# Class core values

1. Be **respect**ful to yourself and others
2. Be **confident** and believe in yourself
3. Always do your **best**
4. Be **cooperative**
5. Be **creative**
6. Have **fun**
7. Be **patient** with yourself while you learn
8. Don’t be shy to **ask “stupid” questions**
9. Be **inclusive** and **accepting**

Week 6, Lecture 1

# Learning on sequences: Transformers

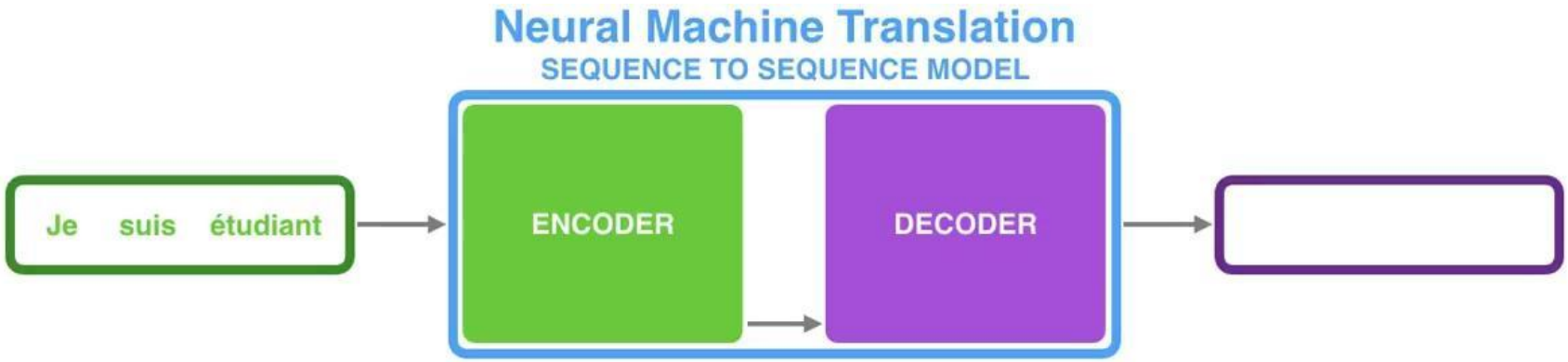
# Learning Objectives

1. Describe the basic concept of a transformer unit
2. Explain positional embedding and its application
3. Explain attention matrix and the difference between its variant
4. Describe decoding and encoding
5. Explain the inferring process in transformers

