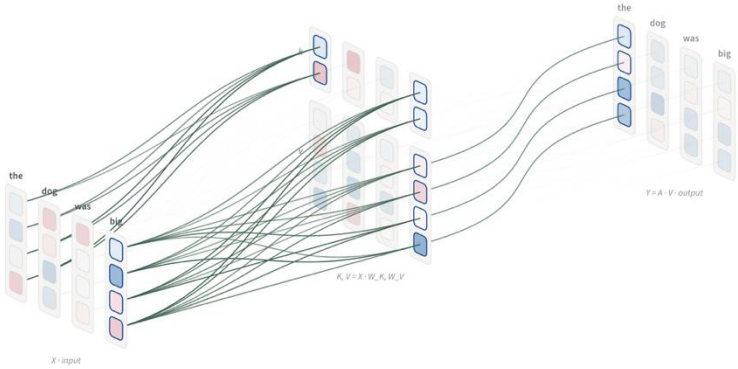
Learned dot product weights





Avoiding symmetry



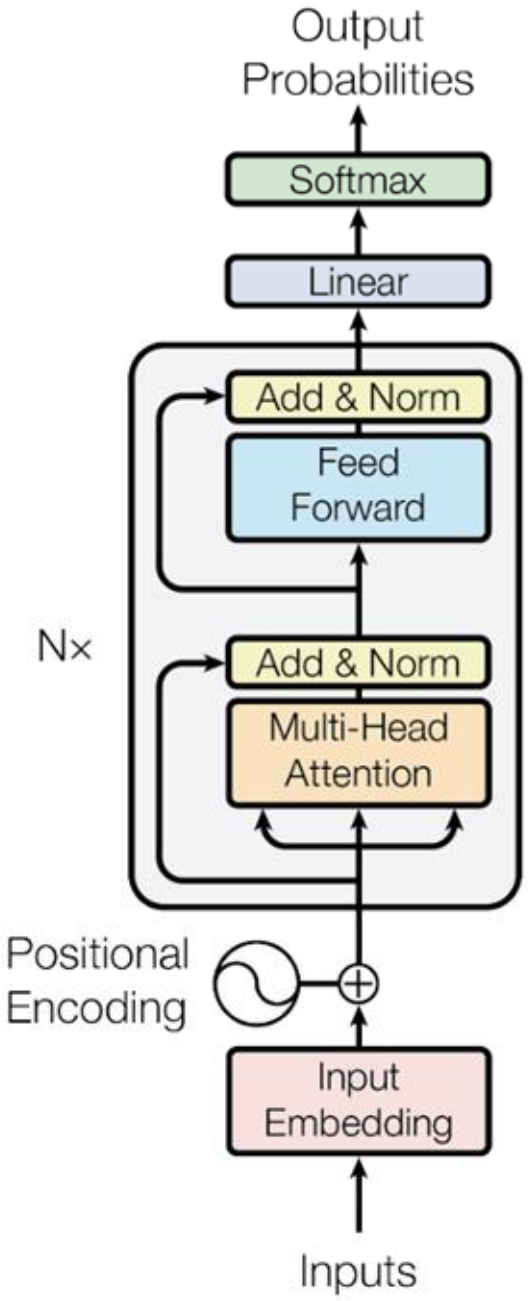


weights A

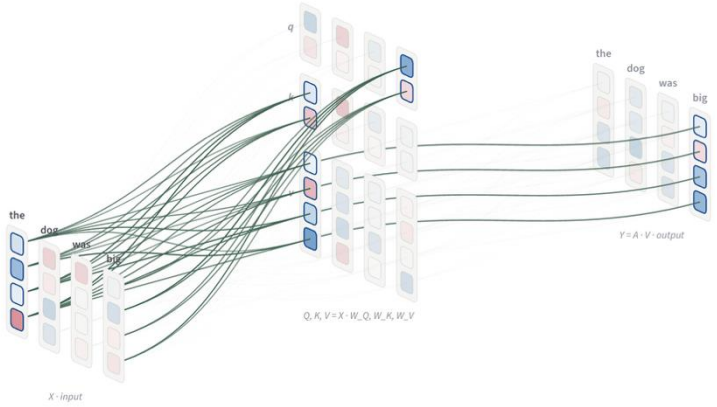
q \ k	the	dog	was	big
the				
dog				
was				
big				

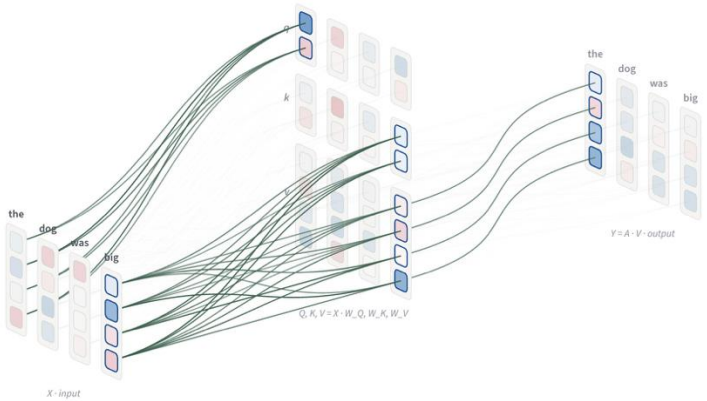
weights A

q \ k	the	dog	was	big
the				
dog				
was				
big				



Avoiding symmetry



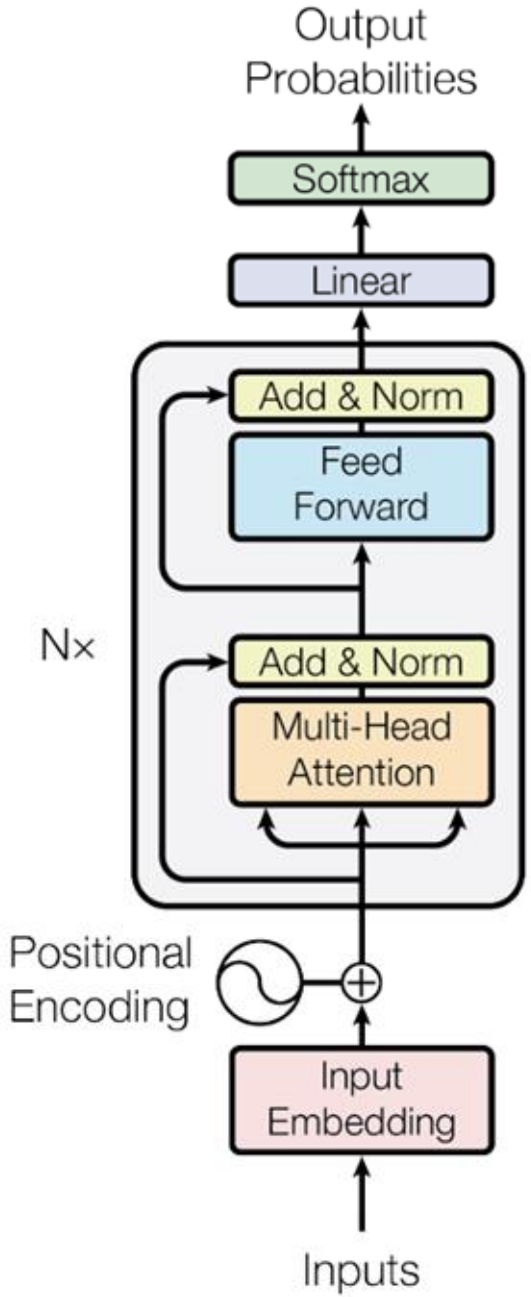


weights A

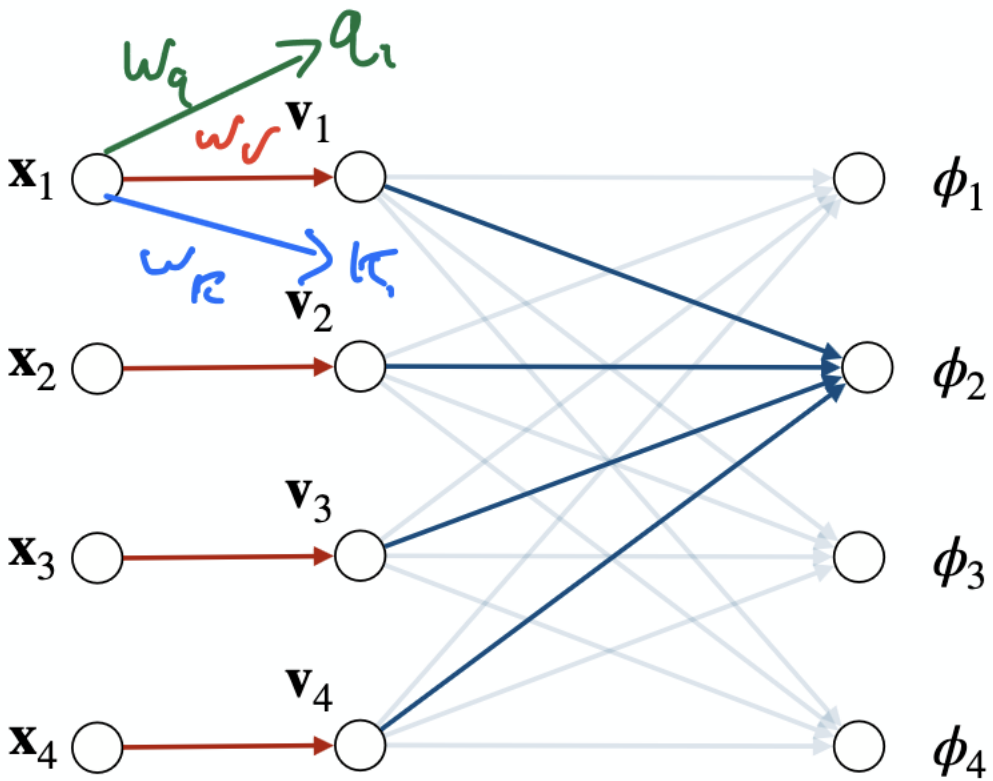
q \ k	the	dog	was	big
the				
dog				
was				
big				

