Carnegie Mellon University

Neural Language Model Architectures

Large Language Models: Methods and Applications

Daphne Ippolito and Chenyan Xiong

Updates

- Homework 1 will be out today or tomorrow.
  - Homework 1: all about training data.
- Homework 1 will be due September 10.
- TA office hours start up next week.
  - Mondays 1 PM - 2 PM
  - Thursdays 11 AM -12 PM
- If you are still on the waitlist, you can still start doing the homework.
- Important: fill out the AWS survey on Canvas

Recap of last class

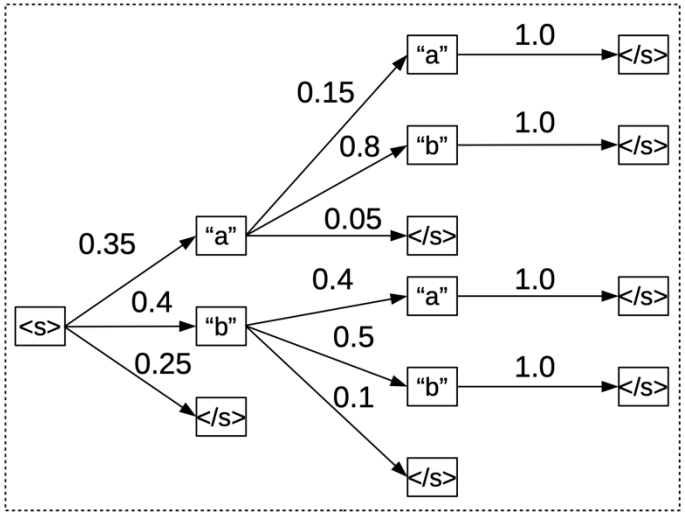
- Tokenization is the process of turning text into a sequence of integers.
- Language models output the probability of a token given the previous tokens in a sequence.
  - $P(y_t \mid y_{1:t-1})$
- Sometimes, we also want to condition language models on some other input, X.
  - $P(y_t \mid y_{1:t-1}; X)$
- In the past, people used statistical language models.
- We choose between encoder-decoder vs. decoder-only model architectures, depending on the assumptions we want to make about how problem decomposes.

Quiz

More on Beam Search

- Let's walk through an example where arg max decoding and beam search won't give the same answer.

Greedy search methods do not always lead to the most likely output.

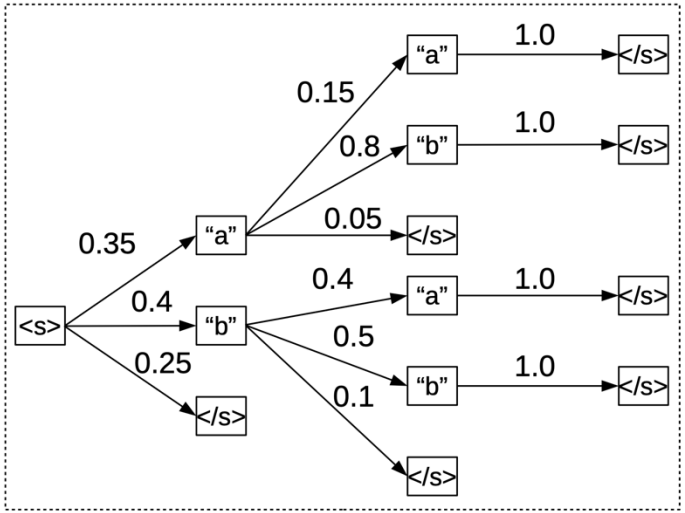


Vocabulary = {a, b, </s>}
Numbers above each edge are the transition probabilities $P(x_t|x_{1:t-t})$

If we were to choose the sequence that maximizes $P(x_1, \ldots, x_T)$, which of the following would get generated?

(a) [a, b, </s>]
(b) [a, a, </s>]
(c) [b, b, </s>]
(d) [b, a, </s>]

Greedy search methods do not always lead to the most likely output.

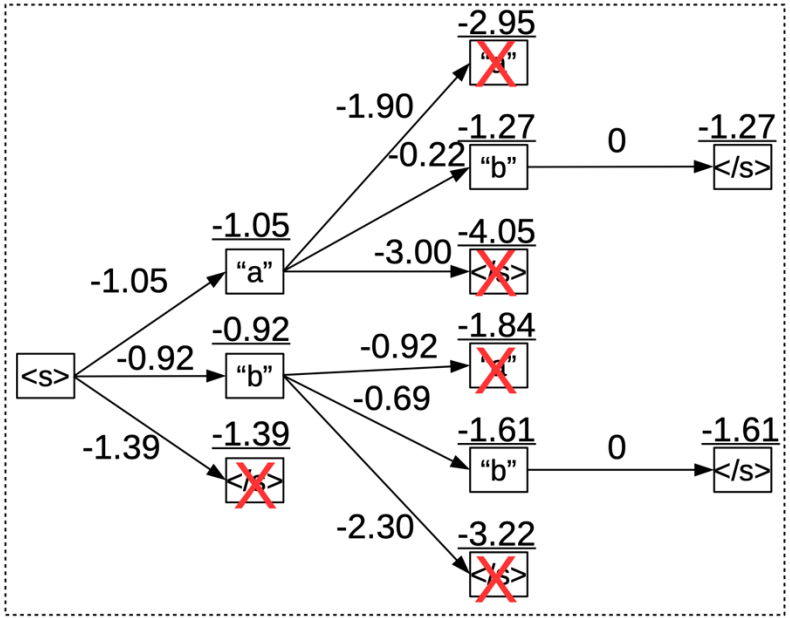


Vocabulary = {a, b, </s>}
Numbers above each edge are the transition probabilities $P(x_t|x_{1:t-t})$

If we were to choose the sequence that maximizes $P(x_1, \ldots, x_T)$, which of the following would get generated?

(a) [a, b, </s>]
(b) [a, a, </s>]
(c) [b, b, </s>]
(d) [b, a, </s>]

Beam search explores multiple possible output sequences, trying to find the overall most likely one.



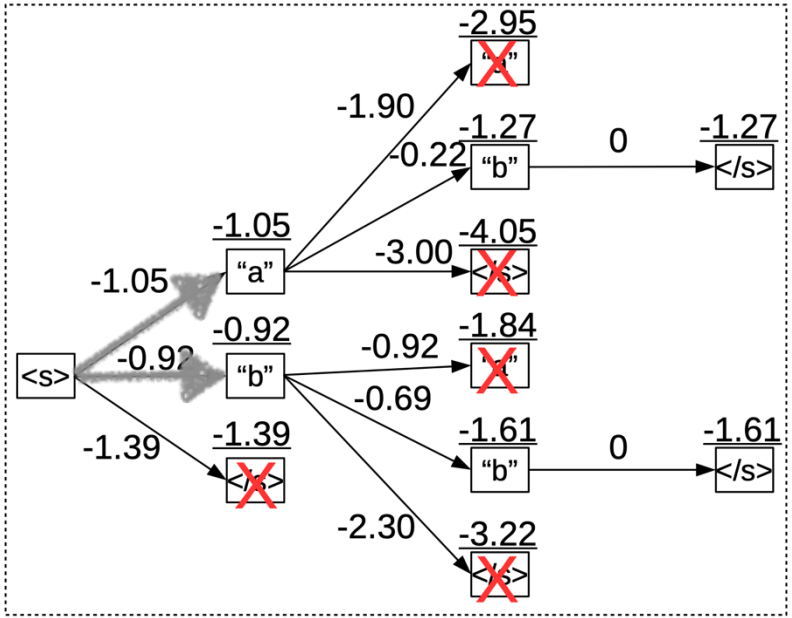
Vocabulary = {a, b, </s>}

Numbers above the boxes are $logP(x_t|x_{1:t-1})$

Numbers shown on edges are $logP(x_1, \ldots, x_t)$

Suppose we use beam search with a **beam size** of 2.

Beam search explores multiple possible output sequences, trying to find the overall most likely one.



Score each path and keep the top 2

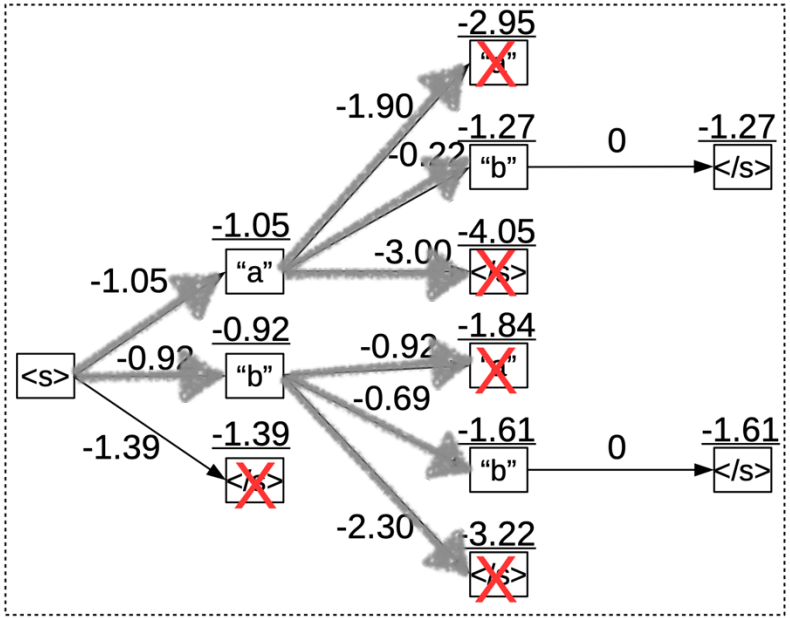
Vocabulary = {a, b, </s>}

Numbers above the boxes are $\log P(x_t|x_{1:t-1})$

Numbers shown on edges are $\log P(x_1, \ldots, x_t)$

Suppose we use beam search with a **beam size** of 2.

Beam search explores multiple possible output sequences, trying to find the overall most likely one.



Score each path and keep the top 2

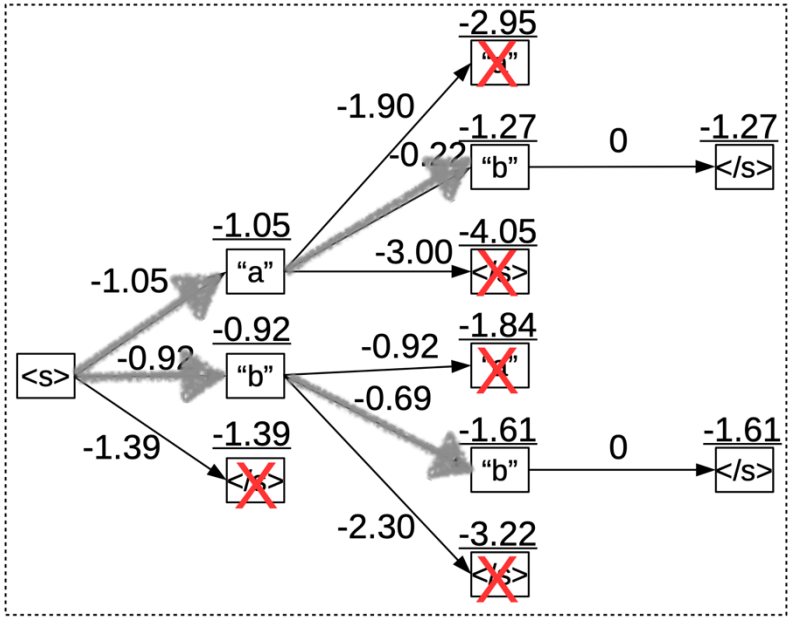
Vocabulary = {a, b, </s>}

Numbers above the boxes are $\log P(x_t|x_{1:t-1})$

Numbers shown on edges are $\log P(x_1, \ldots, x_t)$

Suppose we use beam search with a **beam size** of 2.

Beam search explores multiple possible output sequences, trying to find the overall most likely one.