# Transformers were first developed for Seq2Seq tasks

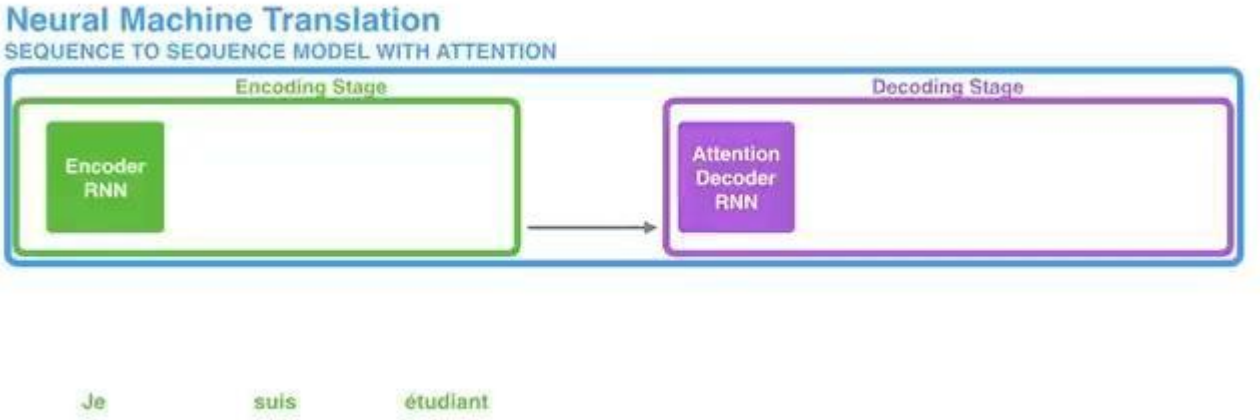

Video from Jay Alammar blog: Seq2seq models with attention

# Seq2Seq models use a simple model of encoding and decoding




Video from Jay Alammar blog: Seq2seq models with attention

# Attention helps seq2seq models to remember the context




Video from Jay Alammar blog: Seq2seq models with attention

# Attention is all you need!

## 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

# Attention is all you need!

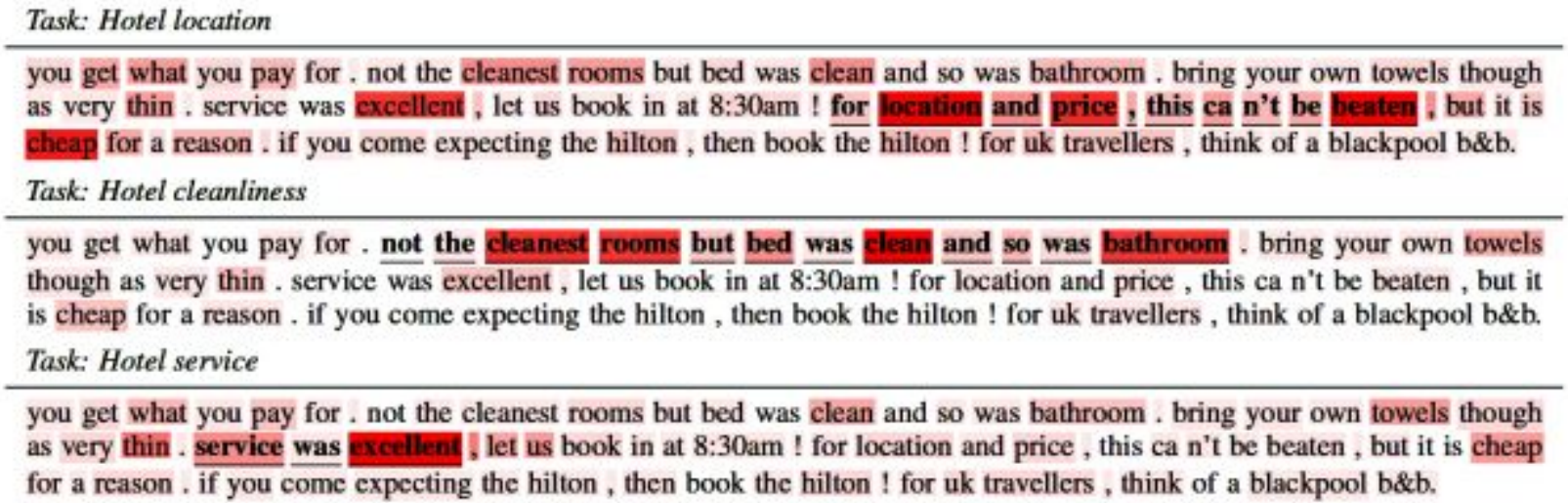



Image from Glassi et al, *IEEE* **2021** 32(10)