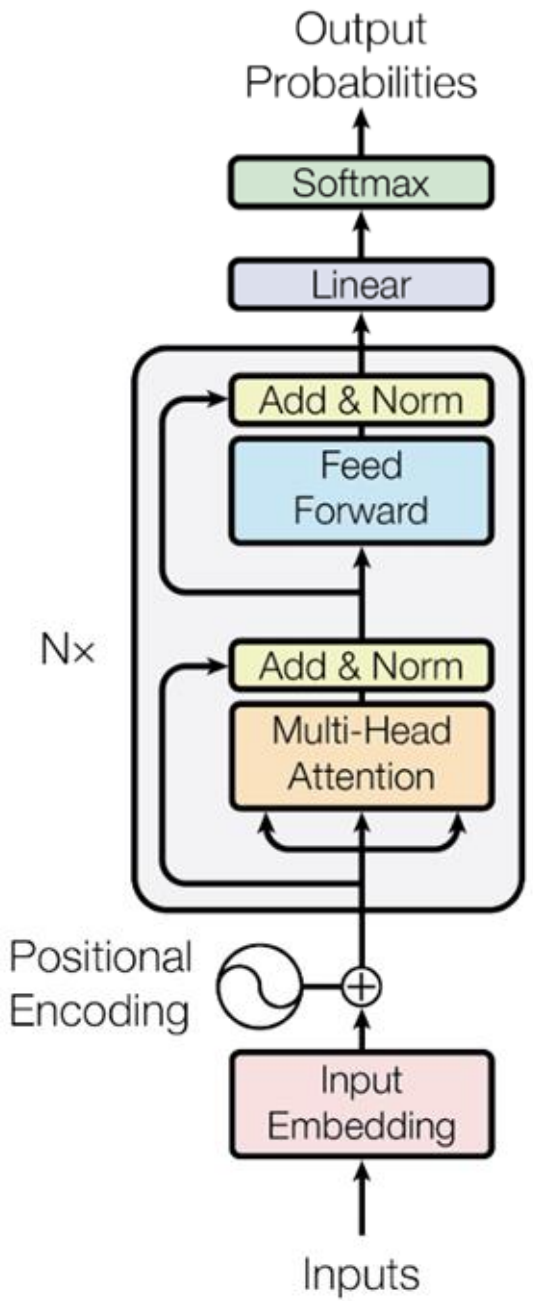
weights A

q \ k	the	dog	was	big
the				
dog				
was				
big				

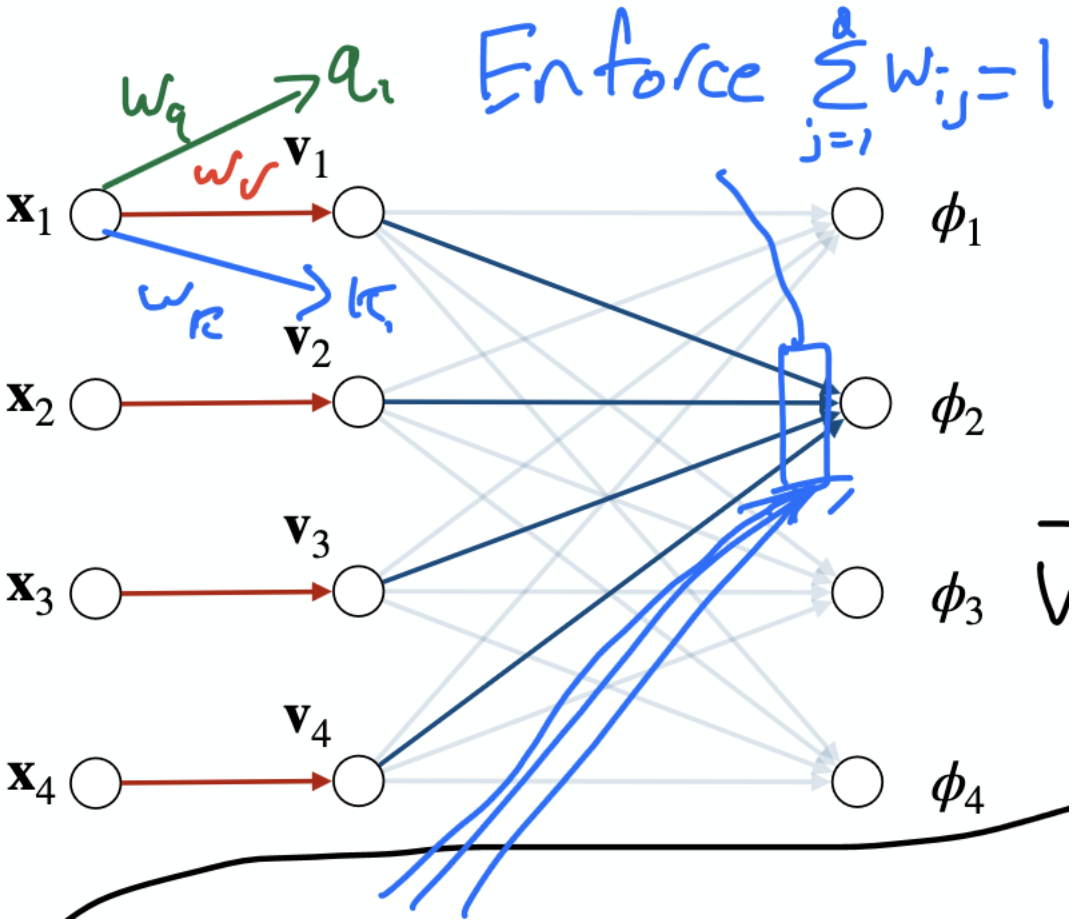


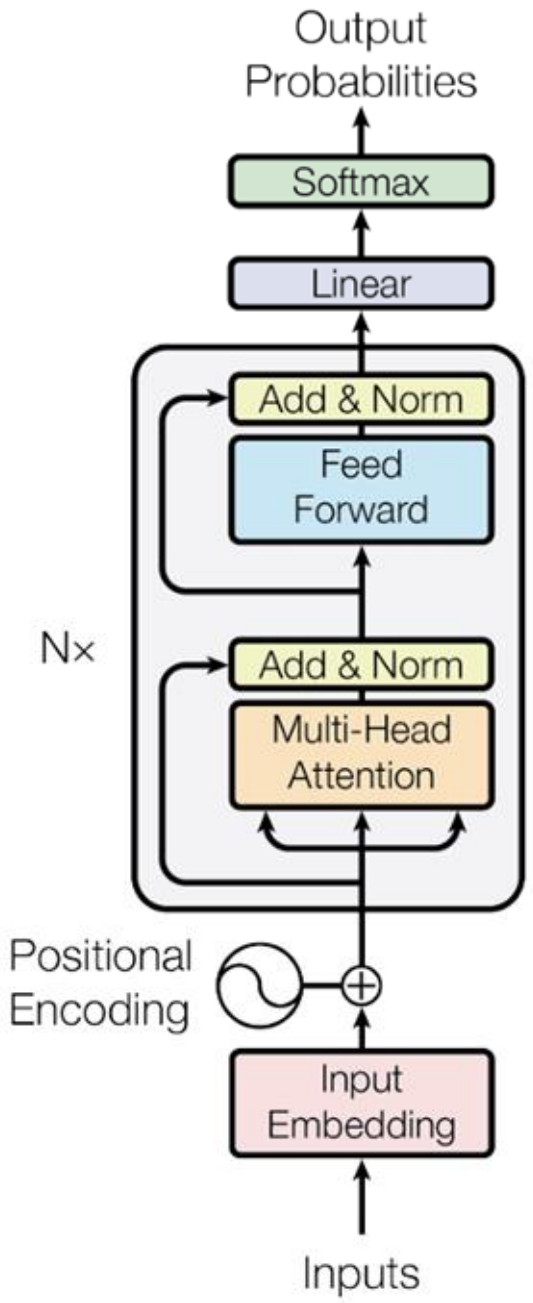
Avoiding scale issues



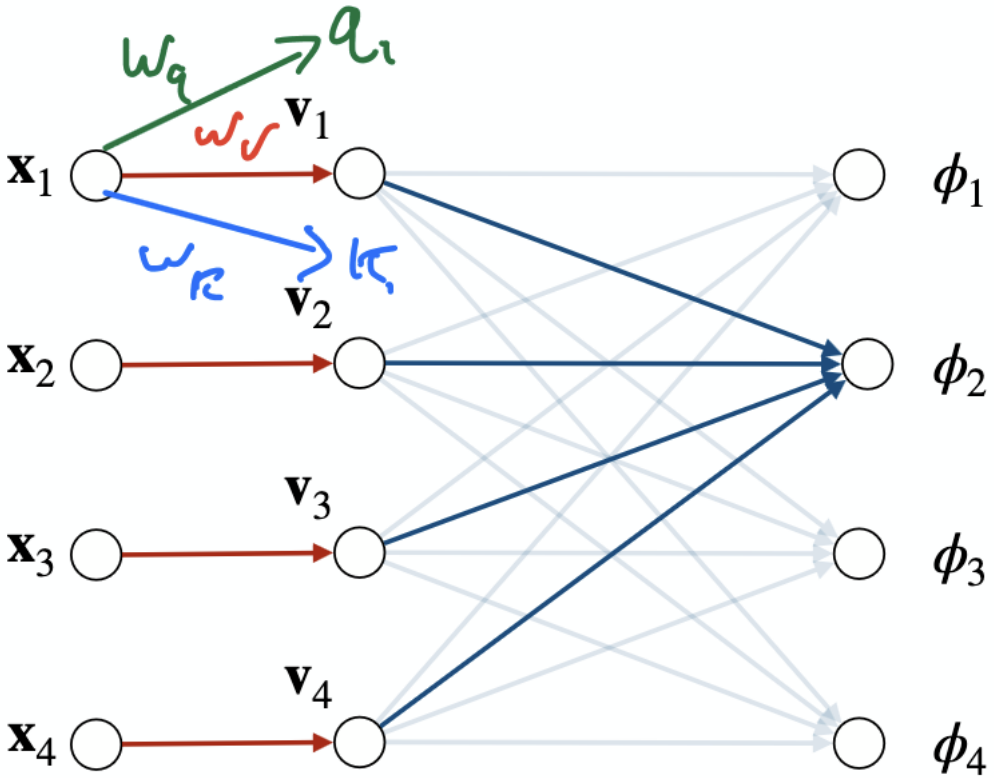


Avoiding scale issues

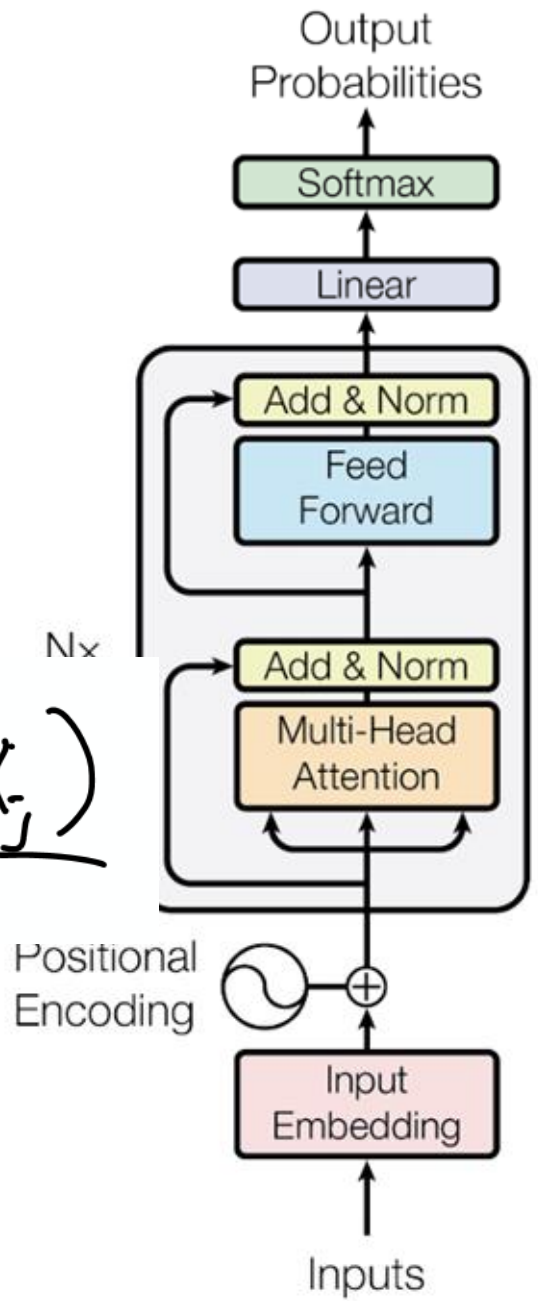




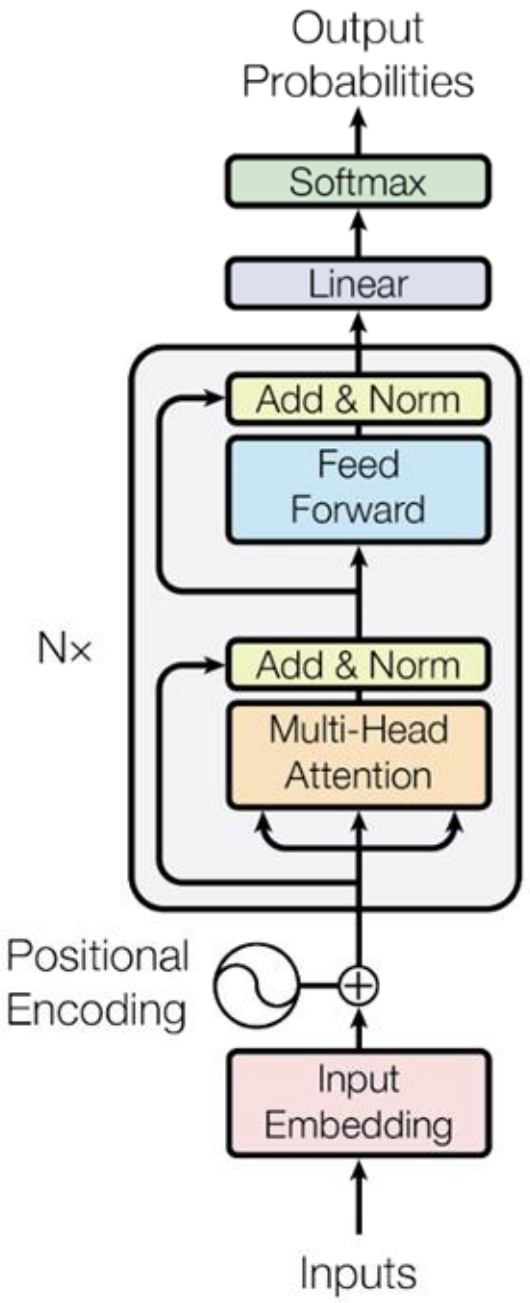
Avoiding scale issues



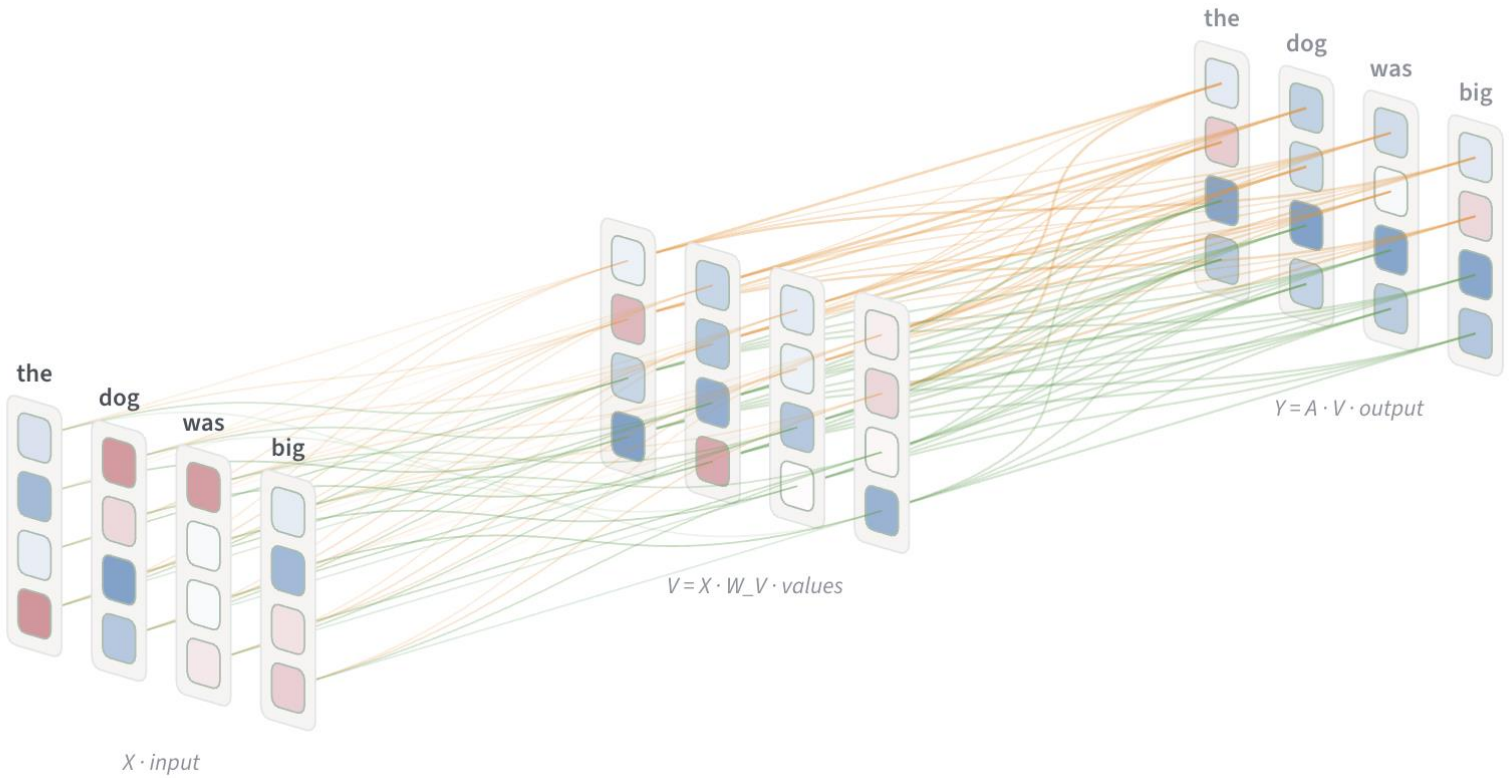
$$w_{ij} = \frac{(W_q x_i)^T (W_k x_j)}{\sqrt{d}}$$