Score each path and keep the top 2

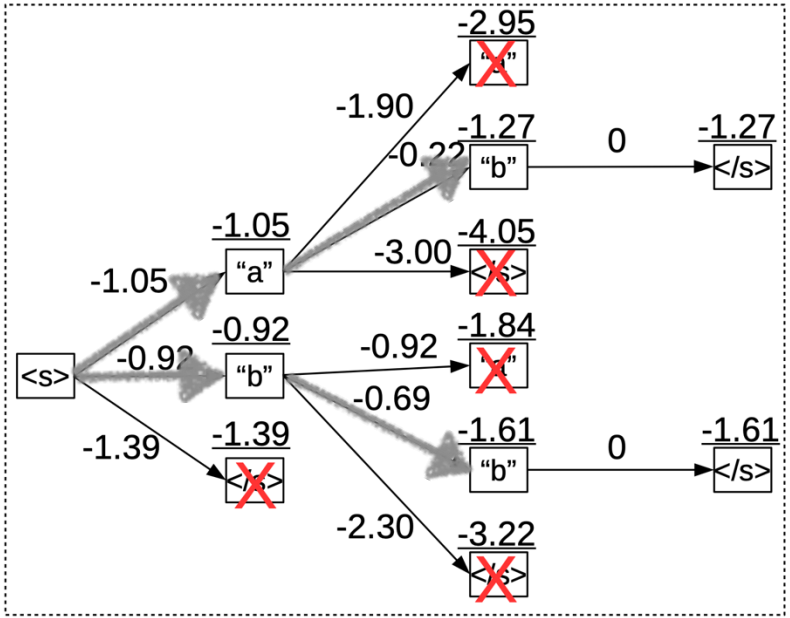
Vocabulary = {a, b, </s>}

Numbers above the boxes are $\log P(x_t|x_{1:t-1})$

Numbers shown on edges are $\log P(x_1, \ldots, x_t)$

Suppose we use beam search with a **beam size** of 2.

Beam search explores multiple possible output sequences, trying to find the overall most likely one.



Score each path and keep the top 2

Vocabulary = {a, b, </s>}

Numbers above the boxes are $\log P(x_t|x_{1:t-1})$

Numbers shown on edges are $\log P(x_1, \ldots, x_t)$

Suppose we use beam search with a **beam size** of 2.

The paths that are still open at any step of the beam search algorithm are called **beams**.

Back to where we left off: attention

How did RNN-based language models connect the encoder with the decoder?

Better approach: an attention mechanism.

[The, hippopotamus, ...

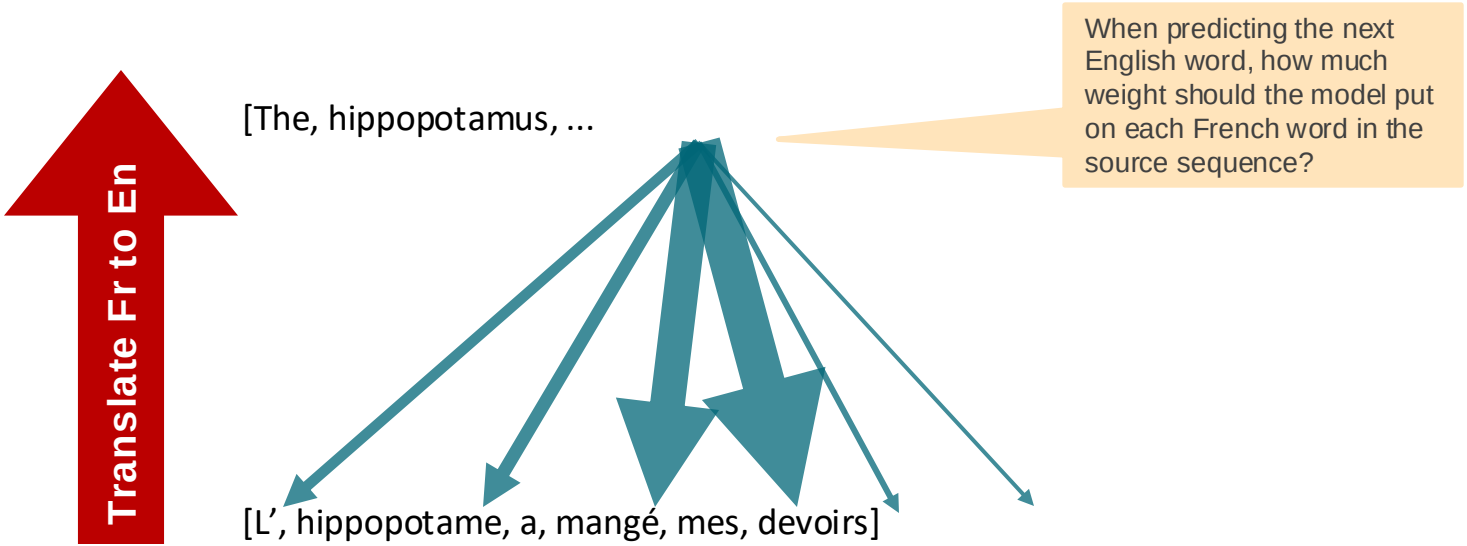
When predicting the next English word, how much weight should the model put on each French word in the source sequence?

[L', hippopotame, a, mangé, mes, devoirs]

Bahdanau, Dzmitry, Kyunghyun Cho, and Yoshua Bengio. "Neural machine translation by jointly learning to align and translate." (2014).

How did RNN-based language models connect the encoder with the decoder?

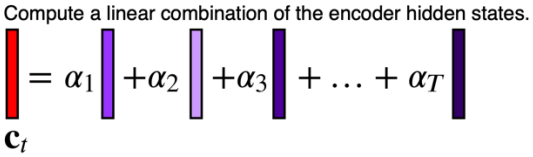
Better approach: an attention mechanism.



Bahdanau, Dzmitry, Kyunghyun Cho, and Yoshua Bengio. "Neural machine translation by jointly learning to align and translate." (2014).

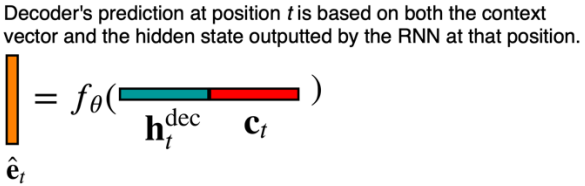
Attention Mechanism

At each step t in the decoder, a context vector is computed which contains all the information from the encoder that is relevant to the decoder making a prediction at this position.



The context vector is a linear sum of the encoder hidden states, i.e., $\boldsymbol{c}_t = \mathbf{H}^{\text{enc}}\boldsymbol{\alpha}_t$.

The decoder's predicted embedding for position t is a function of the context vector and the decoder's hidden state for this position.



$$\hat{\mathbf{e}}_t = f_\theta(, \mathbf{h}_t^{\text{dec}};\ \alpha_{1,t}\mathbf{h}_1^{\text{enc}} + \alpha_{2,t}\mathbf{h}_2^{\text{enc}} + \cdots \alpha_{T,t}\mathbf{h}_T^{\text{enc}})$$

Computing the Attention Weights

The $\alpha_{i,j}$ are scores that indicate how important the encoder hidden state at position i is to the model's prediction at position j. They are typically normalized to sum to 1.

$$\alpha_{i,j} = \frac{\exp e_{i,j}}{\sum_{k=1}^{T} \exp e_{i,k}}$$

Softmax function

$$e_{i,j} = \text{score}(\mathbf{h}_i^{\text{enc}}, \mathbf{h}_{j-1}^{\text{dec}})$$

Computing the Attention Weights

The $\alpha_{i,j}$ are scores that indicate how important the encoder hidden state at position i is to the model's prediction at position j. They are typically normalized to sum to 1.

$$\alpha_{i,j} = \frac{\exp e_{i,j}}{\sum_{k=1}^{T} \exp e_{i,k}}$$ ← Softmax function

$$e_{i,j} = \text{score}(\mathbf{h}_i^{\text{enc}}, \mathbf{h}_{j-1}^{\text{dec}})$$

In **dot-product attention**, we use a very simple scoring function: $\text{score}(\mathbf{q}, \mathbf{k}) = \mathbf{q} \cdot \mathbf{k}$

“At the core of an attention-based approach is the ability to compare an item of interest to a collection of other items in a way that reveals their relevance in the current context.”

-Jurafsky and Martin, Chapter 10

Circa 2017: Transformers

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

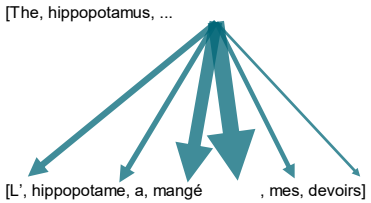
Łukasz Kaiser* Google Brain lukaszkaiser@google.com

Illia Polosukhin* ‡ illia.polosukhin@gmail.com

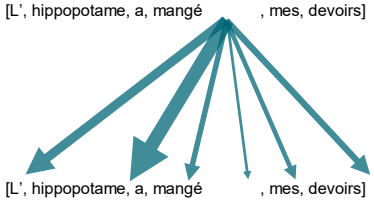
Abstract

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.

Encoder-decoder attention:



Self-attention:



Why drop the recurrence and only use attention?

- Recurrent neural networks are slow to train. Computation cannot be parallelized.
    - The computation at position *t* is dependent on first doing the computation at position *t*-1.
- Recurrent neural networks do poorly with long contexts.
    - If two tokens are *K* positions apart, there are *K* opportunities for knowledge of the first token to be erased from the hidden state before a prediction is made at the position of the second token.
- Transformers solve both these problems.

Components of a Generic Attention Mechanism

- A sequence of <key, value> embeddings pairs
  - The values are always the hidden states from a previous layer of the neural network. The attention mechanism outputs a weighted sum of these.
  - For encoder-decoder attention, the values are the final hidden states of the encoder (as we so in the previous slide) and the keys are the hidden states from the target sequence.
- A sequence of query embeddings
  - The query is the current focus of the attention.
  - We choose weights for each of the values by computing a score between the current query and each of the keys.

$$\text{attention output at position } j = \sum_{i=1}^{T} \text{score}(\mathbf{q}_j, \mathbf{k}_i) \cdot \mathbf{v}_i$$

$$\text{score}(\mathbf{q}_j, \mathbf{k}_i) = \frac{\mathbf{q}_j \cdot \mathbf{k}_i}{\sqrt{d_k}}$$

Components of a Generic Attention Mechanism

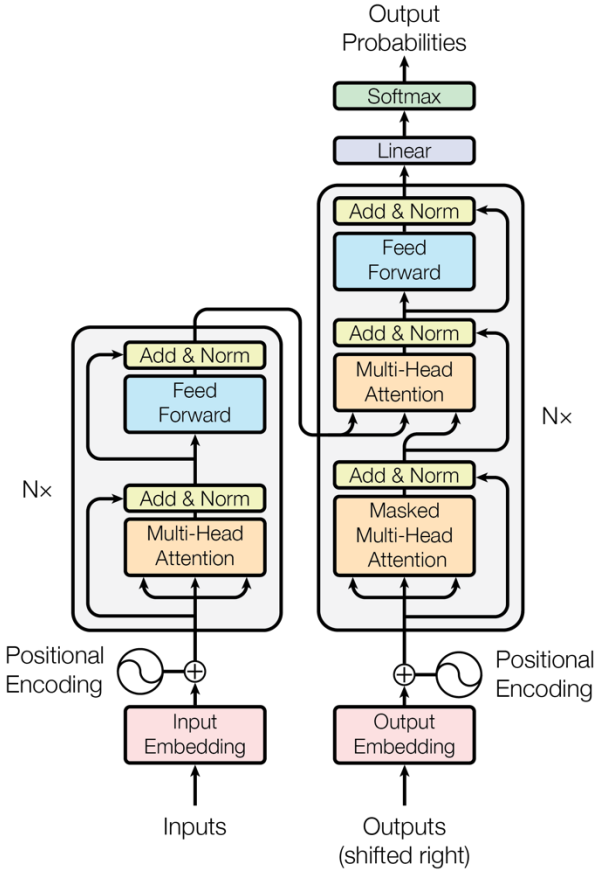
Since the attention computations at each position j are completely independent, we can actually parallelize all these computations and think in terms of matrix multiplications.

For example, instead of thinking of a sequence of embedding vectors $\mathbf{x}_1, \cdots, \mathbf{x}_T$ we can think of a matrix $\mathbf{X} \in \mathbb{R}^{T \times d_x}$.

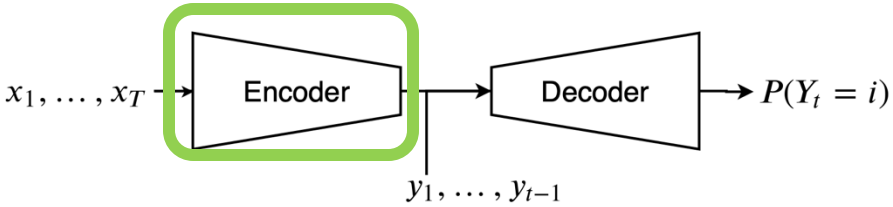
This gives us the attention equation which appear in the “Attention is All You Need” paper.