# What it can do …

https://transformer.huggingface.co/doc/distil-gpt2

# Basic architecture of a transformer

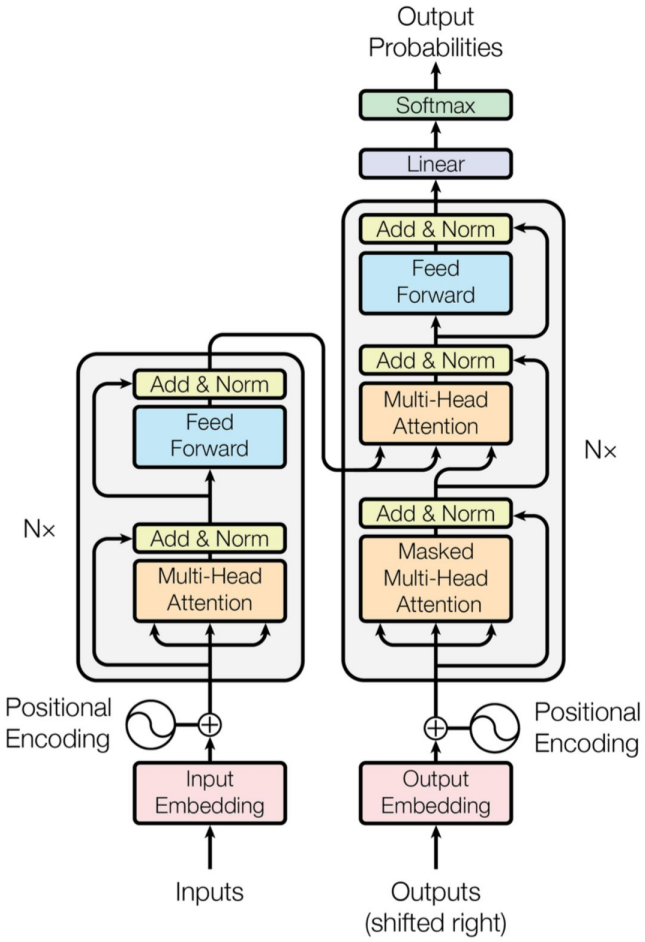
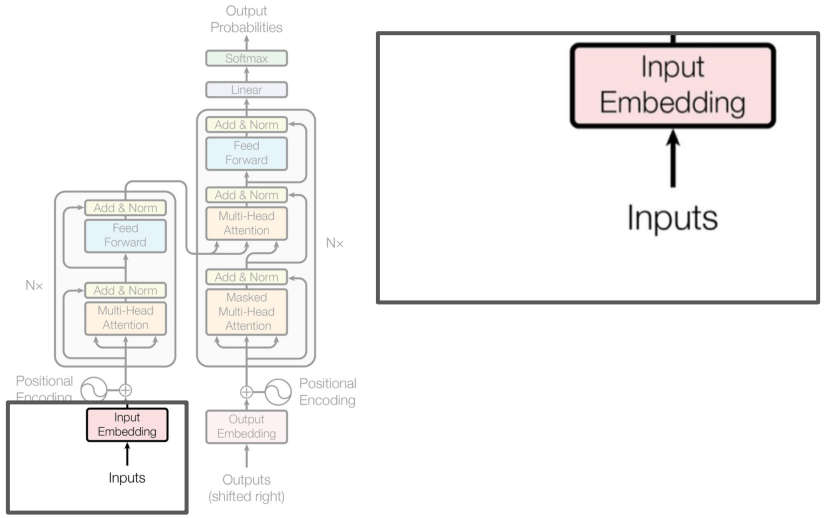



Image from medium: What is a transformer

# The second step is word embedding







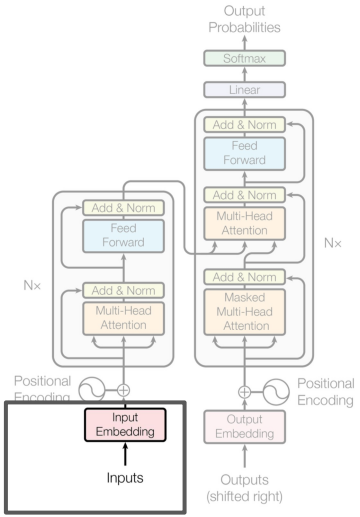
original text "hello world!"

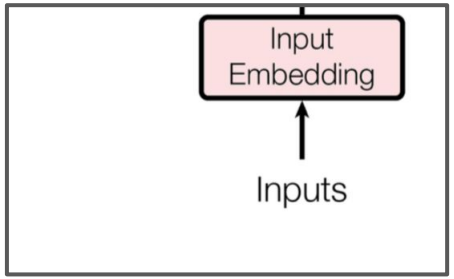
tokens ['hello', 'world', '!']

token IDs [7592, 2088, 999]

Image from TDS, James Briggs: Why are there so many tokenization methods for transformers