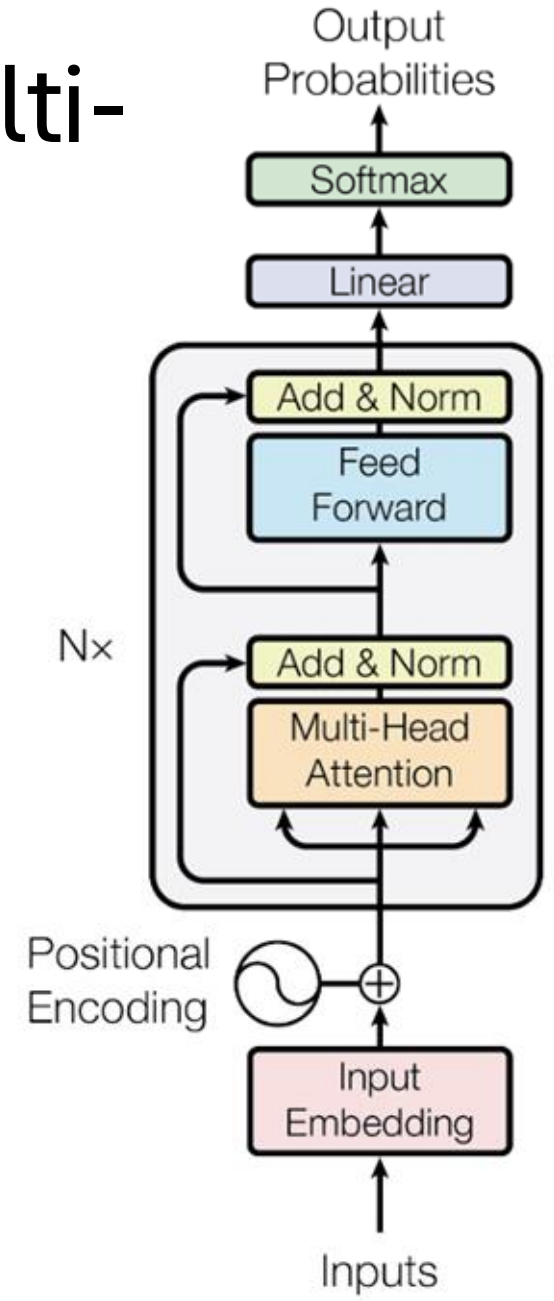


Visualizing connections

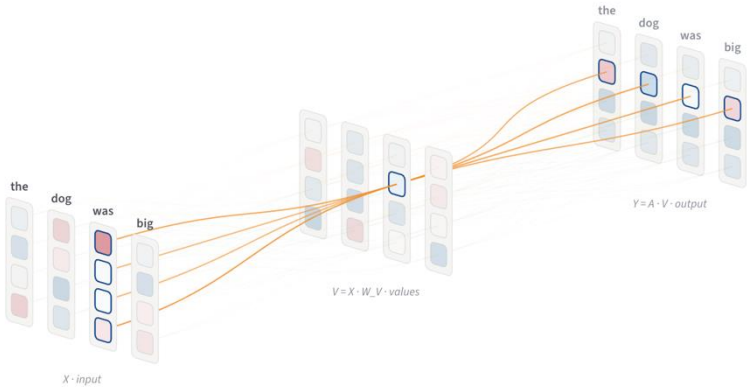


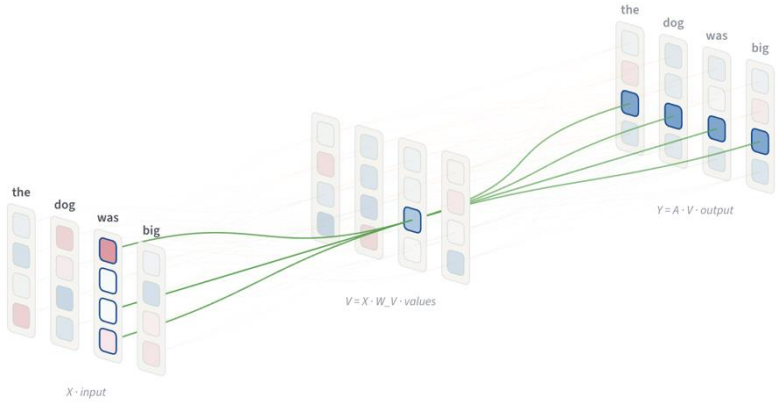
Multiple patterns of connections (multi-head)

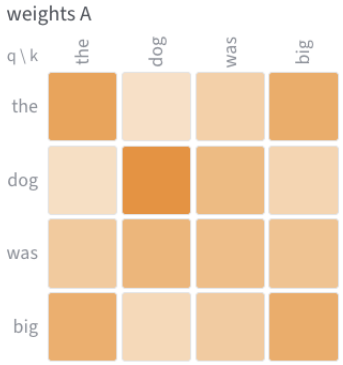


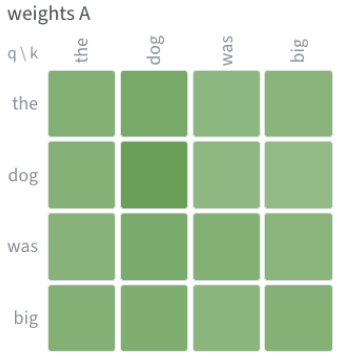


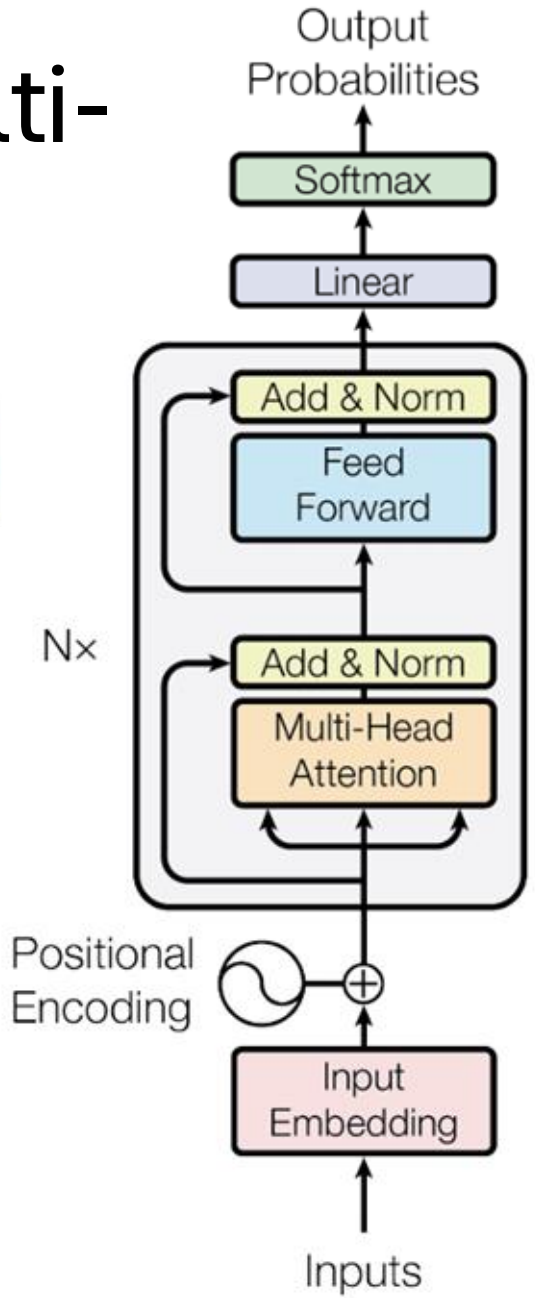
Multiple patterns of connections (multi-head)



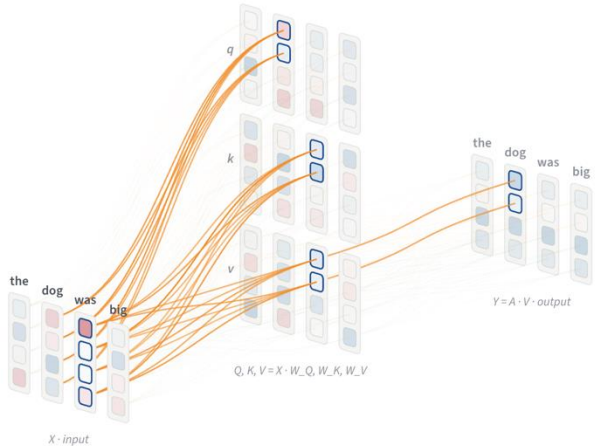


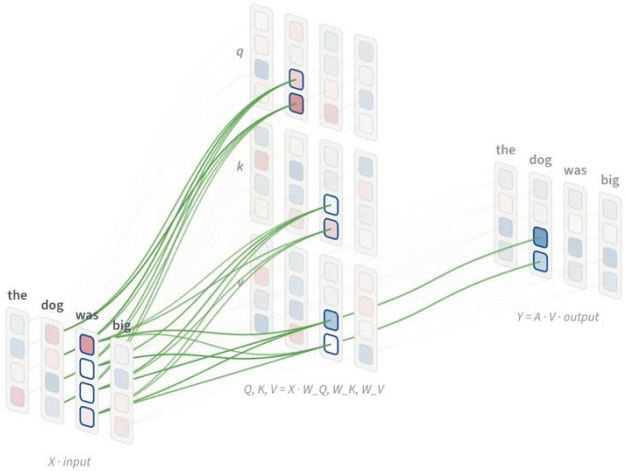


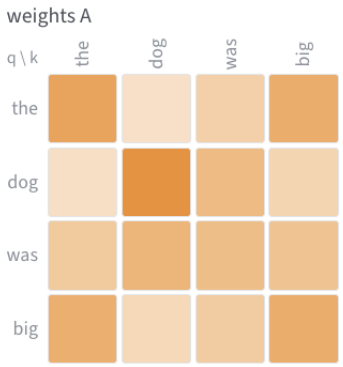


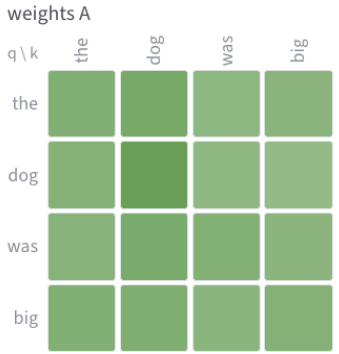


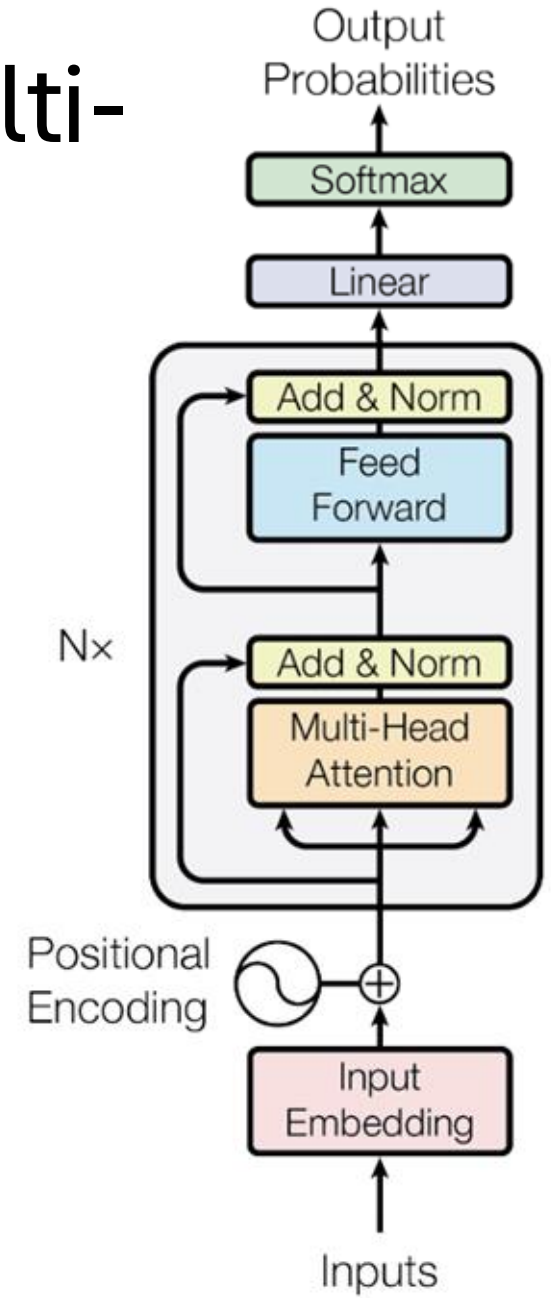
Multiple patterns of connections (multi-head)