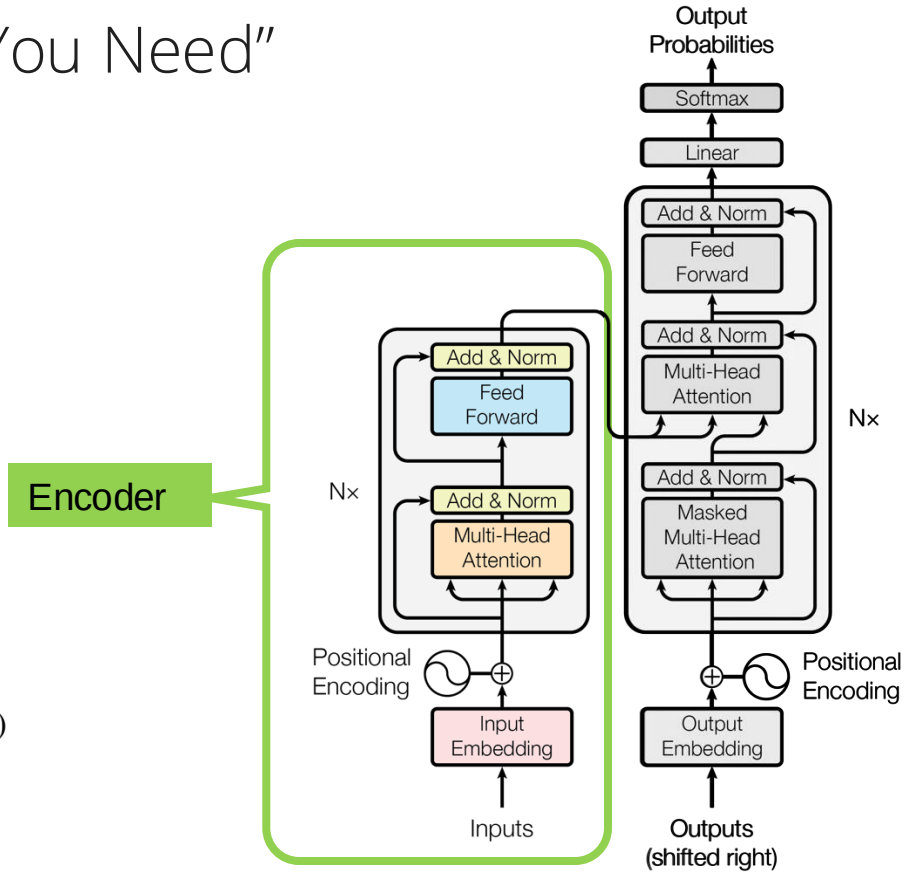
$$\text{attention}(\mathbf{Q}, \mathbf{K}, \mathbf{V}) = \text{softmax}\left(\frac{\mathbf{Q}\mathbf{K}^\top}{\sqrt{d_k}}\right)\mathbf{V}$$

Transformers: “Attention is All You Need”

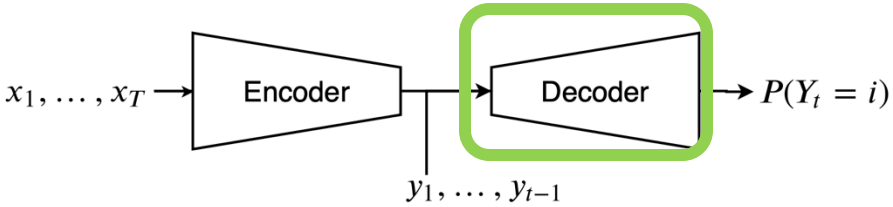


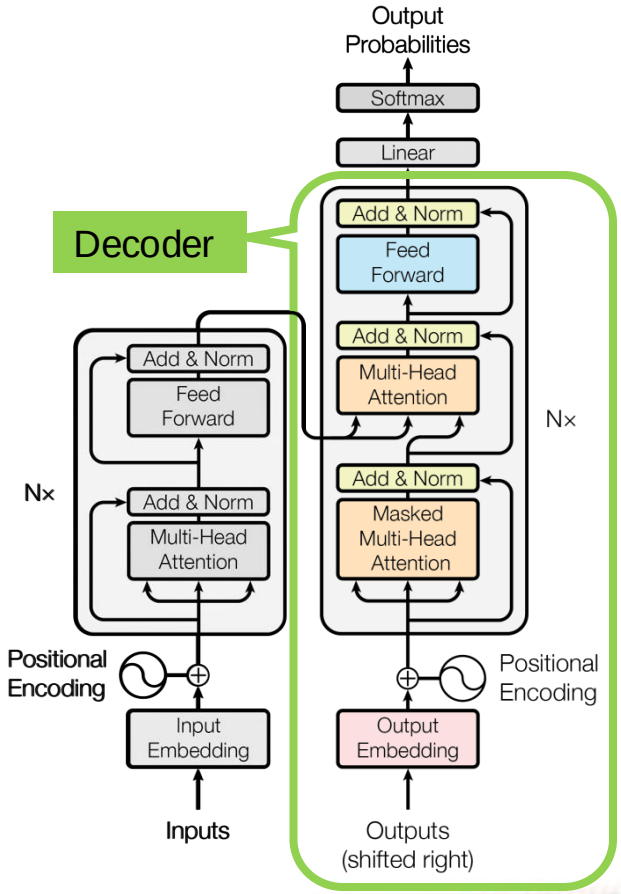
Transformers: “Attention is All You Need”



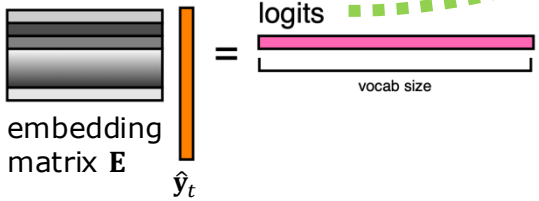


Transformers: “Attention is All You Need”

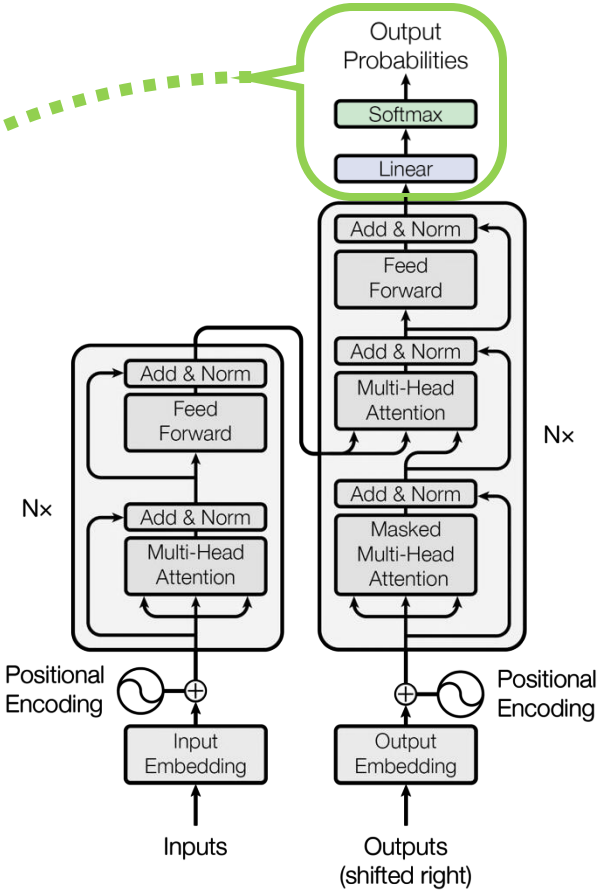




Transformers: “Attention is All You Need”

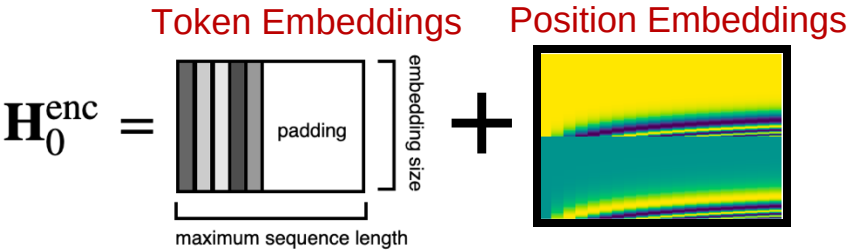


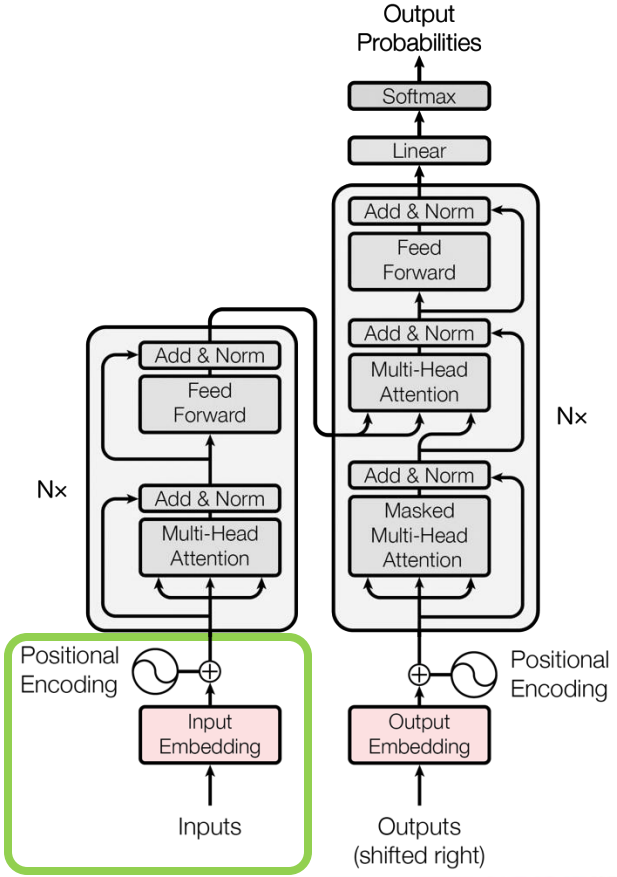
$$P(Y_t = i|\mathbf{x}_{1:T}, \mathbf{y}_{1:t-1}) = \frac{\exp(\mathbf{E}\hat{\mathbf{y}}_t[i])}{\sum_j \exp(\mathbf{E}\hat{\mathbf{y}}_t[j])}$$



Transformers: “Attention is All You Need”

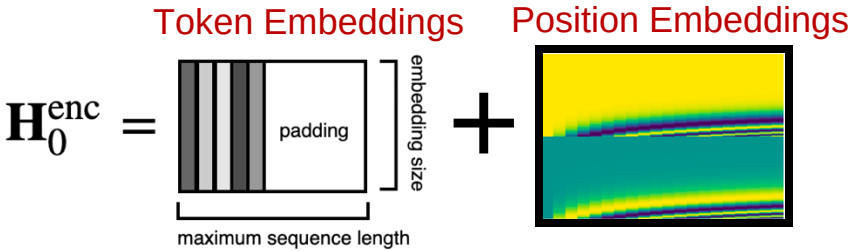
The input into the encoder looks like:



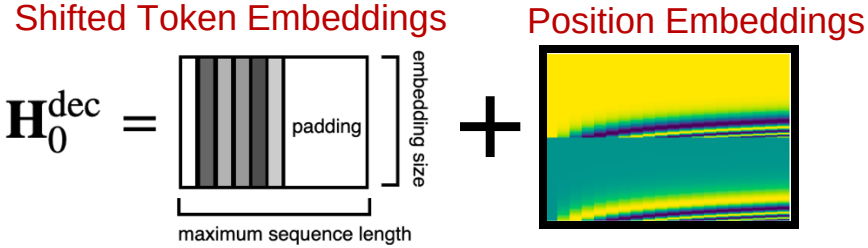


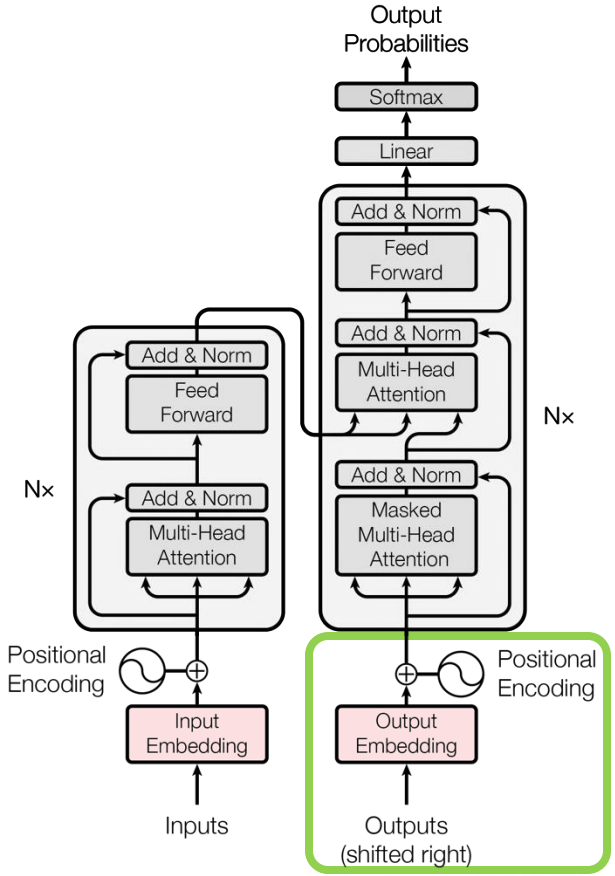
Transformers: “Attention is All You Need”