# The first step of the process is word embedding



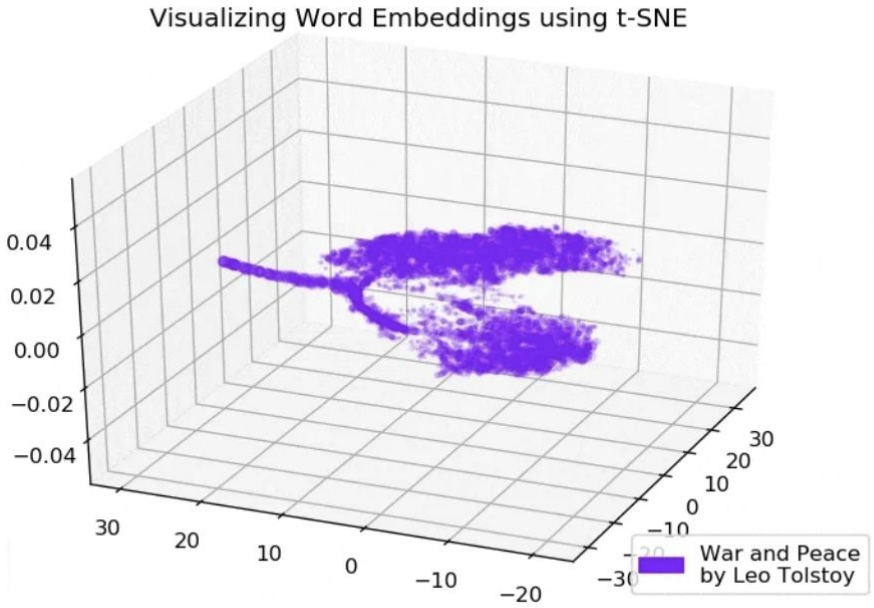
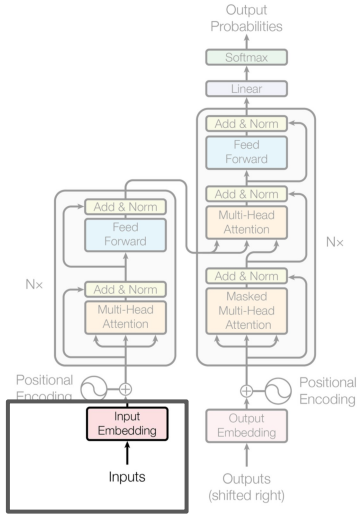
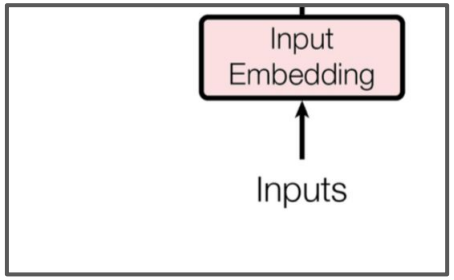




Image from AI Summer, Nicholas Adaloglu: Transformers

# During the tokenization and embedding, we lose the order of words

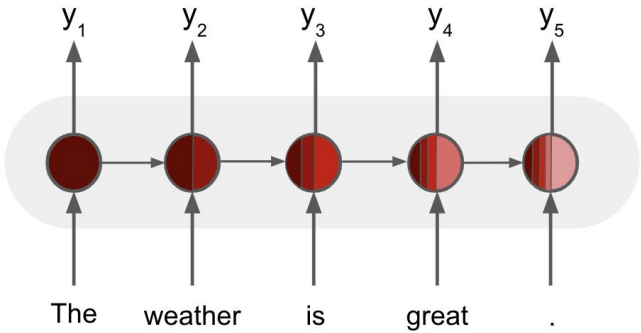


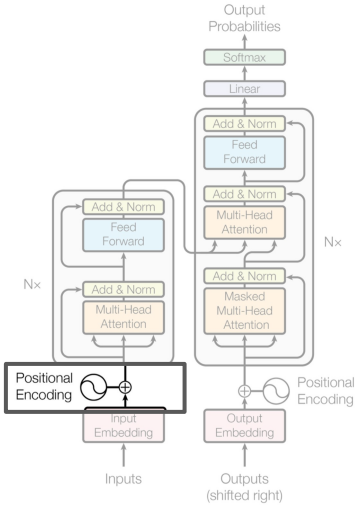

original text "hello world!"

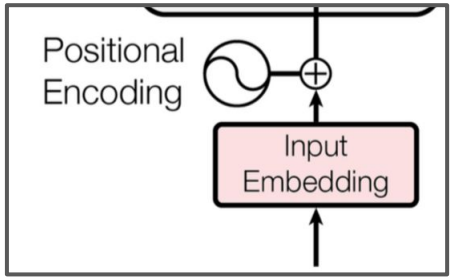
tokens ['hello', 'world', '!']

token IDs [7592, 2088, 999]

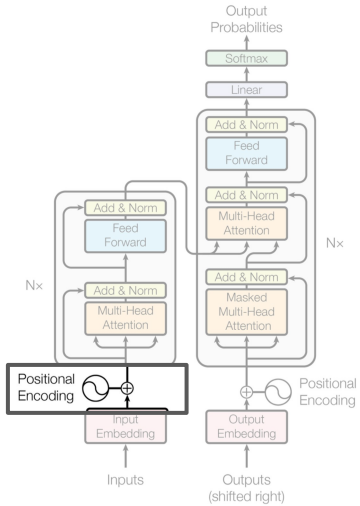


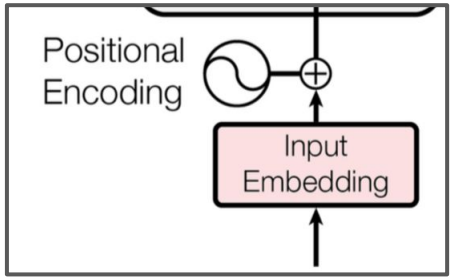
# Positional encoding allows for some notion of order