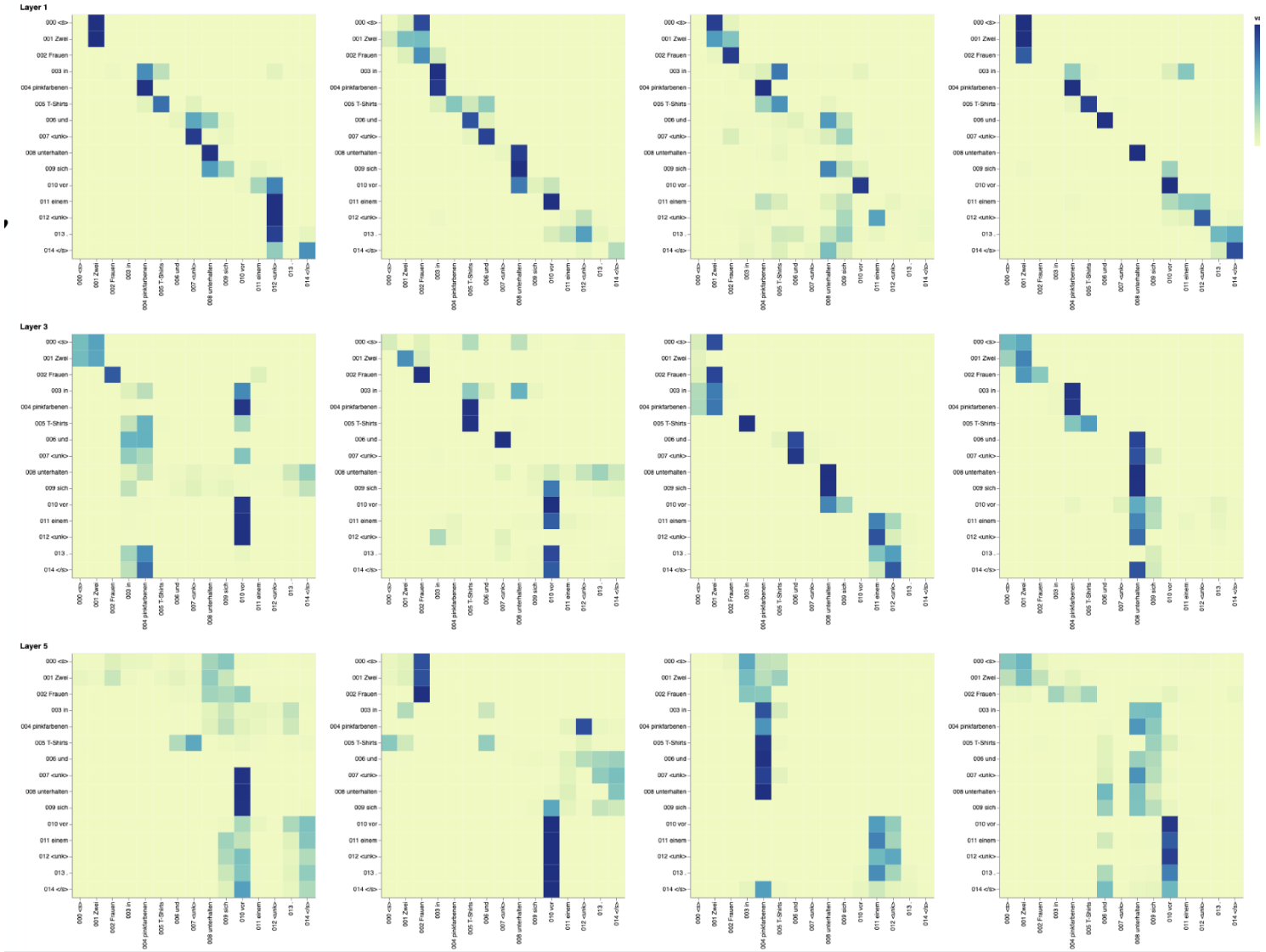


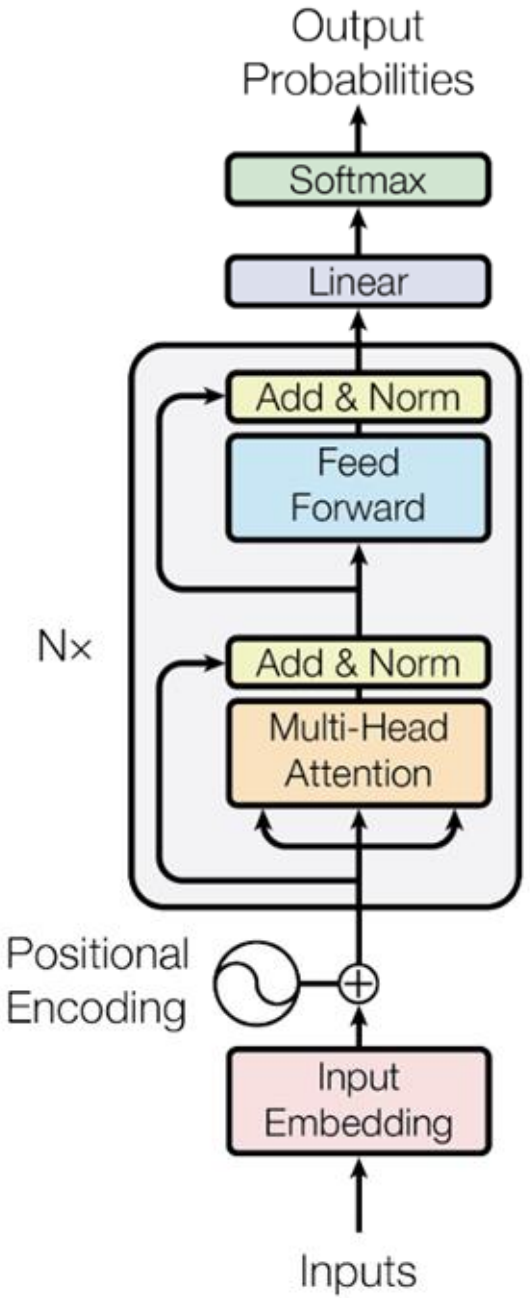




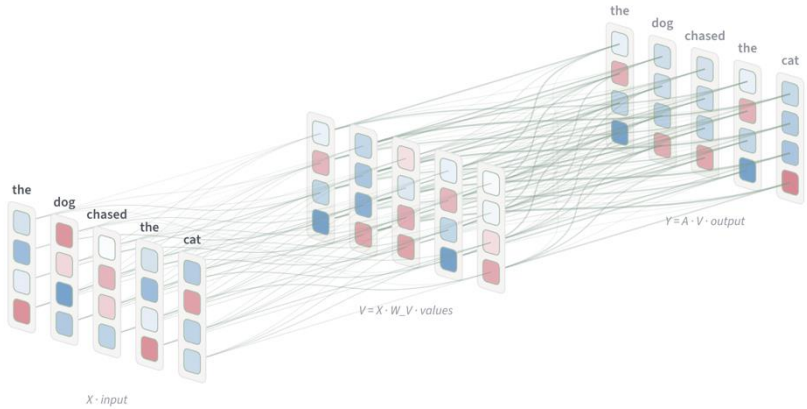


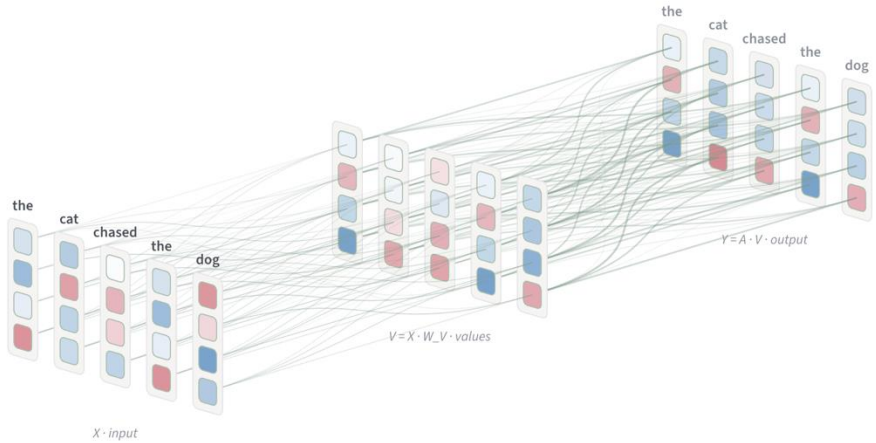
Visualizing multiple heads

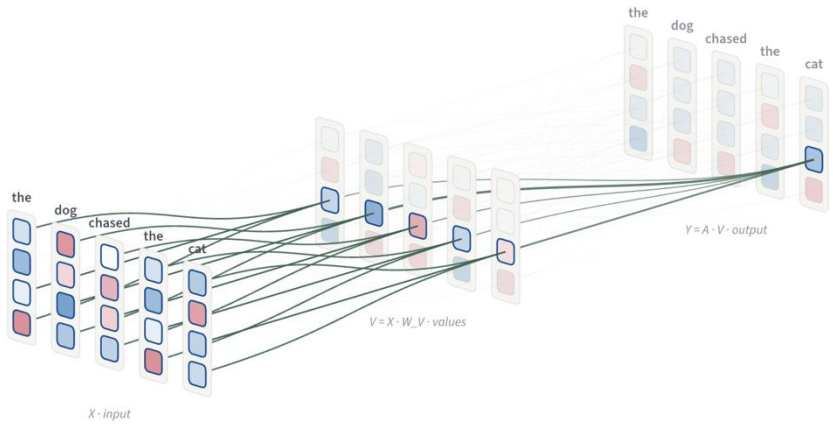


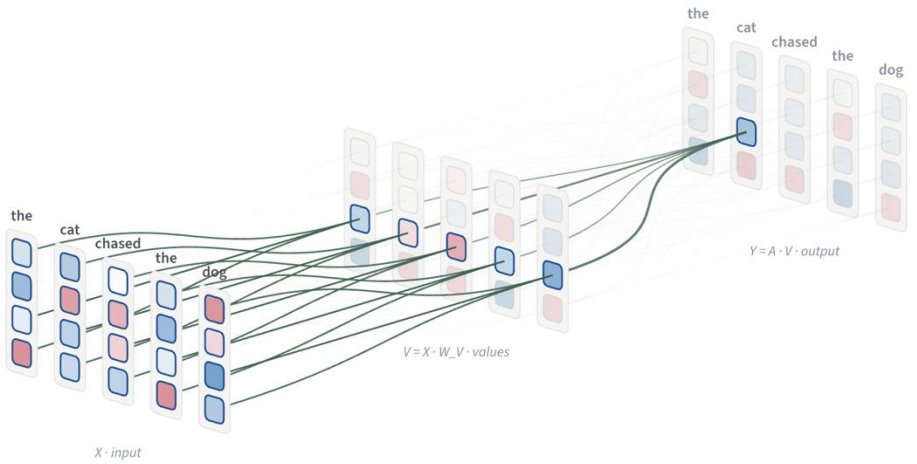


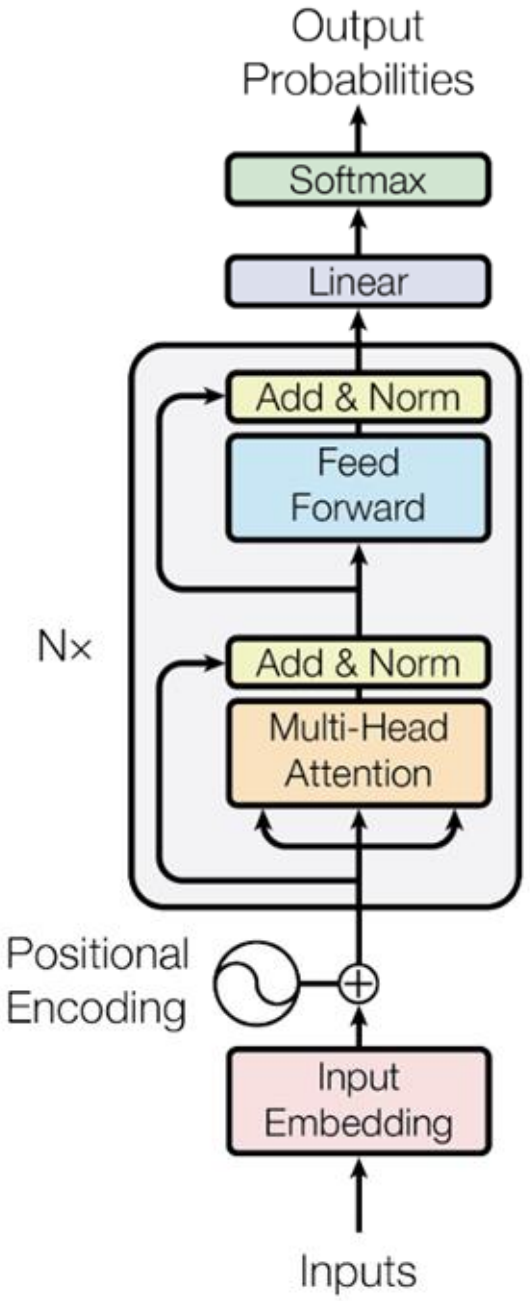
Position invariance











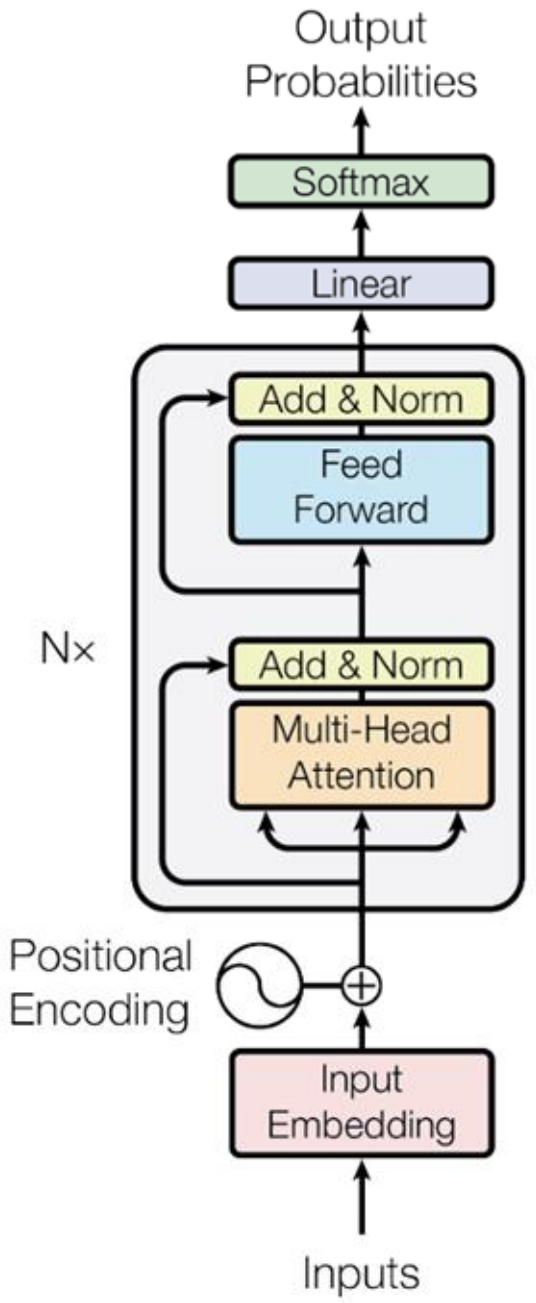
Positional embeddings (encoding)