The input into the encoder looks like:



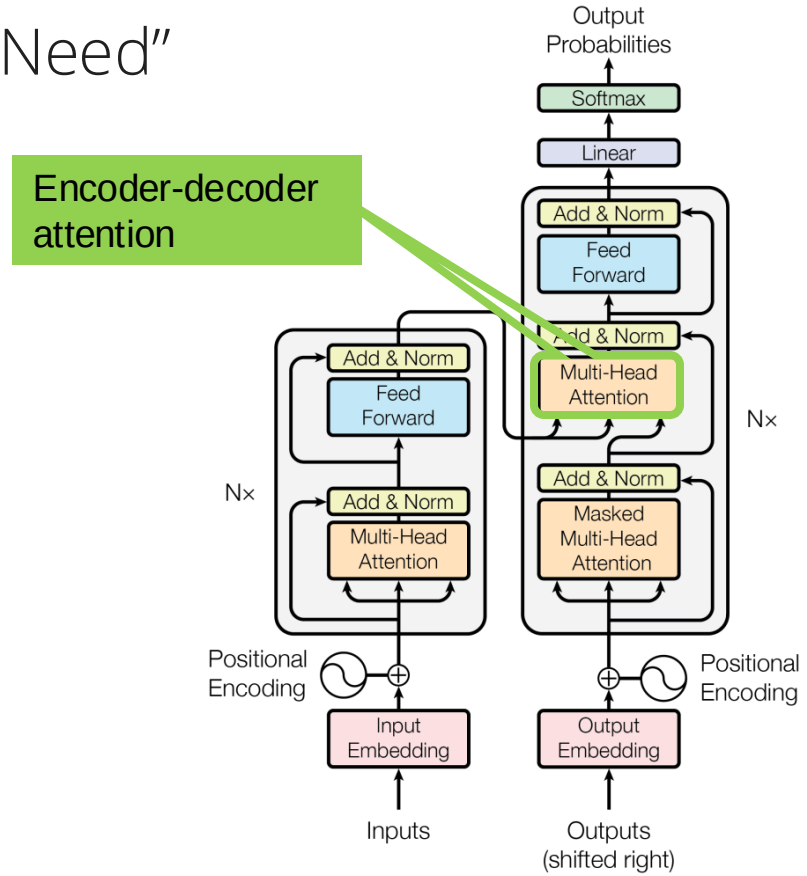
The input to the decoder looks like:



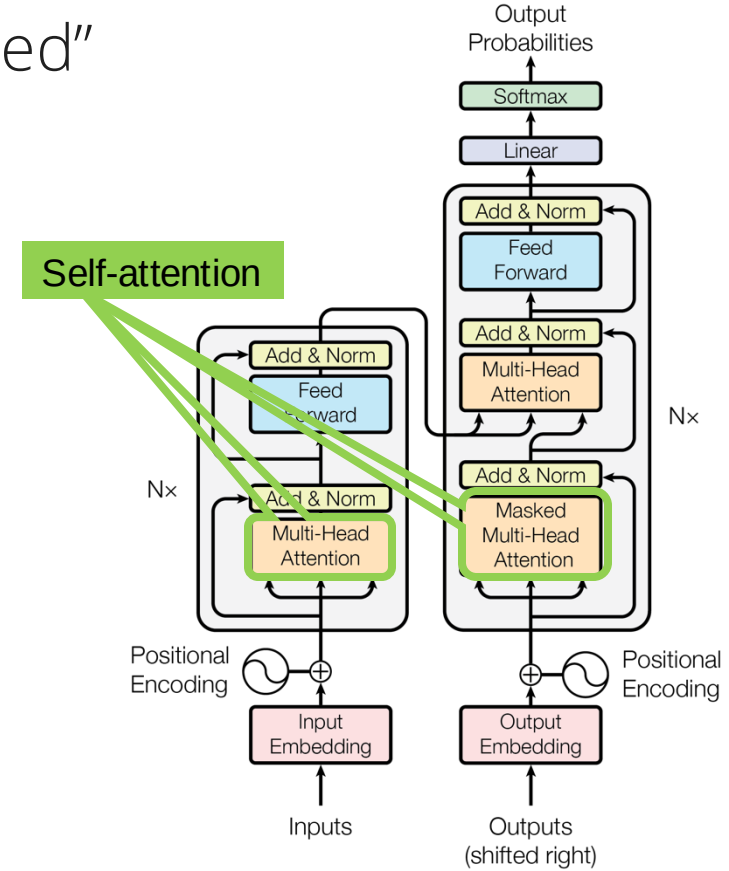


Transformers: “Attention is All You Need”

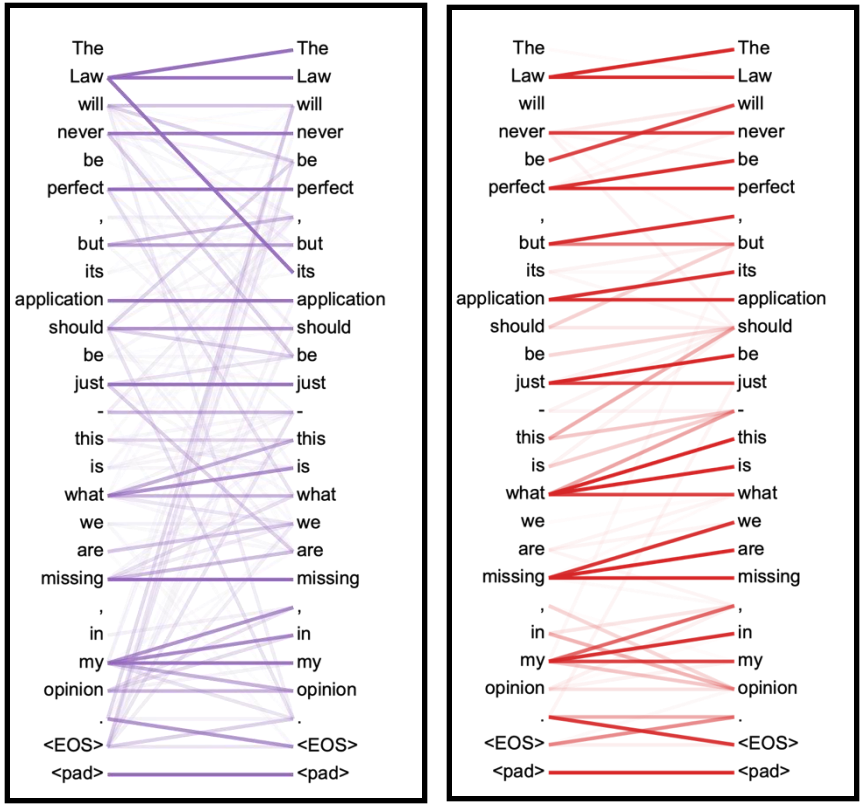
This is almost exactly the same as what the old recurrent seq2seq models had.

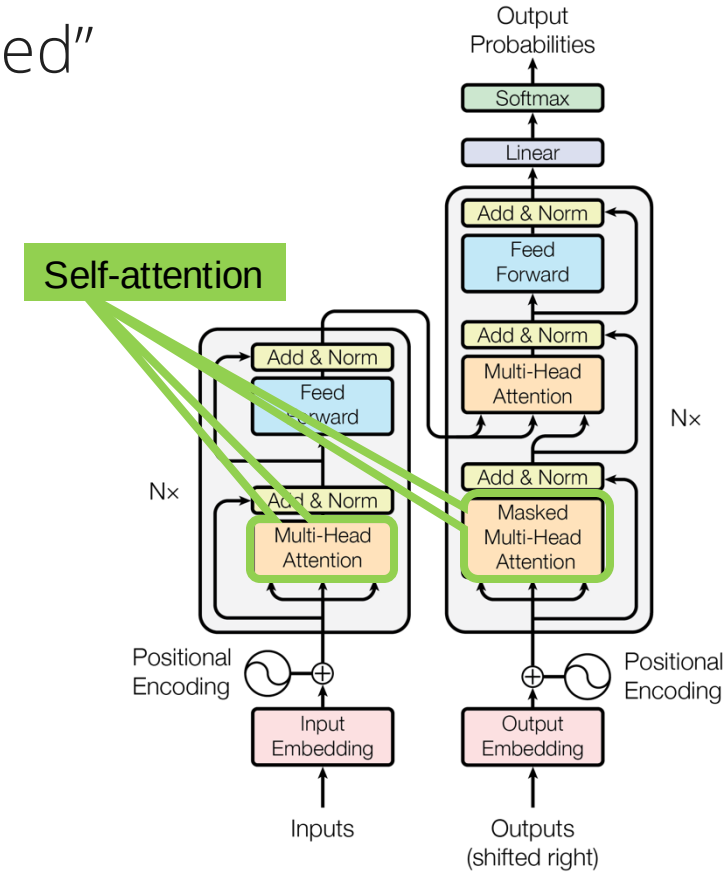


Transformers: “Attention is All You Need”

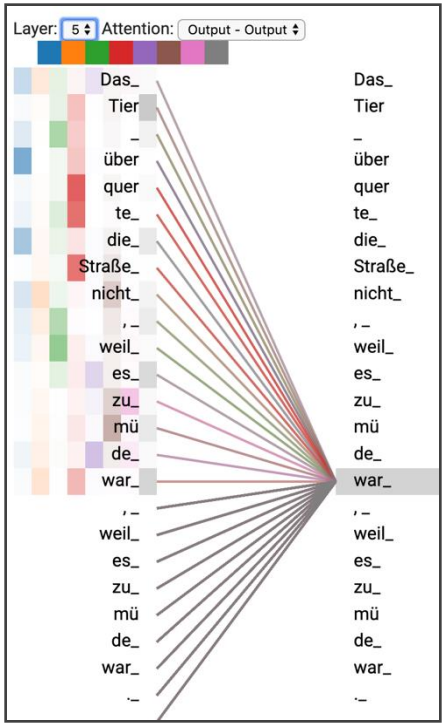


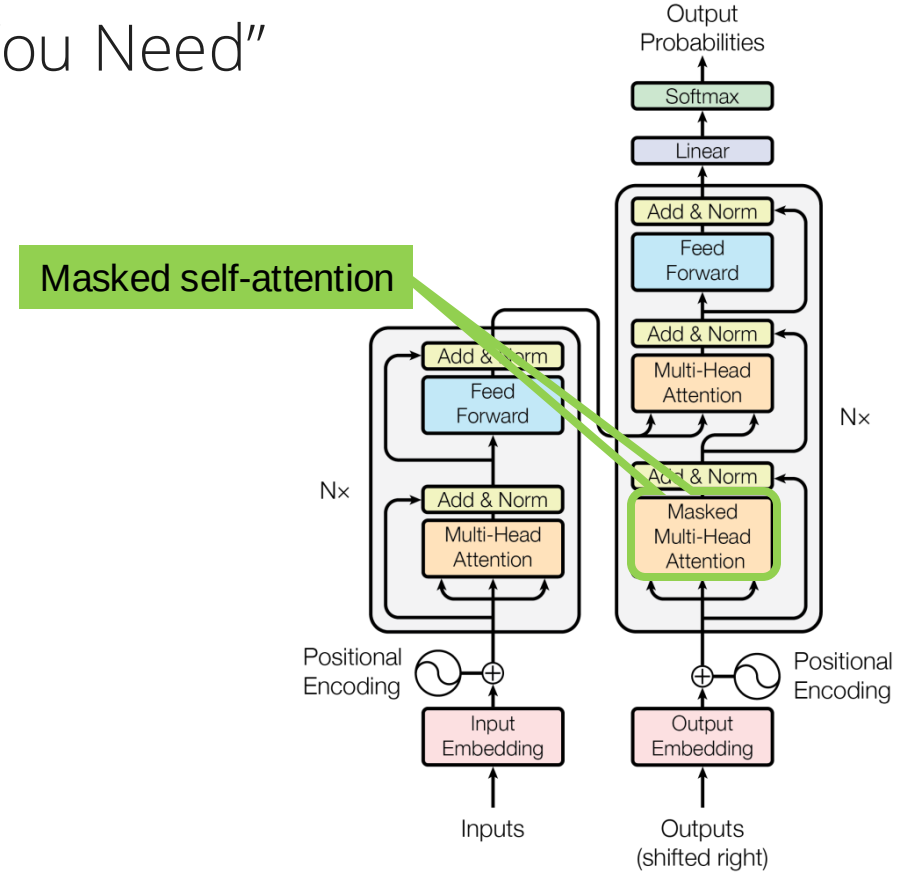
Transformers: “Attention is All You Need”