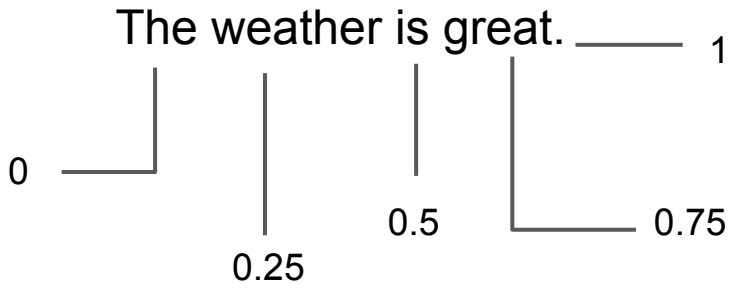


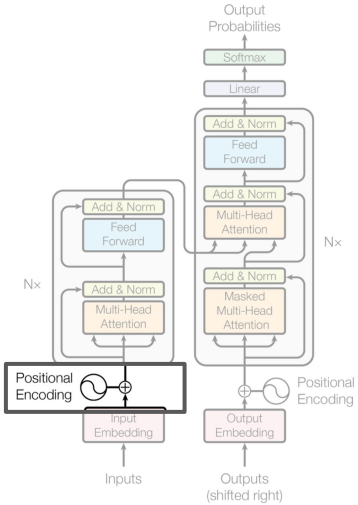
# Positional encoding allows for some notion of order

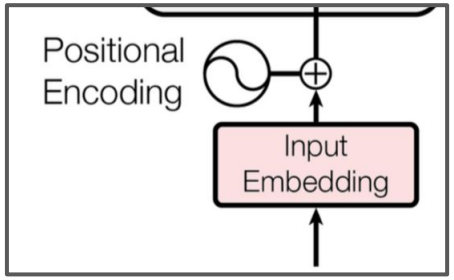


# A good positional encoding should have consistent distance




The weather is great.

0, 0.25, 0.5, 0.75, 1

The weather is really great!

0, 0.2, 0.4, 0.6, 0.8, 1

# A good positional encoding should have

1. Unique stamps for each position in the sentence
2. Distances should be consistent
3. A way to easily generalize to larger sentences
4. Deterministic assignment

# Sinusoidal functions are good embeddings

$$\overrightarrow{p_t}^{(i)} = f(t)^{(i)} := \begin{cases} \sin(\omega_k . t), & \text{if } i = 2k \\ \cos(\omega_k . t), & \text{if } i = 2k+1 \end{cases}$$

where

$$\omega_k = \frac{1}{10000^{2k/d}}$$

Image from Amirhossein Kazeminejad's blog: Transformer architecture, the positional embedding

# Sinusoidal functions are good embeddings

$$\overrightarrow{p_t}^{(i)} = f(t)^{(i)} := \begin{cases} \sin(\omega_k . t), & \text{if } i = 2k \\ \cos(\omega_k . t), & \text{if } i = 2k+1 \end{cases}$$