Word embeddings

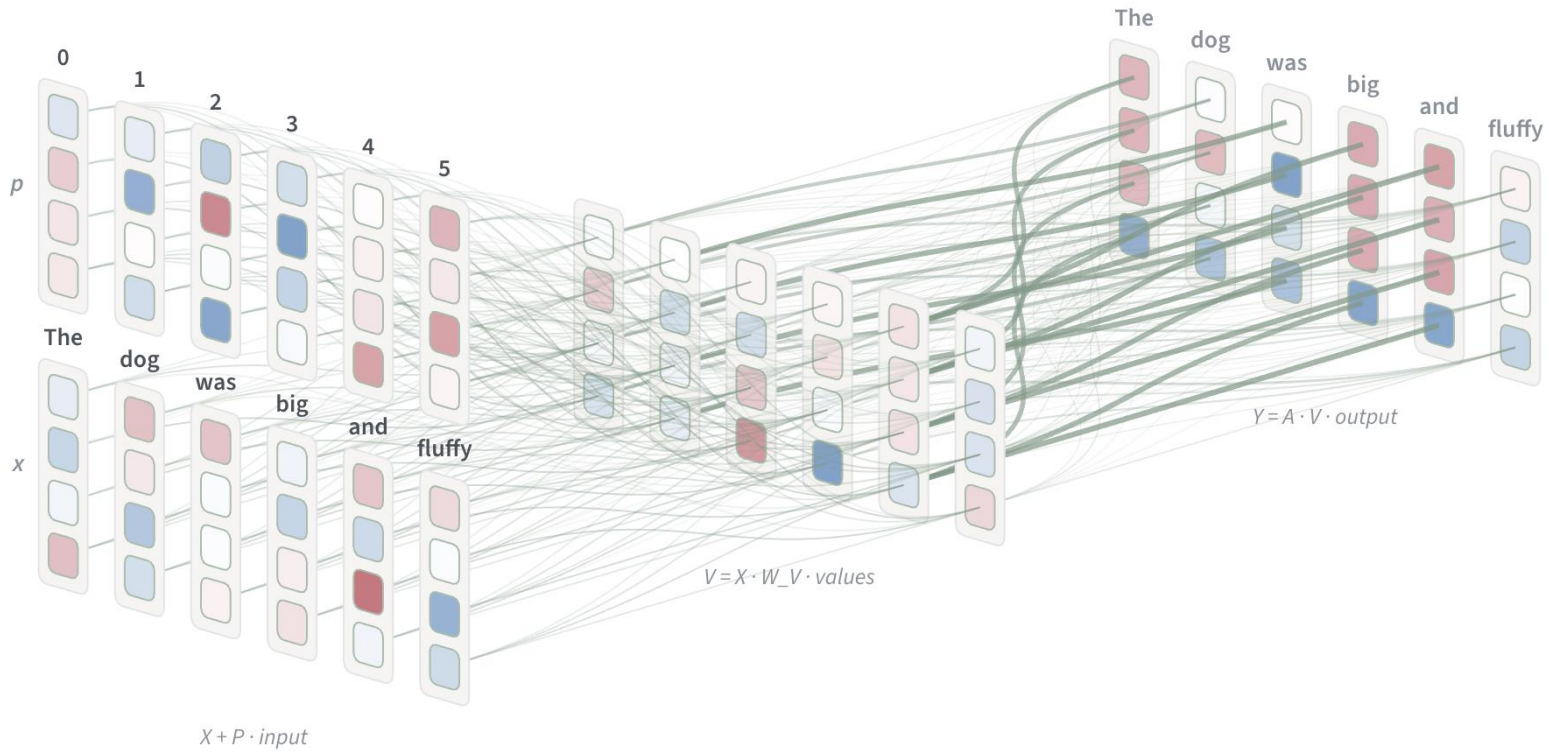
big *word vector*	-.03	-.08	.04	-.14
dog *word vector*	-.18	-.00	-.08	.07
was *word vector*	.01	.08	.04	.04

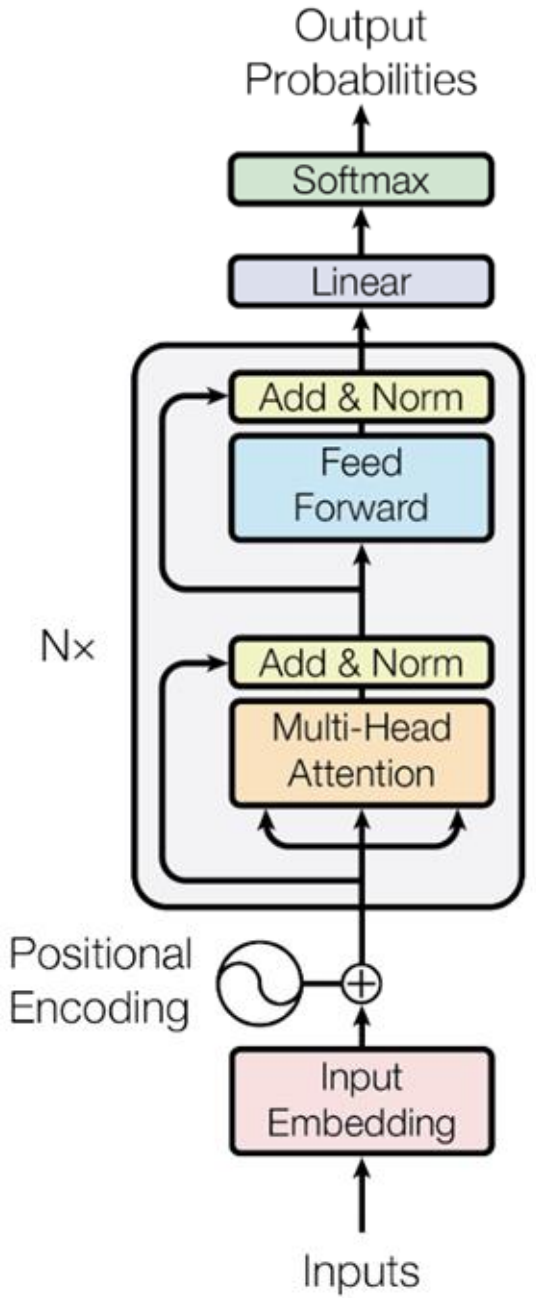
Position embeddings

0	.08	.10	.00	-.17
1	.10	.00	-.20	-.05
2	.09	.05	-.11	.03

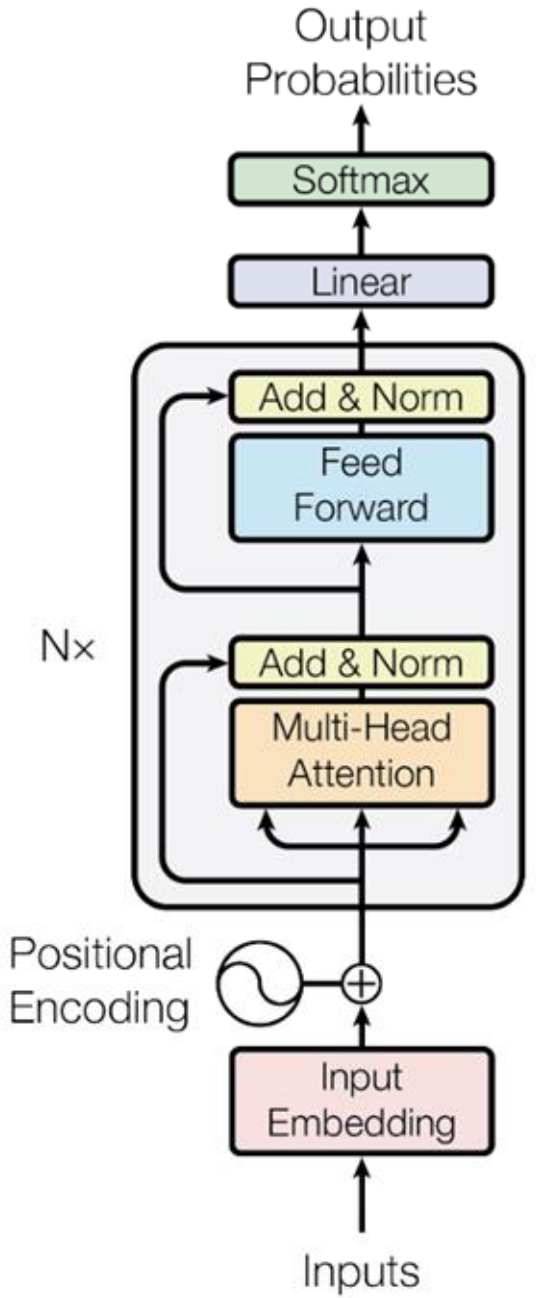


Positional embeddings

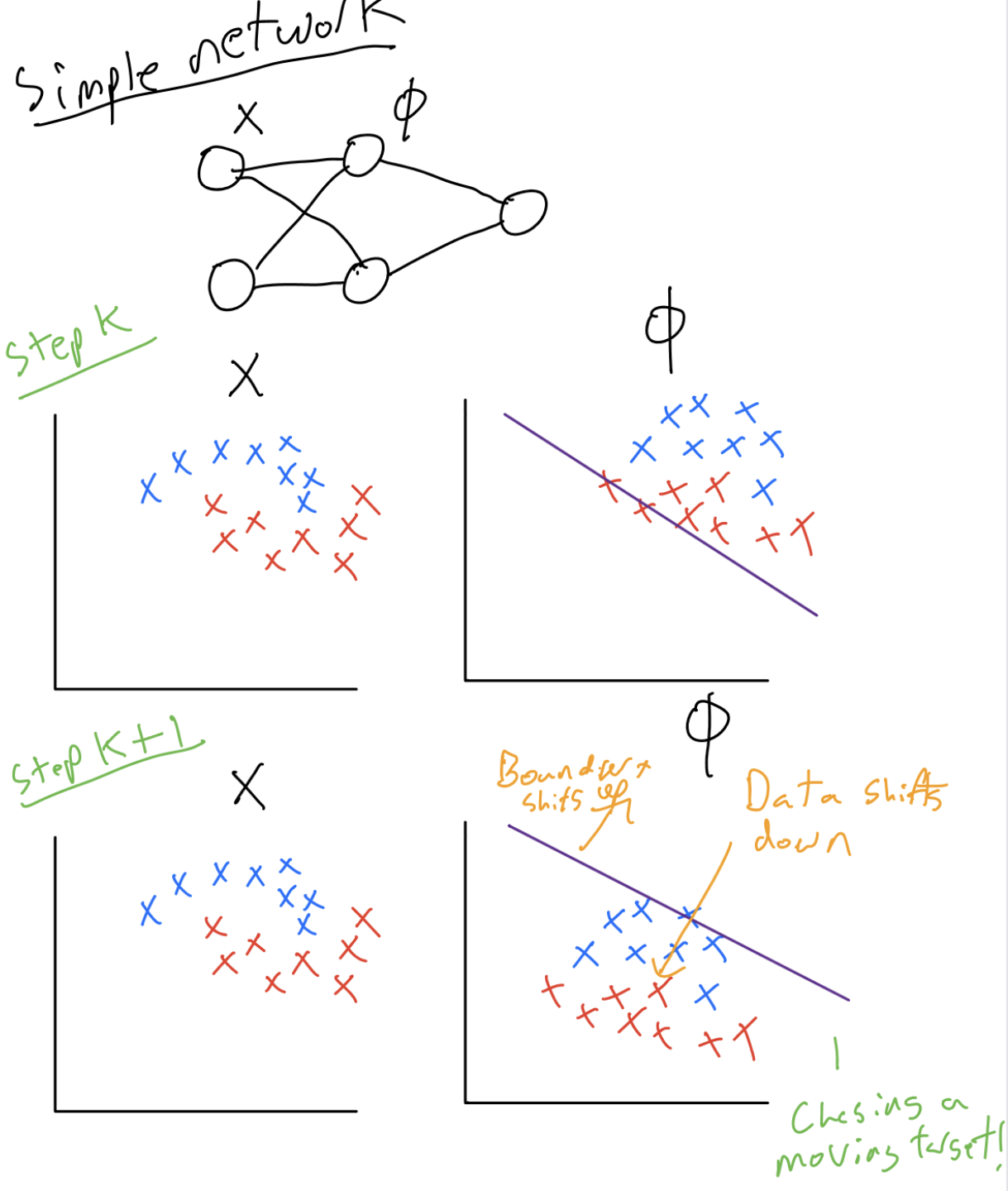




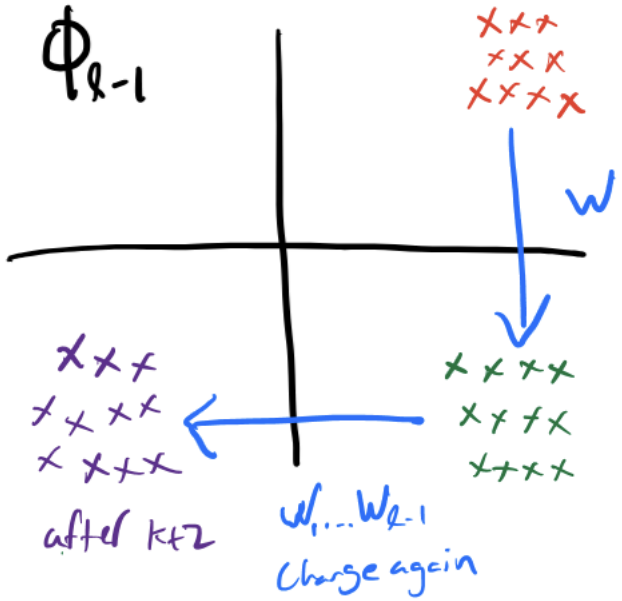
Norm, who's Norm?