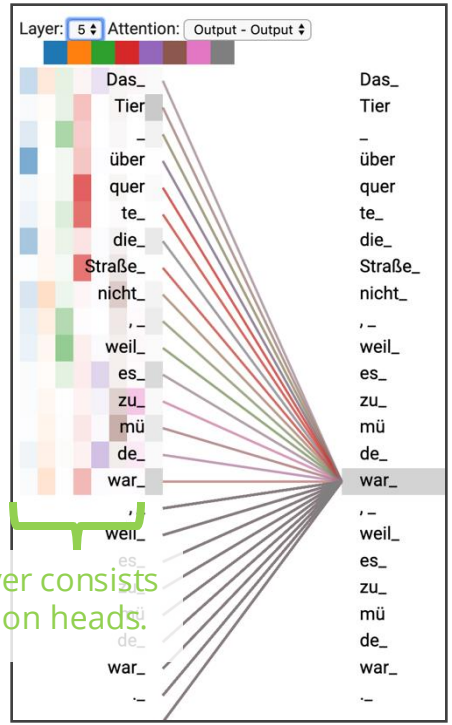


Transformers: “Attention is All You Need”

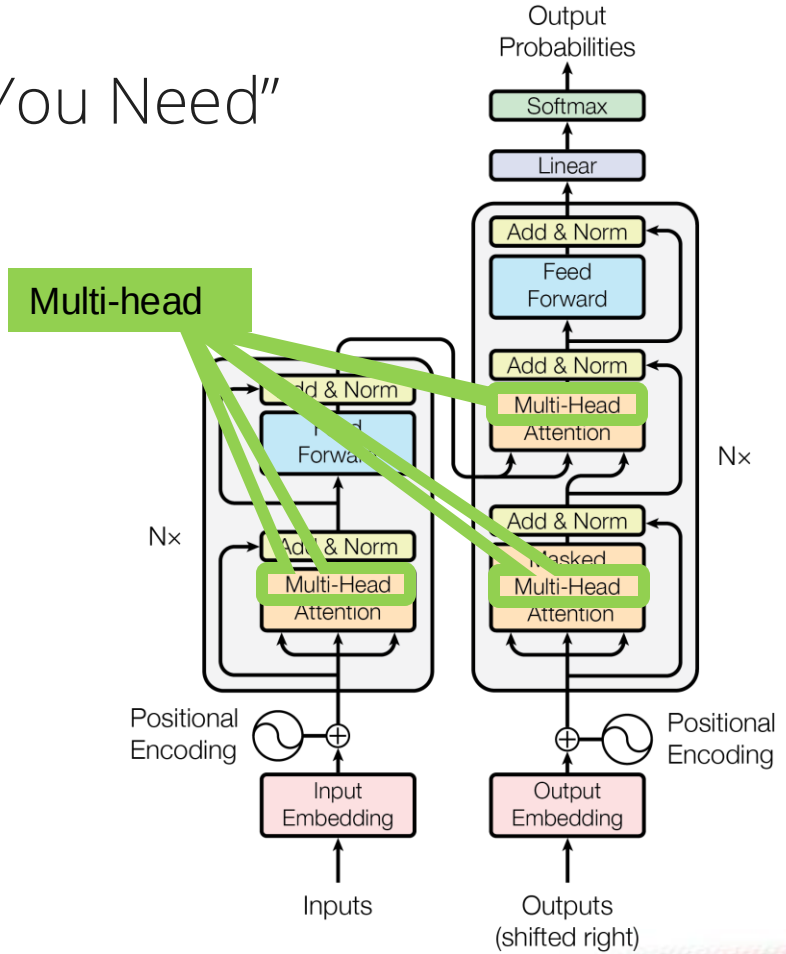




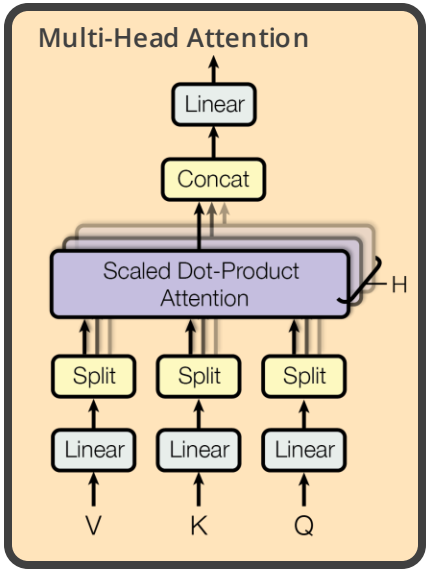
Transformers: “Attention is All You Need”



Each attention layer consists of multiple attention heads.

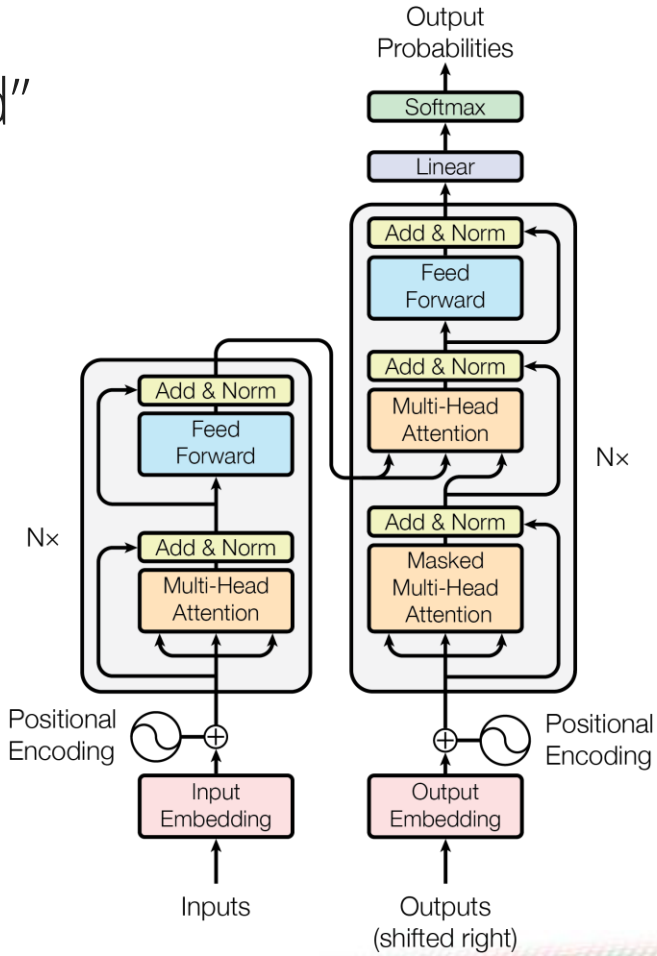


Transformers: “Attention is All You Need”

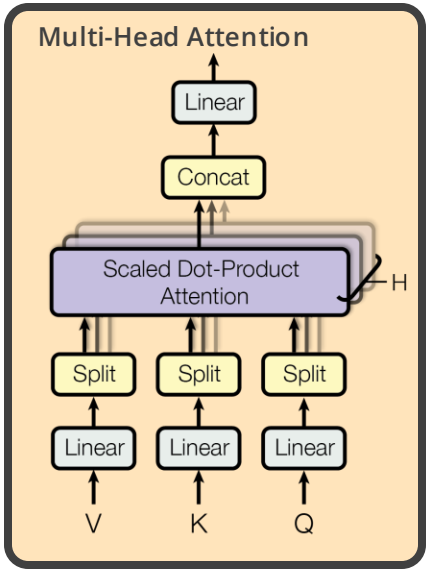


$$\text{head}_1 = \text{Attention}\left(\mathbf{Q}\mathbf{W}_1^Q, \mathbf{K}\mathbf{W}_1^K, \mathbf{V}\mathbf{W}_1^V\right)$$
$$\vdots$$
$$\text{head}_H = \text{Attention}\left(\mathbf{Q}\mathbf{W}_H^Q, \mathbf{K}\mathbf{W}_H^K, \mathbf{V}\mathbf{W}_H^V\right)$$

$$\text{MultiHeadAtt}(\mathbf{Q}, \mathbf{K}, \mathbf{V}) = \text{Concat}(\text{head}_1, \ldots, \text{head}_H)$$



Transformers: “Attention is All You Need”

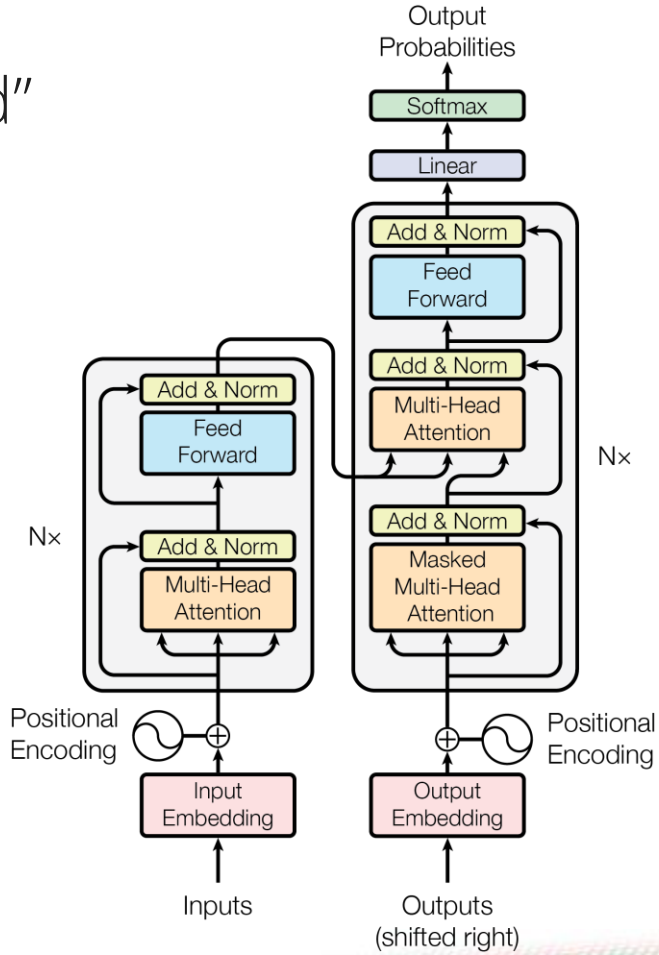


$$\text{head}_1 = \text{Attention}\left(\mathbf{Q}\mathbf{W}_1^Q, \mathbf{K}\mathbf{W}_1^K, \mathbf{V}\mathbf{W}_1^V\right)$$
$$\vdots$$
$$\text{head}_H = \text{Attention}\left(\mathbf{Q}\mathbf{W}_H^Q, \mathbf{K}\mathbf{W}_H^K, \mathbf{V}\mathbf{W}_H^V\right)$$

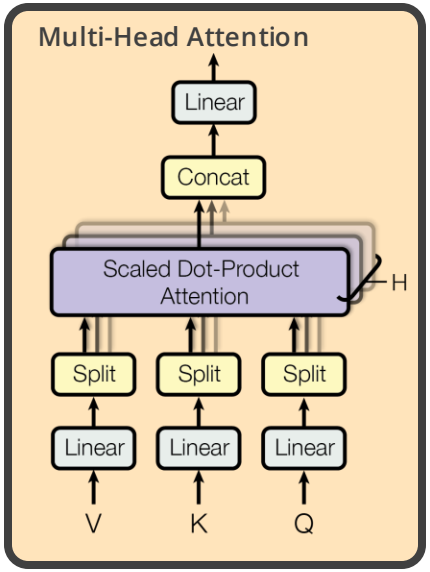
$$\text{MultiHeadAtt}(\mathbf{Q}, \mathbf{K}, \mathbf{V}) = \text{Concat}(\text{head}_1, \ldots, \text{head}_H)$$

Inputs and outputs of each layer are the same dimensions:

$$\mathbf{Q} \in \mathbb{R}^{T \times d_{\text{model}}}$$
$$\mathbf{K} \in \mathbb{R}^{T \times d_{\text{model}}}$$
$$\mathbf{V} \in \mathbb{R}^{T \times d_{\text{model}}}$$
$$\text{MultiHeadAtt}(\mathbf{Q}, \mathbf{K}, \mathbf{V}) \in \mathbb{R}^{T \times d_{\text{model}}}$$



Transformers: “Attention is All You Need”



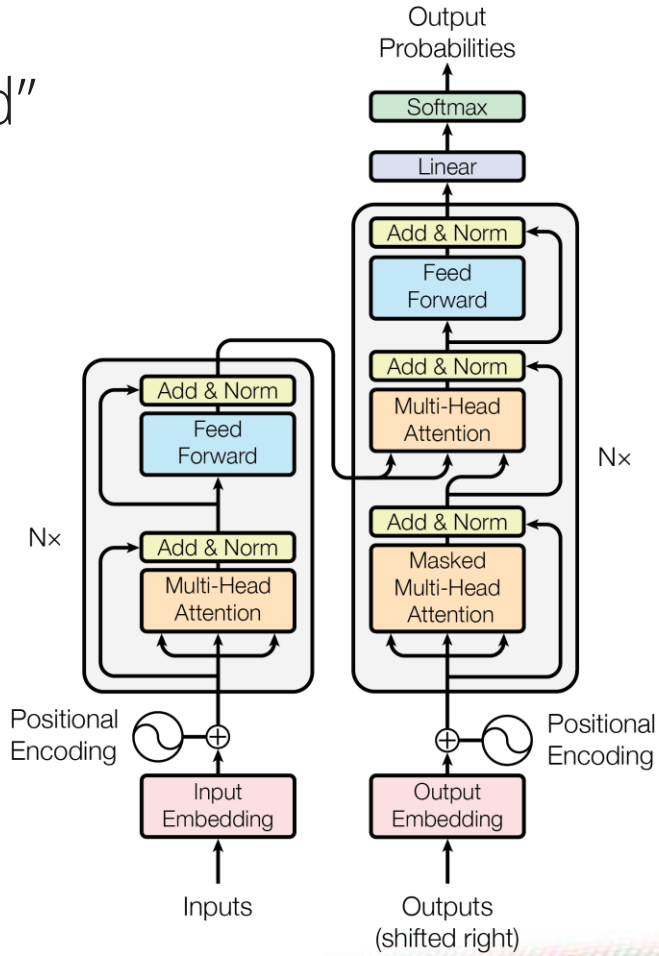
$$\text{head}_1 = \text{Attention}\left(\mathbf{Q}\mathbf{W}_1^Q, \mathbf{K}\mathbf{W}_1^K, \mathbf{V}\mathbf{W}_1^V\right)$$
$$\vdots$$
$$\text{head}_H = \text{Attention}\left(\mathbf{Q}\mathbf{W}_H^Q, \mathbf{K}\mathbf{W}_H^K, \mathbf{V}\mathbf{W}_H^V\right)$$

$$\text{MultiHeadAtt}(\mathbf{Q}, \mathbf{K}, \mathbf{V}) = \text{Concat}(\text{head}_1, \ldots, \text{head}_H)$$

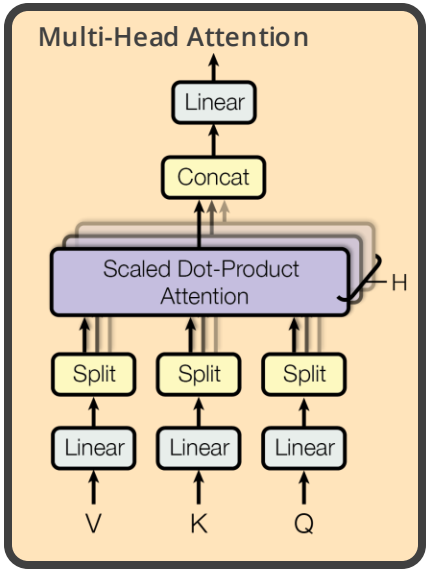
Inputs and outputs of each layer are the same dimensions:

$$\mathbf{Q} \in \mathbb{R}^{T \times d_{\text{model}}}$$
$$\mathbf{K} \in \mathbb{R}^{T \times d_{\text{model}}}$$
$$\mathbf{V} \in \mathbb{R}^{T \times d_{\text{model}}}$$
$$\text{MultiHeadAtt}(\mathbf{Q}, \mathbf{K}, \mathbf{V}) \in \mathbb{R}^{T \times d_{\text{model}}}$$

Concrete example:
$d_{\text{model}} = 512$ and $H = 8$.



Transformers: “Attention is All You Need”



$$\text{head}_1 = \text{Attention}(\mathbf{Q}\mathbf{W}_1^Q, \mathbf{K}\mathbf{W}_1^K, \mathbf{V}\mathbf{W}_1^V)$$
$$\vdots$$
$$\text{head}_H = \text{Attention}(\mathbf{Q}\mathbf{W}_H^Q, \mathbf{K}\mathbf{W}_H^K, \mathbf{V}\mathbf{W}_H^V)$$

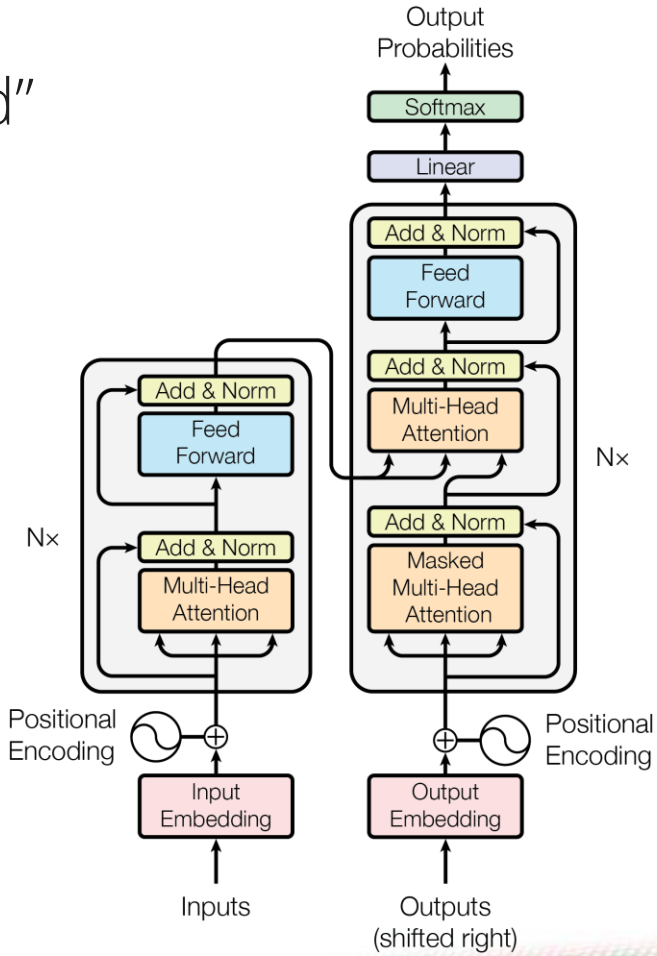
$$\text{MultiHeadAtt}(\mathbf{Q}, \mathbf{K}, \mathbf{V}) = \text{Concat}(\text{head}_1, \dots, \text{head}_H)$$

Inputs and outputs of each layer are the same dimensions:

$$\mathbf{Q} \in \mathbb{R}^{T \times d_{\text{model}}}$$
$$\mathbf{K} \in \mathbb{R}^{T \times d_{\text{model}}}$$
$$\mathbf{V} \in \mathbb{R}^{T \times d_{\text{model}}}$$
$$\text{MultiHeadAtt}(\mathbf{Q}, \mathbf{K}, \mathbf{V}) \in \mathbb{R}^{T \times d_{\text{model}}}$$

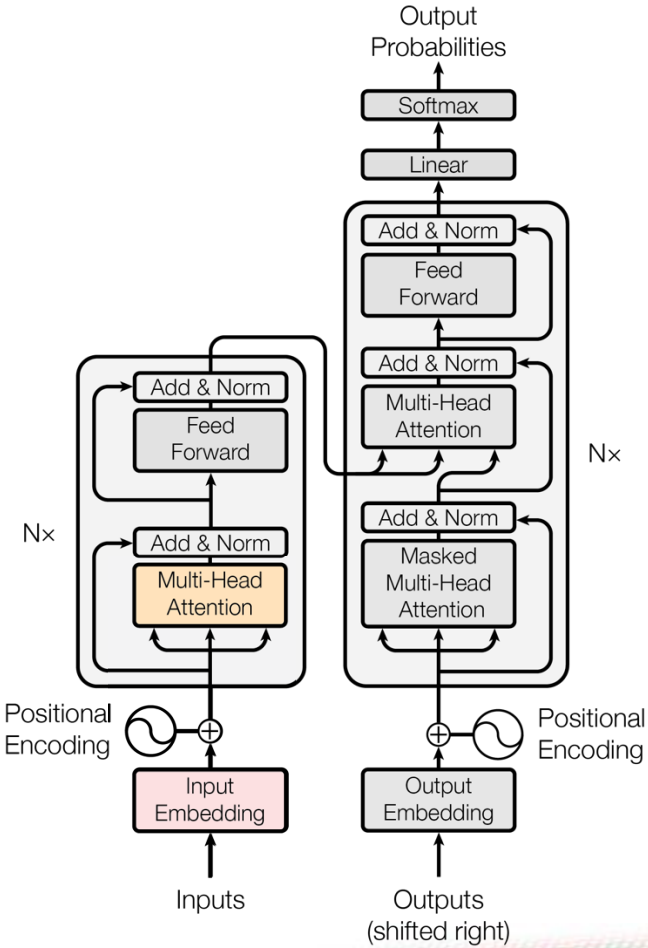
Concrete example:
$d_{\text{model}} = 512$ and $H = 8$.

This means: $\mathbf{W}_i^Q \in \mathbb{R}^{512 \times 64}$, $\mathbf{W}_i^K \in \mathbb{R}^{512 \times 64}$, $\mathbf{W}_i^V \in \mathbb{R}^{512 \times 64}$



The Encoder Step-by-Step

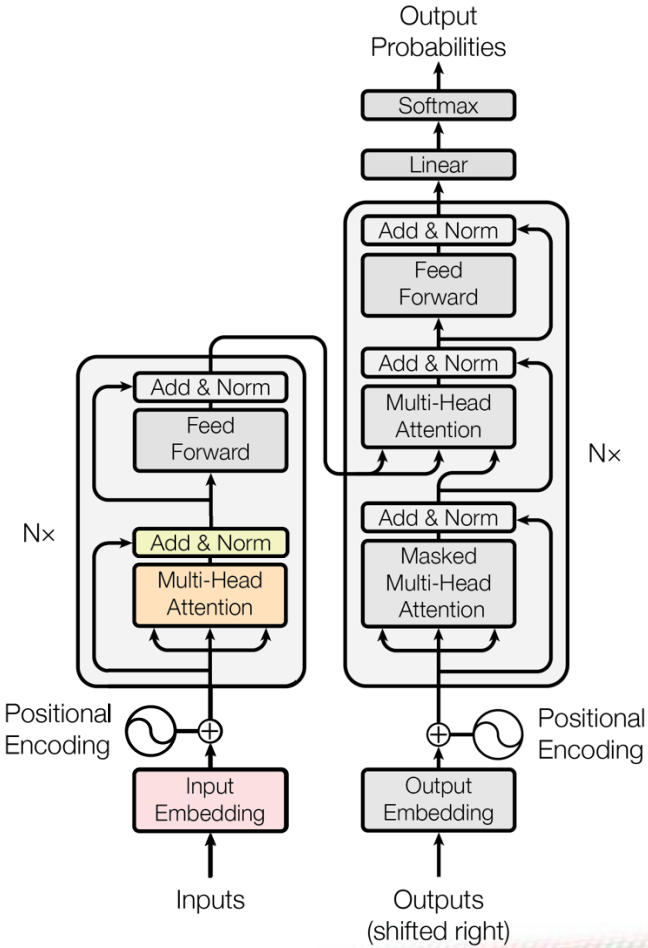
Multi-Head Attention = MultiHeadAtt($\mathbf{H}_i^{enc}$, $\mathbf{H}_i^{enc}$, $\mathbf{H}_i^{enc}$)