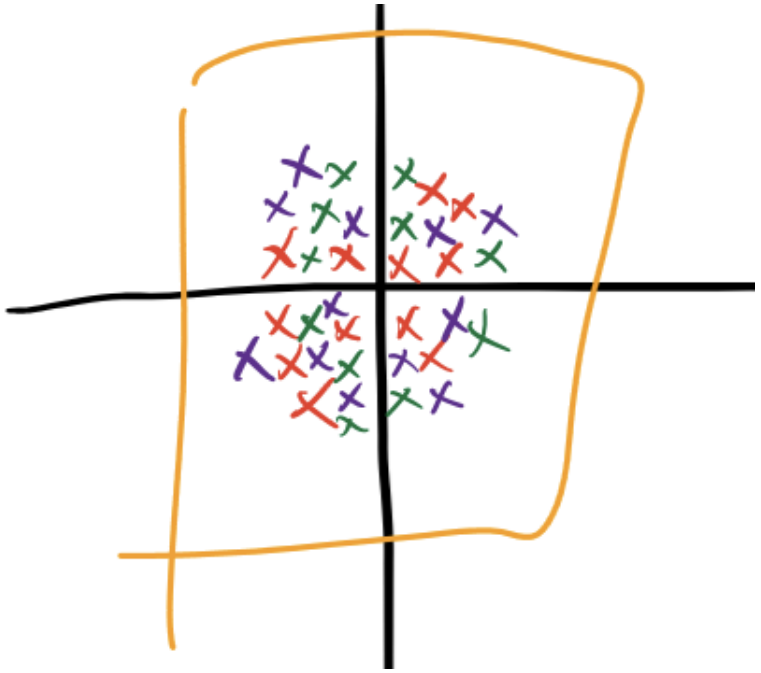


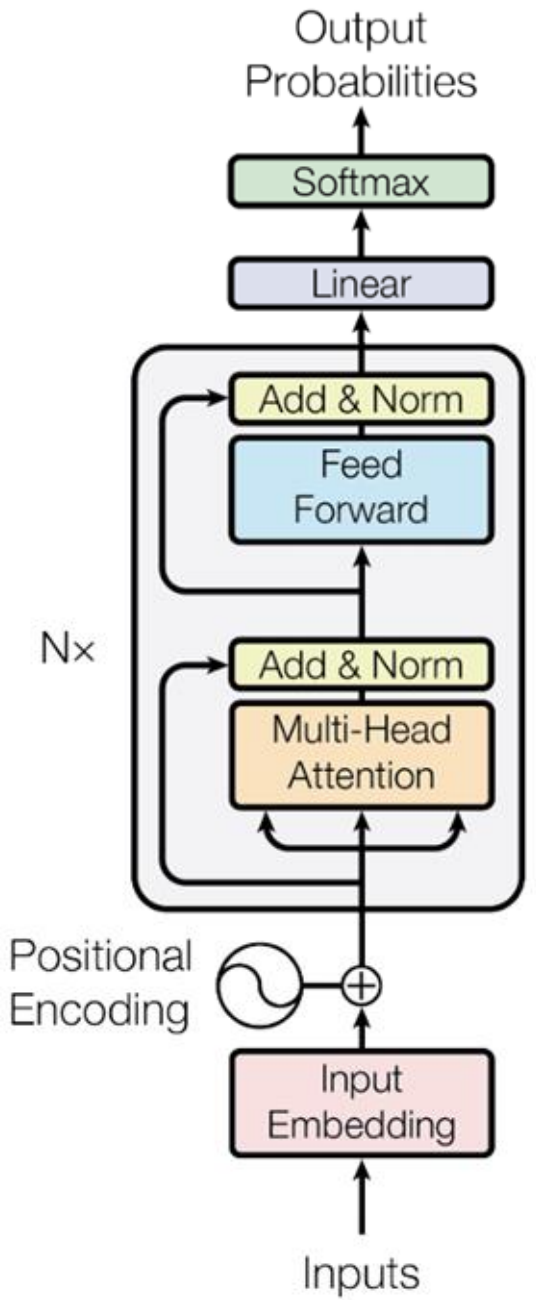
Chasing a moving target



Layer Normalization







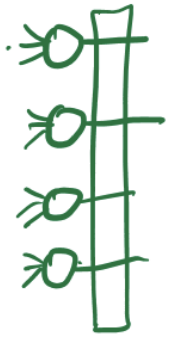
Layer Normalization Details

Layer Norm

$$\text{LayerNorm}(x_i) = \frac{x_i - \bar{x}_i}{\sqrt{s_i^2 + \varepsilon}}$$

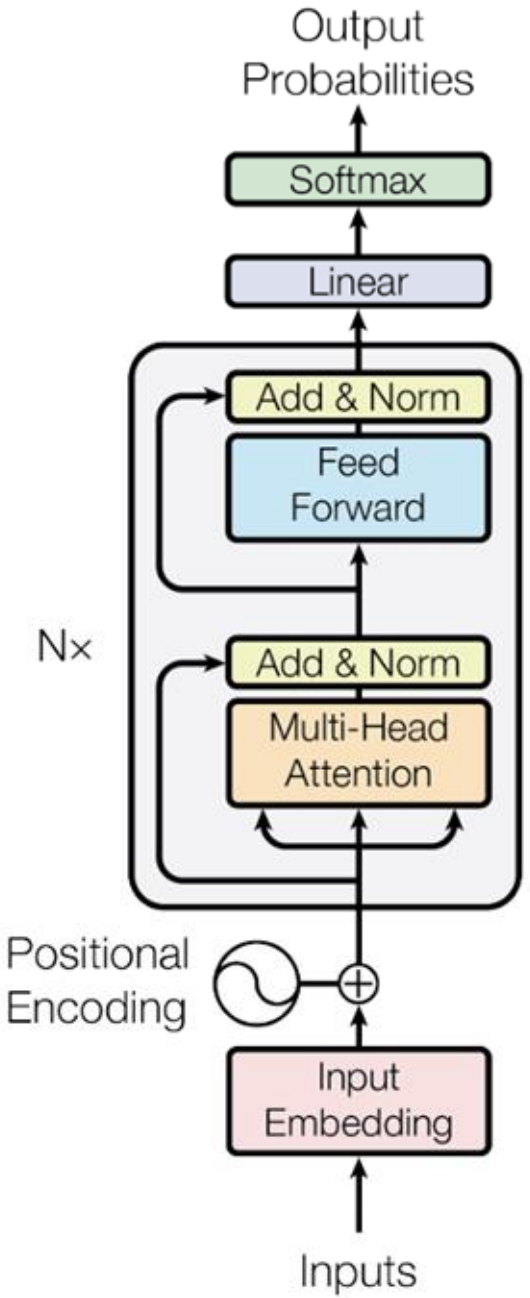
$$\bar{x}_i = \frac{1}{d}\sum_{j=1}^{d} x_{ij} \qquad s^2 = \frac{1}{d-1}\sum_{j=1}^{d}(x_{ij} - \bar{x}_i)^2$$

In network

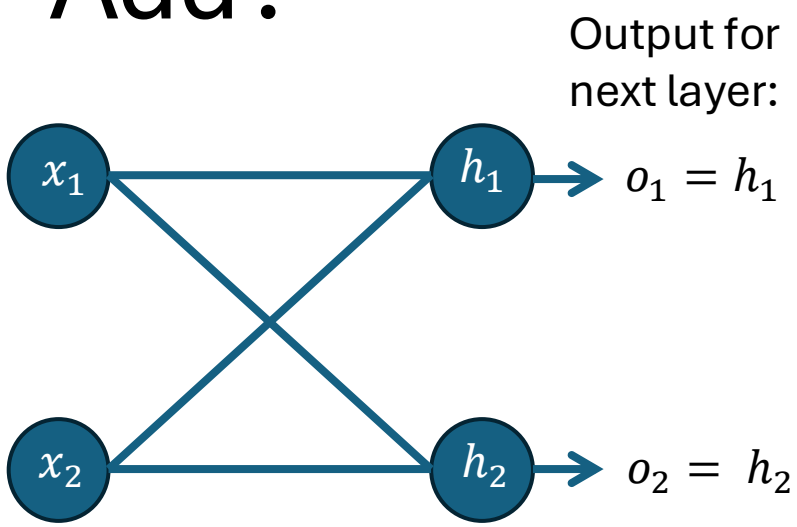


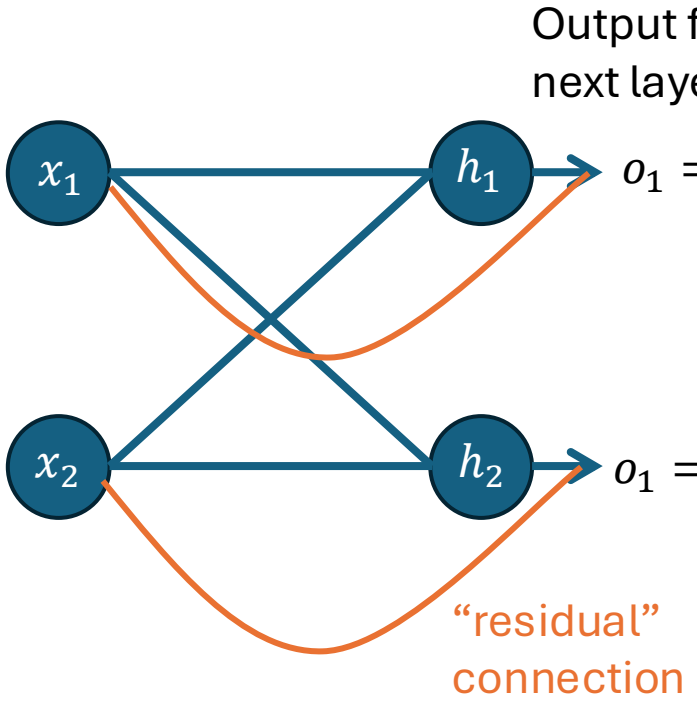
$E[x] = 0, \text{var}[x] = 1$

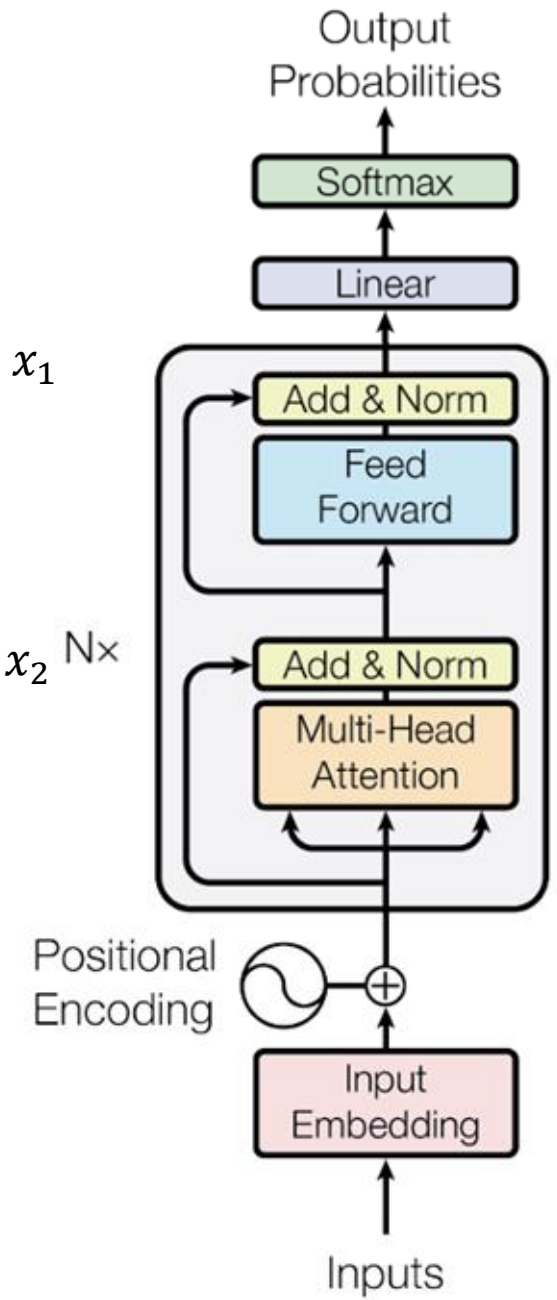
Can be applied w/ Batch size 1! As long as $d > 1$



Add?





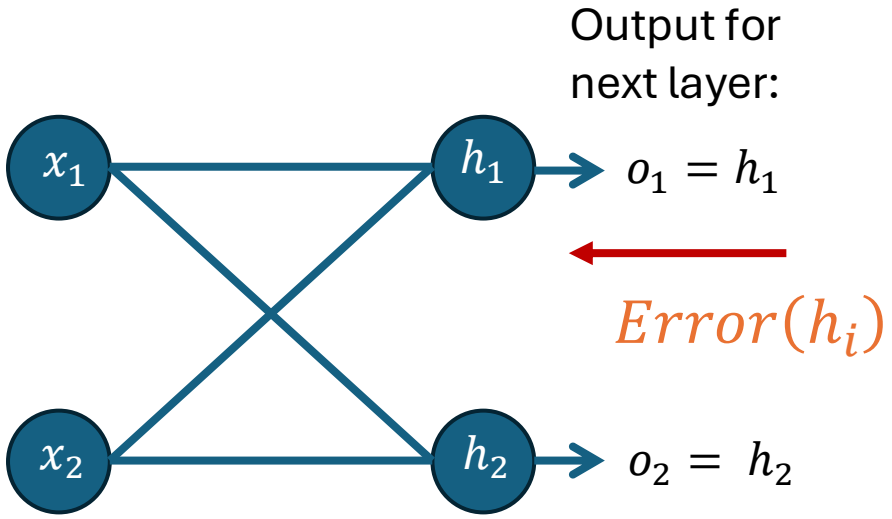


Neuron 1 activation (sigmoid activation func.):

$$\sigma\left(\sum_{i=1}^{n} w_{i,1}x_i + b_1\right)$$

Neuron 1 activation w/ residual (sigmoid activation func.):

$$\sigma\left(\sum_{i=1}^{n} w_{i,1}x_i + b_1\right) + x_1$$

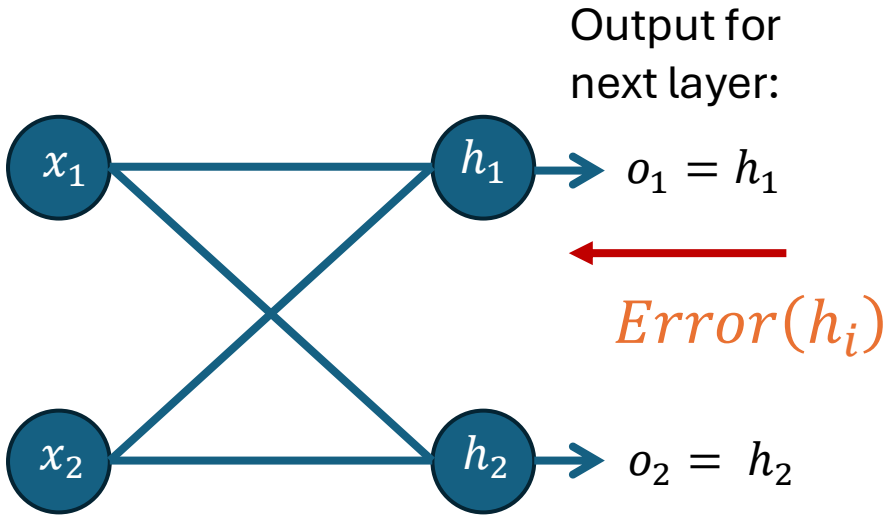


Why do this?

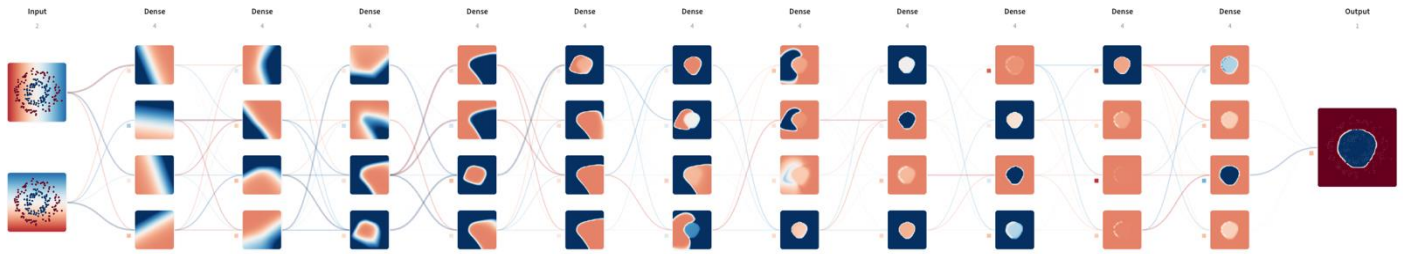
$$h_1 = \sigma\left(\sum_{i=1}^{n} w_{i,1}x_i + b_1\right)$$

$$Error(x_1) = \sum_{i=1}^{n} Error(h_i)(h_i)(1 - h_i)w_{1,i}$$

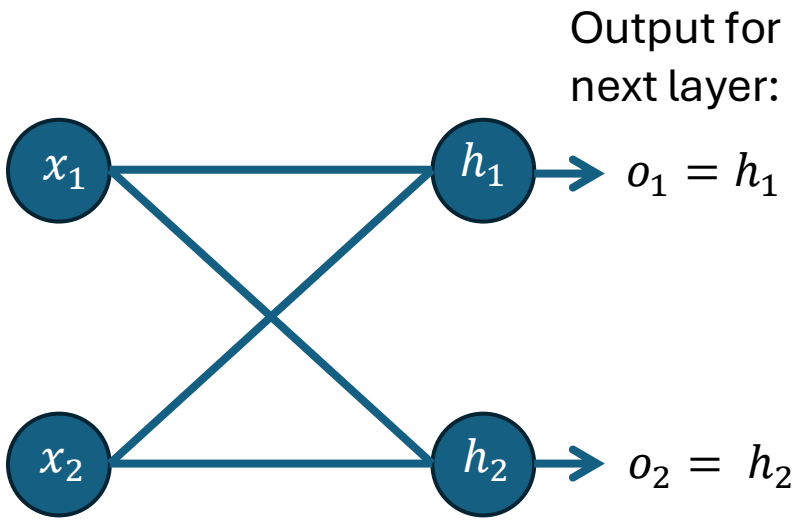
Why do this?