The Encoder Step-by-Step

Multi-Head Attention $= \text{MultiHeadAtt}(\mathbf{H}_i^{enc}, \mathbf{H}_i^{enc}, \mathbf{H}_i^{enc})$

Add & Norm $= \text{LayerNorm}($ Multi-Head Attention $+\mathbf{H}_i^{enc})$

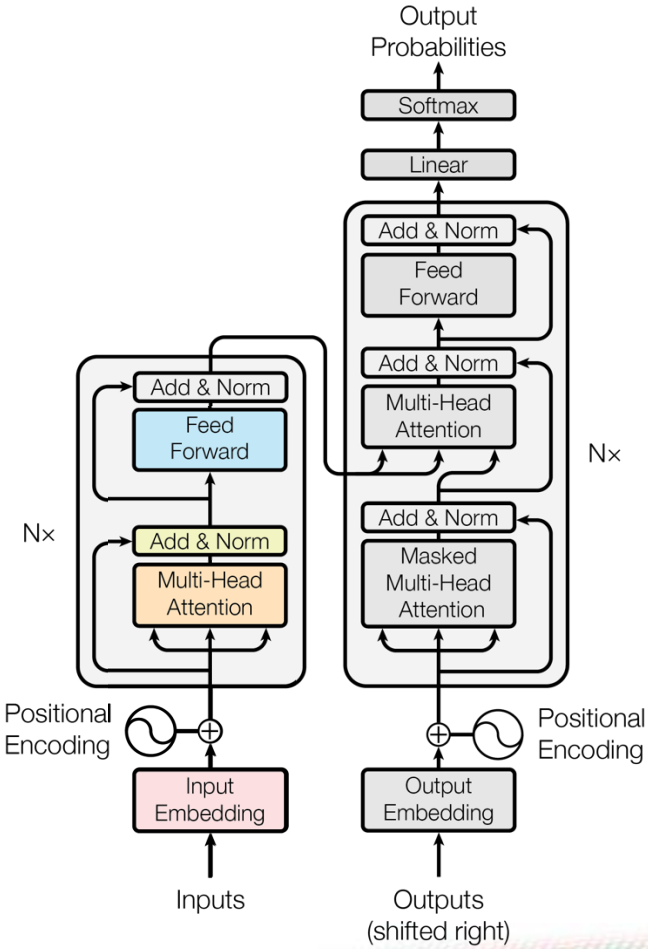


The Encoder Step-by-Step

Multi-Head Attention $= \text{MultiHeadAtt}(\mathbf{H}_i^{enc}, \mathbf{H}_i^{enc}, \mathbf{H}_i^{enc})$

Add & Norm $= \text{LayerNorm}($ Multi-Head Attention $+\mathbf{H}_i^{enc})$

Feed Forward $= \max(0,$ Add & Norm $\mathbf{W}_1 + b_1)\mathbf{W}_2 + b_2$



The Encoder Step-by-Step

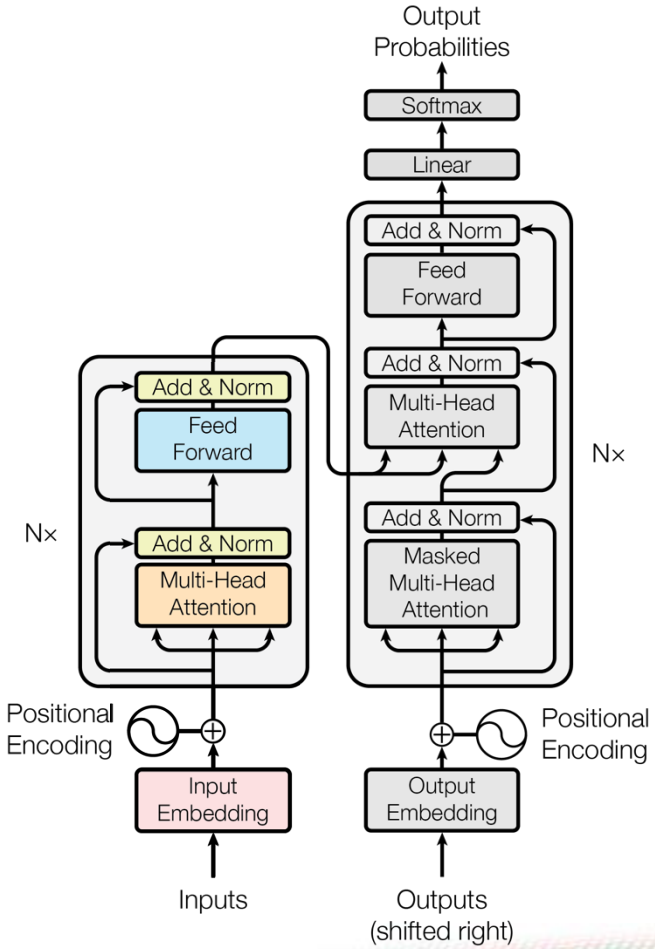
Multi-Head Attention $= \text{MultiHeadAtt}(\mathbf{H}_i^{enc}, \mathbf{H}_i^{enc}, \mathbf{H}_i^{enc})$

Add & Norm $= \text{LayerNorm}($ Multi-Head Attention $+\mathbf{H}_i^{enc})$

Feed Forward $= \max(0,$ Add & Norm $\mathbf{W}_1 + b_1)\mathbf{W}_2 + b_2$

Add & Norm (2) $= \text{LayerNorm}($ Feed Forward $+$ Add & Norm $)$

$\mathbf{H}_{i+1}^{enc} =$ Add & Norm(2)



The Decoder Step-by-Step

Masked Multi-Head Attention $= \text{MultiHeadAtt}(\mathbf{H}_i^{dec}, \mathbf{H}_i^{dec}, \mathbf{H}_i^{dec})$

Add & Norm $= \text{LayerNorm}($ Masked Multi-Head Attention $+\mathbf{H}_i^{dec})$

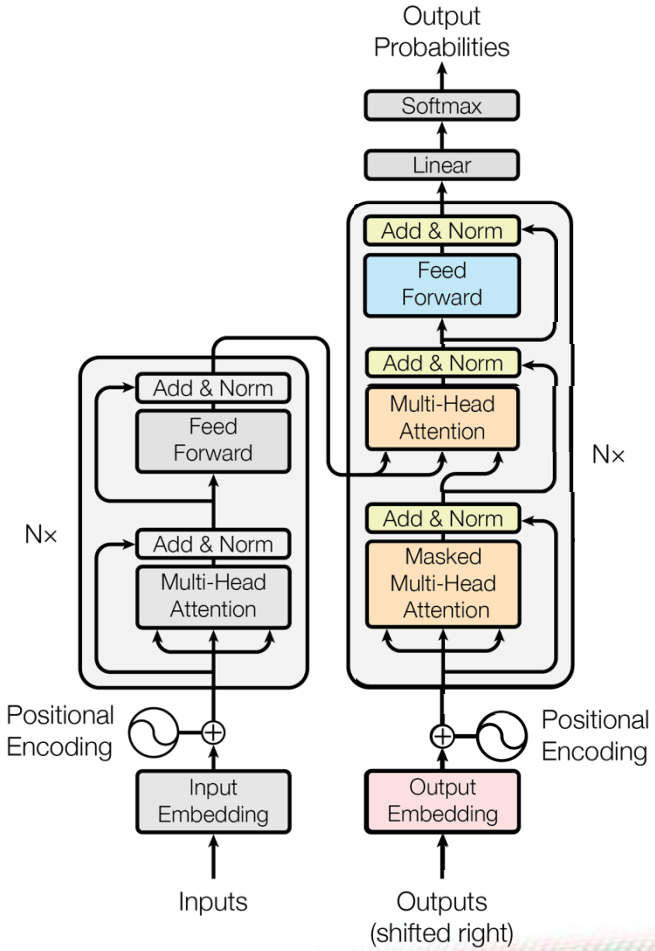
Enc-Dec Multi-Head Attention $= \text{MultiHeadAtt}(\mathbf{H}_i^{enc}, \mathbf{H}_i^{enc},$ Add & Norm $)$

Add & Norm (2) $= \text{LayerNorm}($ Enc-Dec Multi-Head Attention $+$ Add & Norm $)$

Feed Forward $= \max(0,$ Add & Norm (2) $\mathbf{W}_1 + b_1)\mathbf{W}_2 + b_2$

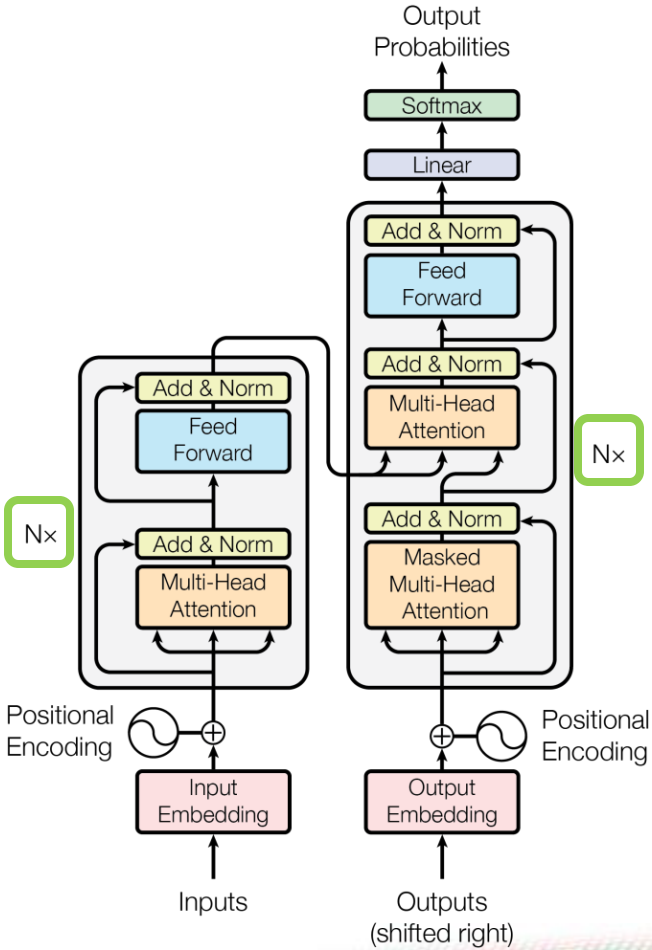
Add & Norm (3) $= \text{LayerNorm}($ Feed Forward $+$ Add & Norm (2) $)$

$\mathbf{H}_{i+1}^{enc} =$ Add & Norm(3)



Multiple Layers

In practice, there are many attention layers.



Transformers Generated Text Circa 2018

'''The Transformer''' are a Japanese [[hardcore punk]] band.

==Early years==
The band was formed in 1968, during the height of Japanese music history. Among the legendary [[Japanese people|Japanese]] composers of [Japanese lyrics], they prominently exemplified Motohiro Oda's especially tasty lyrics and psychedelic intention. Michio was a longtime member of the every Sunday night band PSM. His alluring was of such importance as being the man who ignored the already successful image and that he municipal makeup whose parents were – the band was called Jenei.<ref>http://www.separatist.org/se_frontend/post-punk-musician-the-kidney.html</ref> From a young age the band was very close, thus opting to pioneer what had actually begun as a more manageable core hardcore punk band.<ref>http://www.talkradio.net/article/independent-music-fades-from-the-closed-drawings-out</ref>

==History==
===Born from the heavy metal revolution===
In 1977 the self-proclaimed King of Tesponsors, [[Joe Lus:

: It was somewhere... it was just a guile ... taking this song to Broadway. It was the first record I ever heard on A.M., After some opposition I received at the hands of Parsons, and in the follow-up notes myself.<ref>http://www.discogs.com/artist/The+Op%C5%8Dn+&+Psalm</ref>

The band cut their first record album titled "Transformed, furthered and extended Extended",<ref>[https://www.discogs.com/album/69771 MC – Transformed EP (CDR) by The Moondrawn – EMI, 1994]</ref> and in 1978 the official band line-up of the three-piece pop-punk-rock band TEEM. They generally played around [[Japan]], growing from the Top 40 standard.

===1981-2010: The band to break away===
On 1 January 1981 bassist Michio Kono, and the members of the original line-up emerged. Niji Fukune and his [[Head poet|Head]] band (now guitarist) Kazuya Kouda left the band in the hands of the band at the May 28, 1981, benefit season of [[Led Zeppelin]]'s Marmarin building. In June 1987, Kono joined the band as a full-time drummer, playing a few nights in a 4 or 5 hour stint with [[D-beat]]. Kono played through the mid-1950s, at Shinlie, continued to play concerts with drummers in Ibis, Cor, and a few at the Leo Somu Studio in Japan. In 1987, Kono recruited new bassist Michio Kono and drummer Ayaka Kurobe as drummer for band. Kono played trumpet with supplement music with Saint Etienne as a drummer. Over the next few years Kono played as drummer and would get many alumni news invitations to the bands' "Toys Beach" section. In 1999 he joined the [[CT-182]].

His successor was Barrie Bell on a cover of [[Jethro Tull (band)|Jethro Tull]]'s original 1967 hit "Back Home" (last appearance was in Jethro), with whom he shares a name.

Transformers Generated Text Circa 2018

===2010 – present: The band to split===
In 2006 the band split up and the remaining members reformed under the name Starmirror, with Kono in tears, Kurobe, and Kurobe all playing harmonica with Kooky Bell and a new guitarist again. While Jaari also had the realist DJ experience, The SkykelDaten asked New Bantherhine, who liked Kono and Kurobe, to join him on guitar. Kono is now playing in the studio a new formation, and at their 11th anniversary concert, made a wide variety of music and DJ equipment including two new vocalists: Dejeh Faida and Janis. NARCO (Inc.) privileges areas sections until 2012. and in 2015 both were members as members as were Dicent and Cautty.

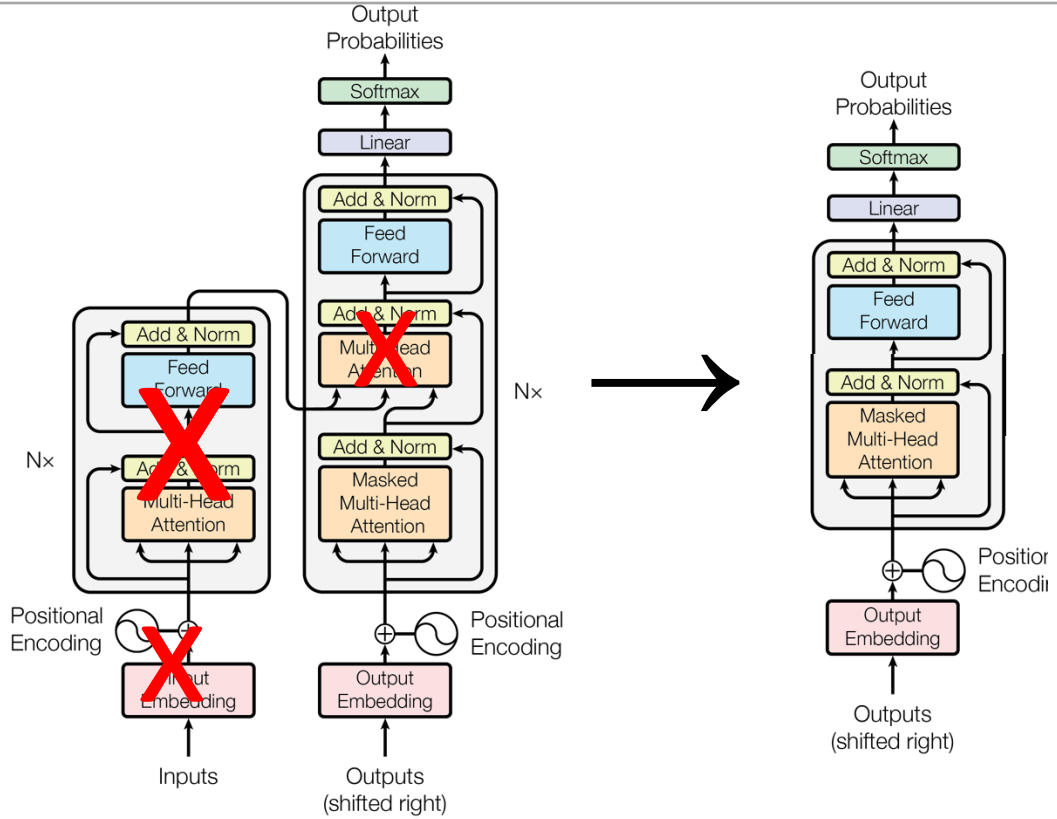
In 2014, the album "[[Marco Victoriano in Focus]]" was released, and entered the Japanese albums chart at number 69 in the [[Oricon Singles Chart]].<ref name="oricon">{{cite web|title=THE GER: THE TALENT RAILWAY LIVERSITIES|url=http://www.oricon.co.jp/prof/inductee/30565/ranking/cd/1/|work=OriconStyle|accessdate=11 Jul 2014}}</ref> The album was also in Japan, where [[Hoite (musician)|Hoite]] recorded and released an album in October 2011, and a short album, titled "[[Grateful]]" in 2012.<ref>{{cite web|url=http://www.oricon.co.jp/prof/artist/229337/ranking/cd/1/|title=HORIZON HISTORY|publisher=oricon.co.jp|accessdate=9 Nov 2014|language=ja}}</ref> Kono played the [[N9ne]] bass with Tony "Shadows Without a Face", and released his music from the new [[Smile (record label)|Smile]] label with original Fat Joe Lang "Remix and Bachian" cassette.

==Style==
The band's style has been compared to [[Radar|radar-based]], influenced by bands such as [[Metallica]], [[Damage (Japanese band)|Damage]], [[Dreadzone]], and [[Girlschool]].

The group classifies itself as "the first band to play a Used Of American Inside the Outer East".<ref>http://www.funonline.jp/i/main.php?conID=005&artID=12548</ref> including the band's usual Eugene Terre ensemble. He later stated that the raw interviews in Metal Hammer and Metallica gave reason to what they considered that an explosion of the live band started by the band.<ref>http://stillenterprise.com/en/interviews/last-night-reunion-confronts-kooky-spearing-the-charismatic-dynamic-partnership/</ref>

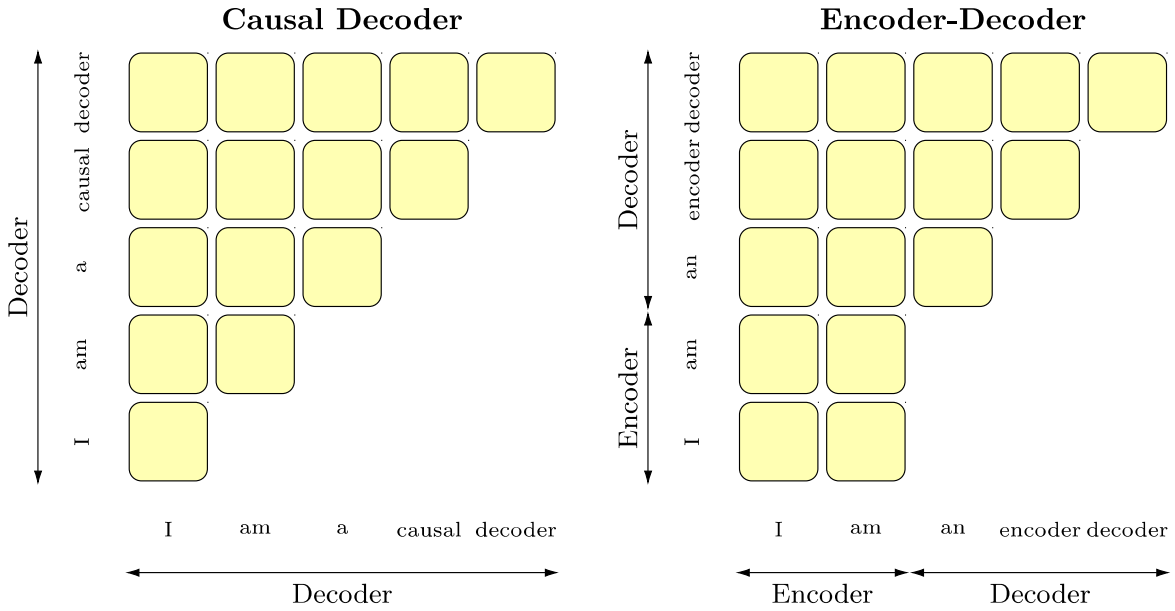
[[Hunt's Brigade]], a surf-rock band from Tokyo, has cited the band's music as being "straight ahead of their time / I did but only read five songs (played now), a beat, rock blast, a bit of a bass blast, good lyrics, and a few".<ref>http://stillenterprise.com/en/announcements/Package-a2/Swag-w-Vause-042332/</ref> Halutodrinking, a popular-sounding drum style, has described the band as being "supporting [mod]ed distrust", because it incorporated the track as an ensemble and did not fit into any of the more darling songs from their previous incarnation, which was forging a strong style to a lot of different

Turning this into a decoder-only architecture



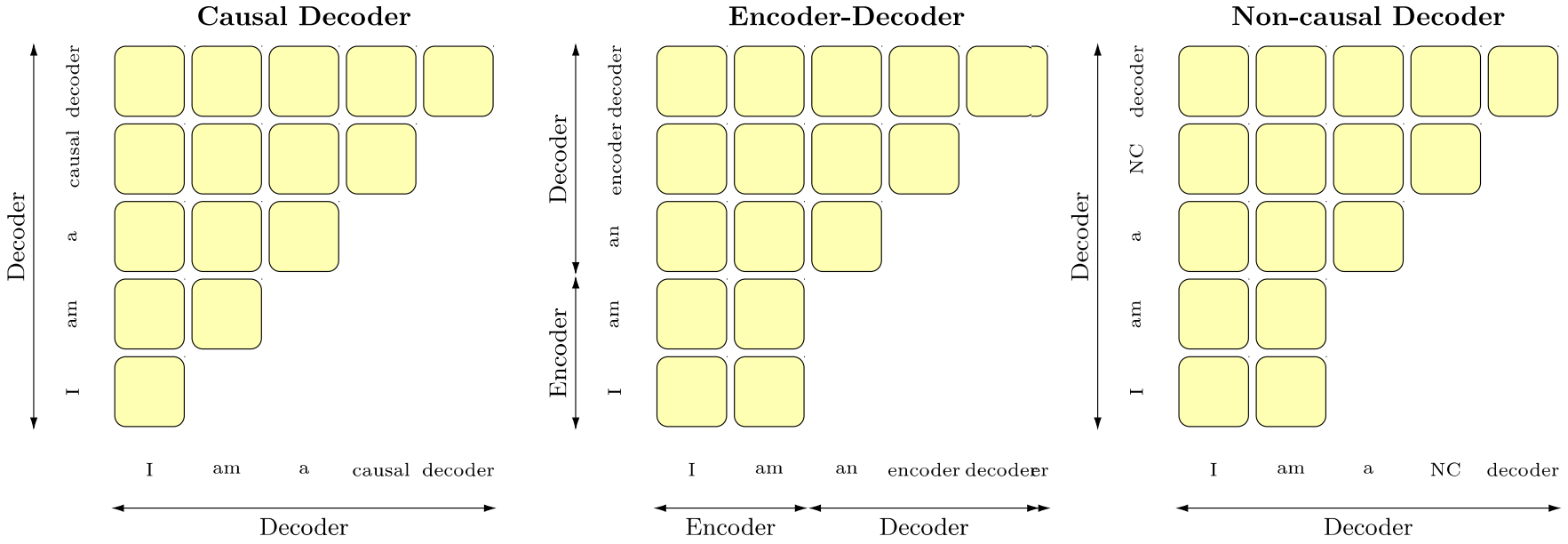
Attention in different Transformer architectures.

Yellow boxes indicate positions that are allowed to attend to each other.



Attention in different Transformer architectures.

Yellow boxes indicate positions that are allowed to attend to each other.



Carnegie Mellon University

Learning Objectives

Large Language Models: Methods and Applications

Daphne Ippolito and Chenyan Xiong

What I’ve told you so far:

Language models are trained with the objective of predicting the next word in a sequence given the previous words (and possibly some other conditioning signal).

What I've told you so far:

Language models are trained with the objective of predicting the next word in a sequence given the previous words (and possibly some other conditioning signal).

We can change the learning objective by changing up these sequences.

What do we want from our learning objective.

The goals of **pre-training** (the first stage of training) are to get the language model to:

- learn the structure of natural language
- learn humans' understanding of the world (as encoded in the training data).

We want a learning objective that facilitates these goals.

Possible objectives for pre-training an encoder-decoder model:

- Predict a suffix given a prefix.
  - Input: **I took my dog, Fido, to the**
  - Target: **park for his walk.**
- Masked language modeling
  - Input: **I took <x> to <y> his walk.**
  - Target: **<x> my dog, Fido, <y> the park for**