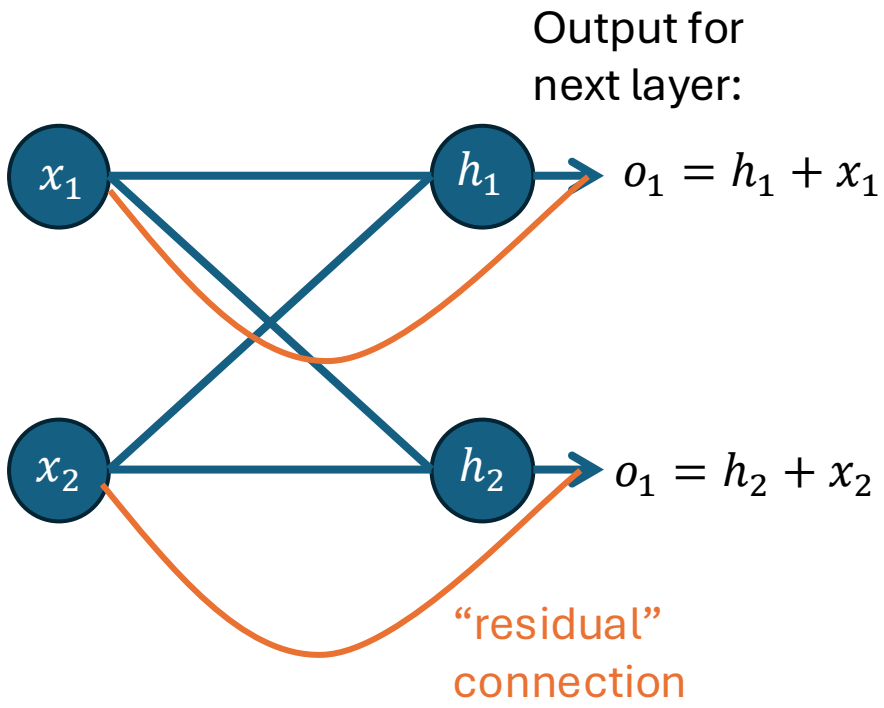


$$h_1 = \sigma\left(\sum_{i=1}^{n} w_{i,1}x_i + b_1\right)$$



$$Error(x_1) = \sum_{i=1}^{n} Error(h_i)(h_i)(1 - h_i)w_{1,i}$$





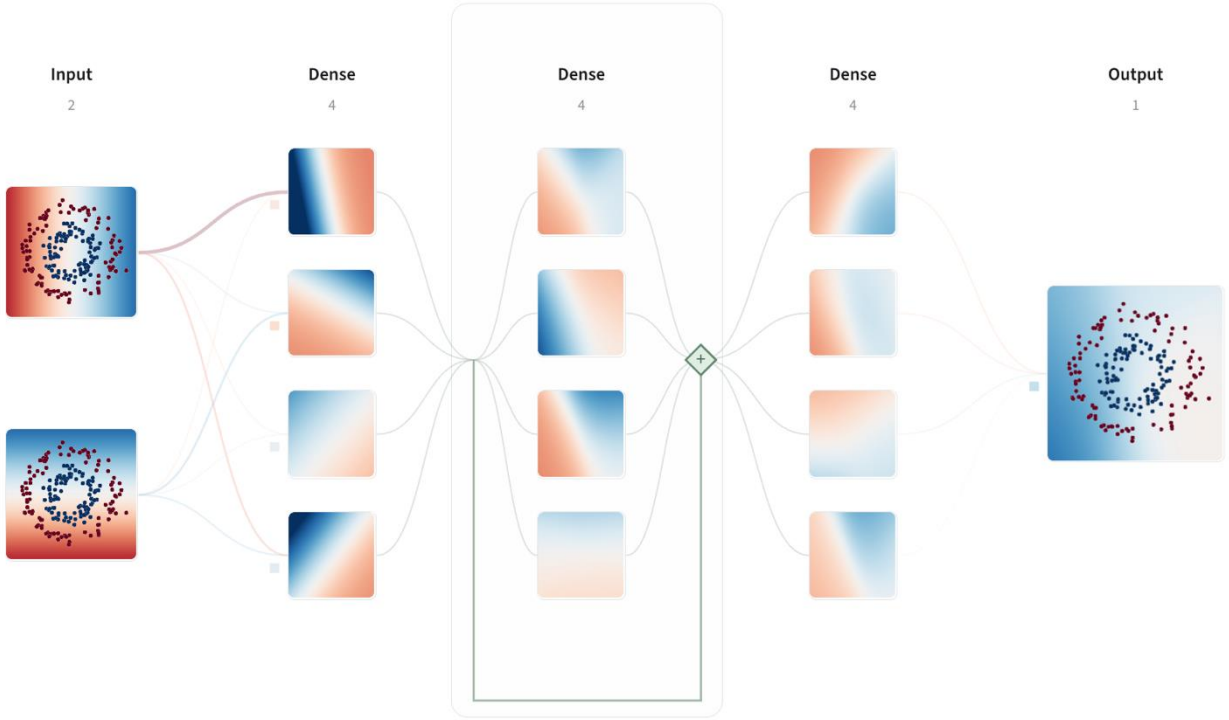
$$\sigma\left(\sum_{i=1}^{n} w_{i,1}x_i + b_1\right)$$

$$\sigma\left(\sum_{i=1}^{n} w_{i,1}x_i + b_1\right) + x_1$$

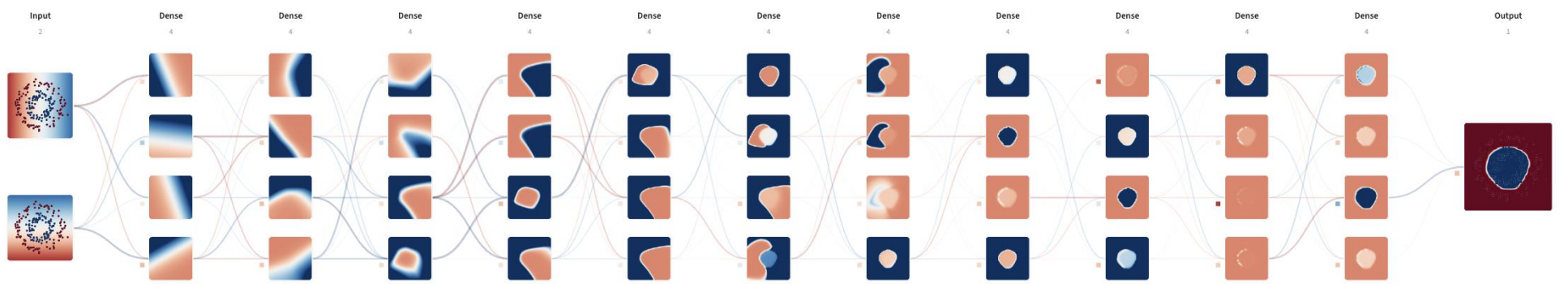
$$Error(x_1) = \sum_{i=1}^{n} Error(h_i)(h_i)(1-h_i)w_{1,i}$$

$$Error(x_1) = Error(h_1) + \sum_{i=1}^{n} Error(h_i)(h_i)(1-h_i)w_{1,i}$$

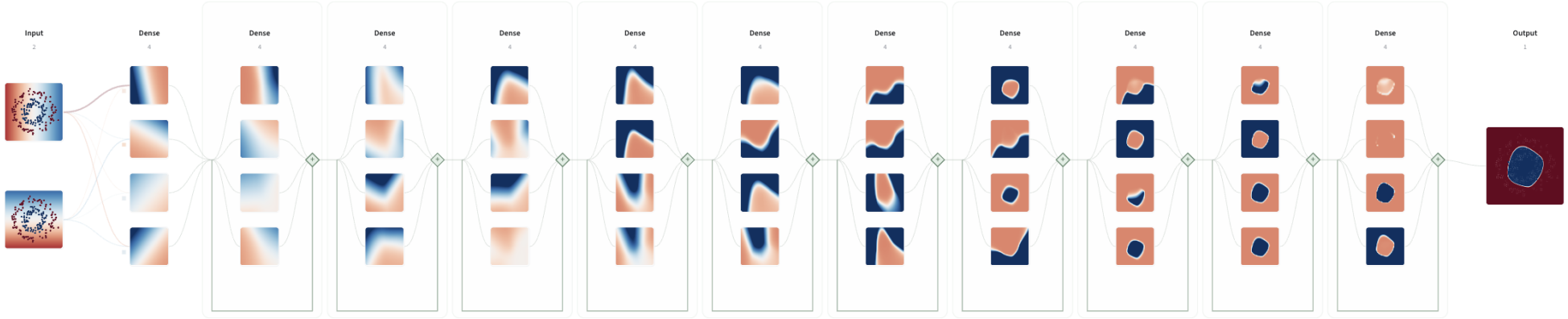
Effect of Residual connections



Effect of Residual connections



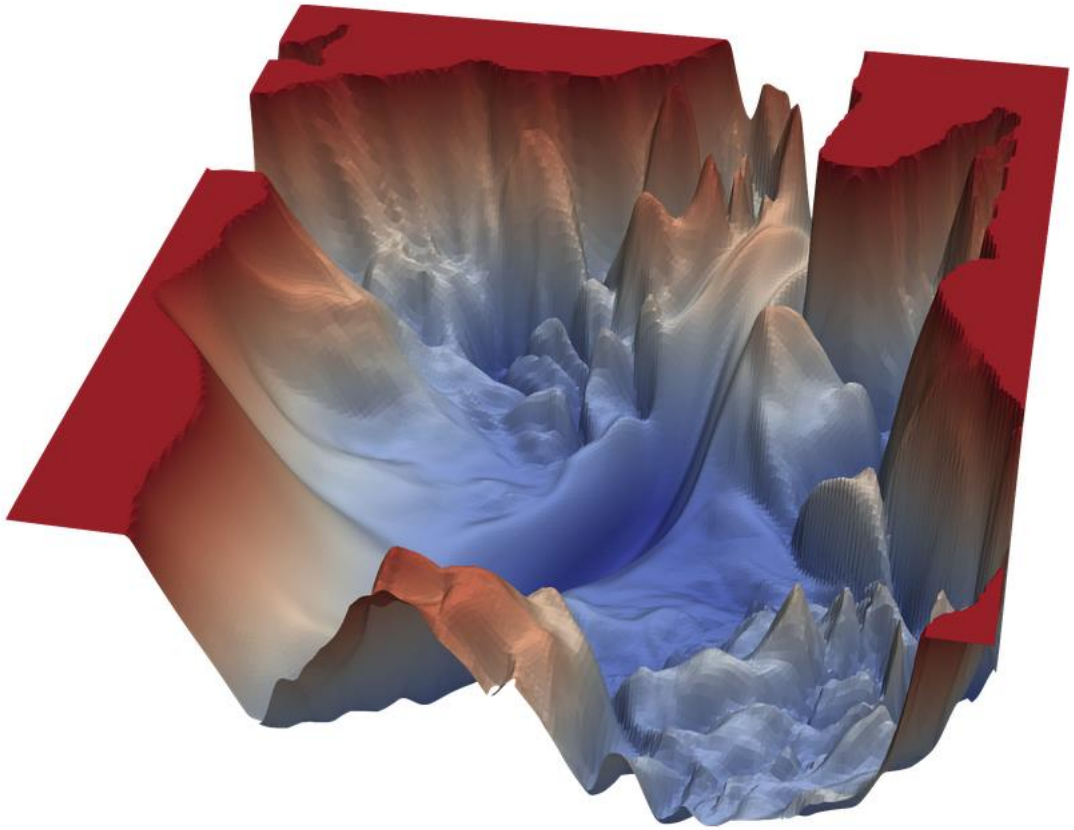




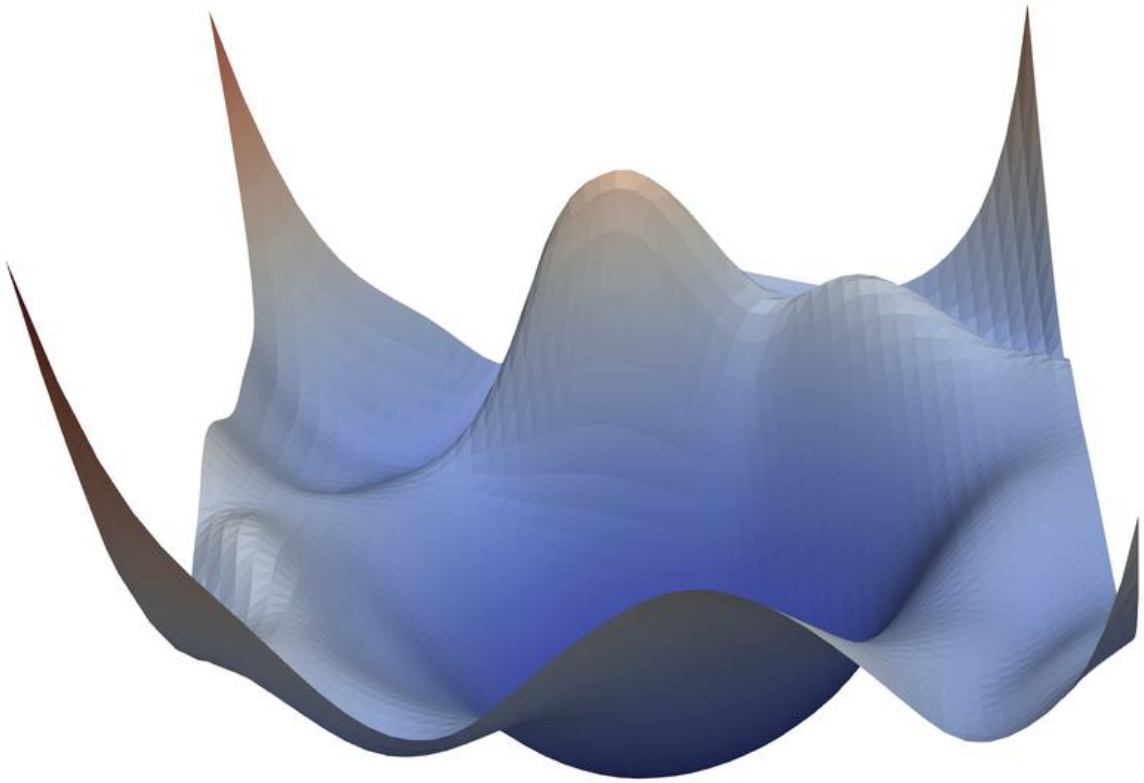


Effect of Residual connections

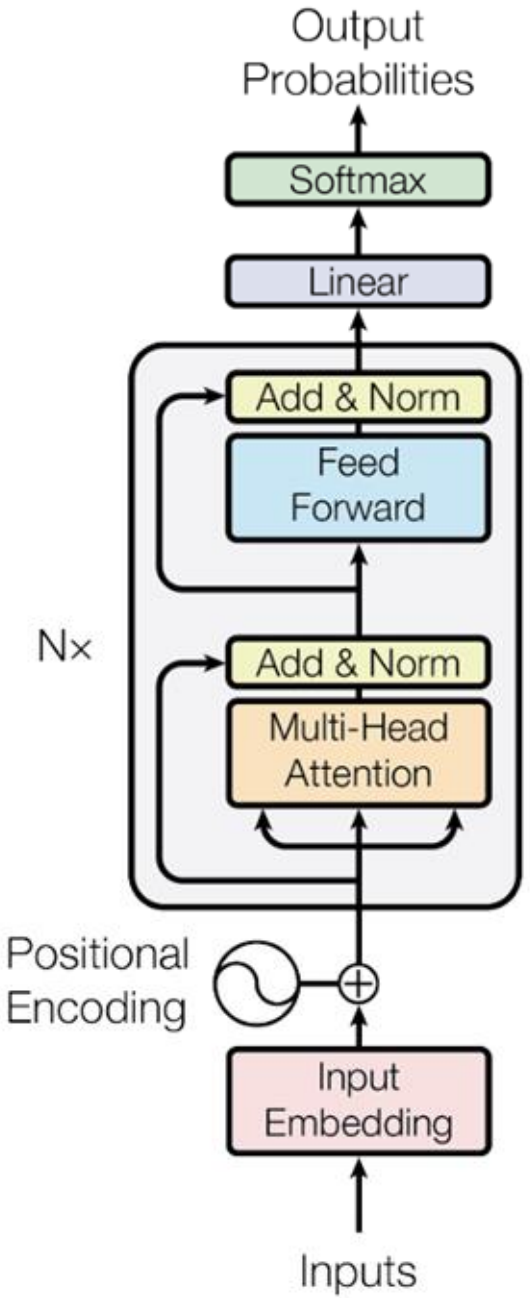
VGG-56



Renset-56



Done! Demo time



Aside: Residuals and normalization in CNNs

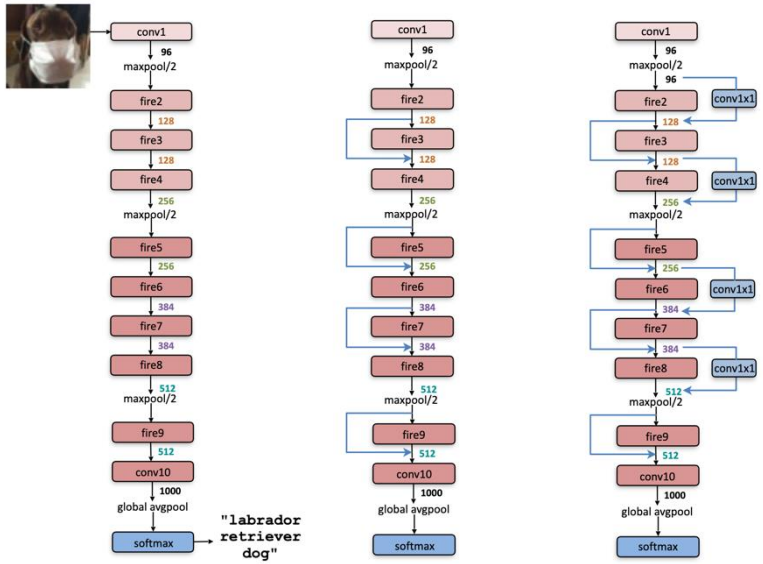
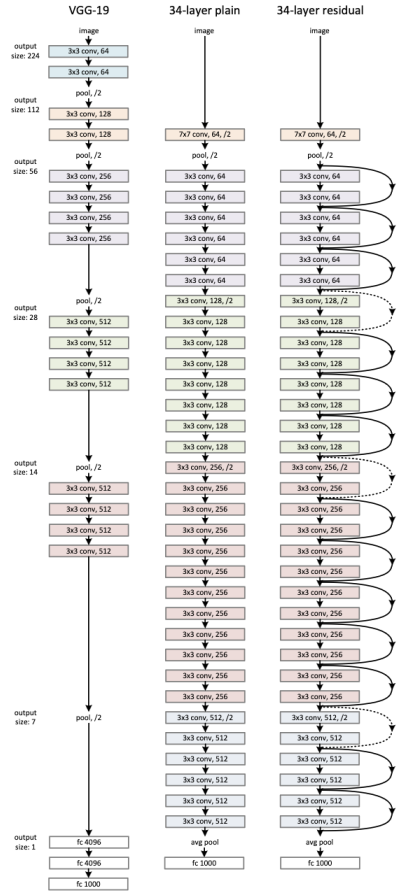
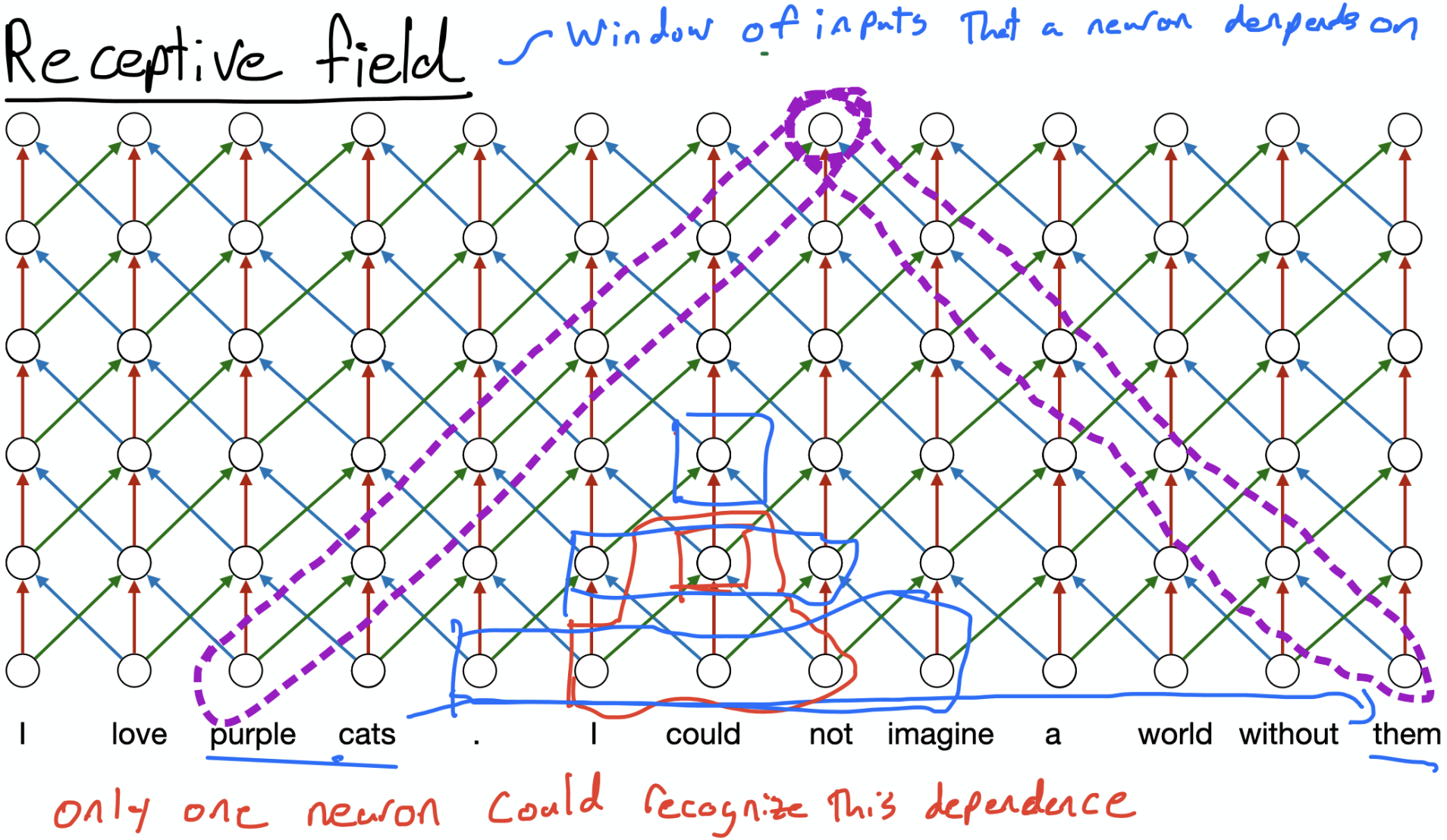


Figure 2: Macroarchitectural view of our SqueezeNet architecture. Left: SqueezeNet (Section 3.3); Middle: SqueezeNet with simple bypass (Section 6); Right: SqueezeNet with complex bypass (Section 6).



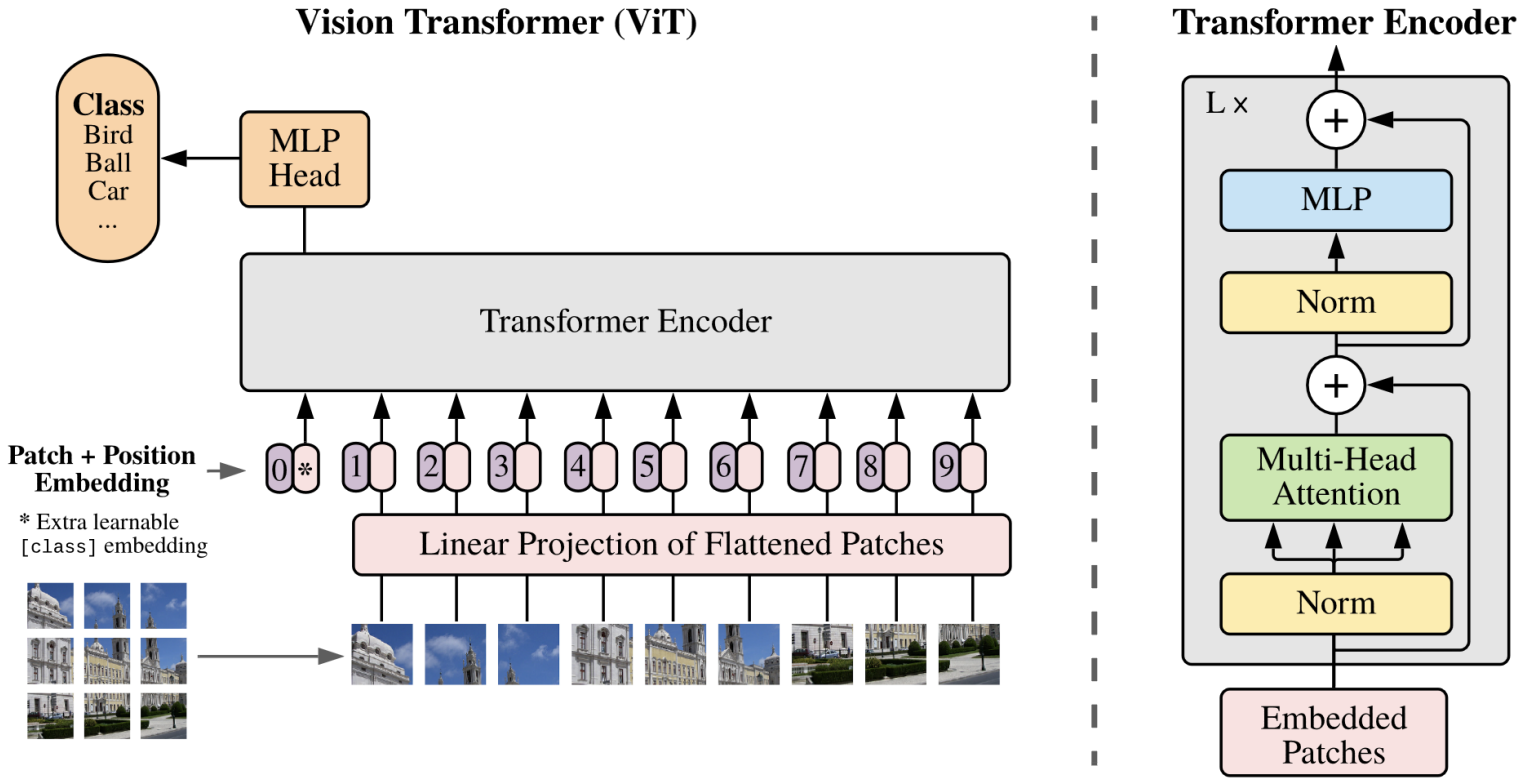
Why not CNNs for language?



Immediate long range dependencies

Geoffrey Everest Hinton CC FRS FRSC[12] (born 6 December 1947) is a British-Canadian computer scientist and cognitive psychologist, most noted for his work on artificial neural networks. From 2013 to 2023, he divided his time working for Google (Google Brain) and the University of Toronto, before publicly announcing his departure from Google in May 2023, citing concerns about the risks of artificial intelligence (AI) technology.[13] In 2017, he co-founded and became the chief scientific advisor of the Vector Institute in Toronto.[14][15]

Vision transformers



Transformers, summarized