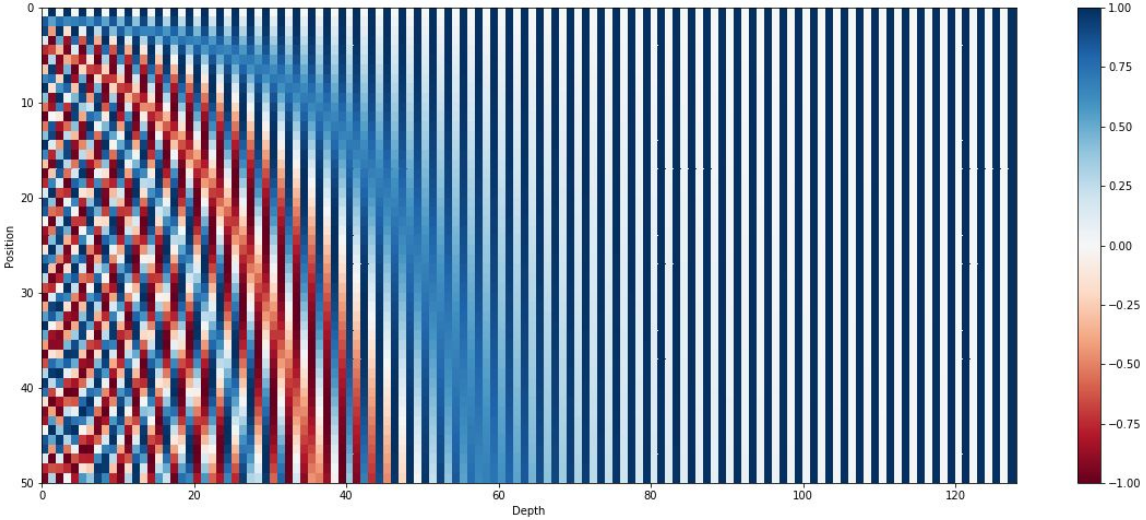
where

$$\omega_k = \frac{1}{10000^{2k/d}}$$

$$\overrightarrow{p_t} = \begin{bmatrix} \sin(\omega_1 . t) \\ \cos(\omega_1 . t) \\ \\ \sin(\omega_2 . t) \\ \cos(\omega_2 . t) \\ \\ \vdots \\ \\ \sin(\omega_{d/2} . t) \\ \cos(\omega_{d/2} . t) \end{bmatrix}_{d \times 1}$$

Image from Amirhossein Kazeminejad's blog: Transformer architecture, the positional embedding

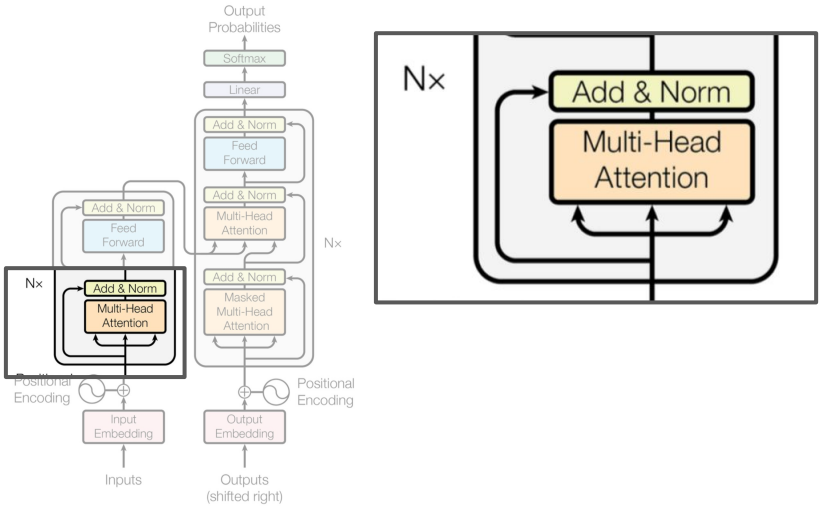
# Sinusoidal functions are good embeddings




$$\overrightarrow{p_t} = \begin{bmatrix} \sin(\omega_1 . t) \\ \cos(\omega_1 . t) \\ \\ \sin(\omega_2 . t) \\ \cos(\omega_2 . t) \\ \\ \vdots \\ \\ \sin(\omega_{d/2} . t) \\ \cos(\omega_{d/2} . t) \end{bmatrix}_{d \times 1}$$

Image from Amirhossein Kazeminejad's blog: Transformer architecture, the positional embedding

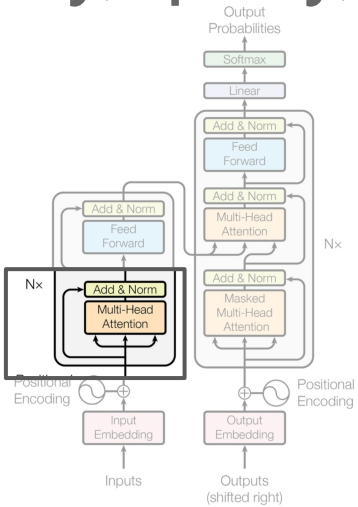
# The main module of a transformer is an *attention* unit

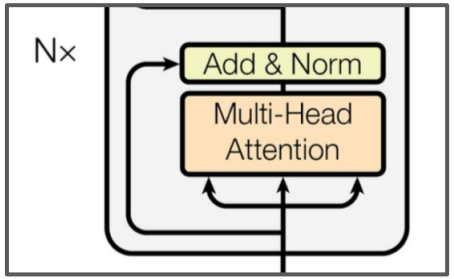


# Self-attention allows for finding correlation between words

| | the | weather | is | great |
|---|---|---|---|---|
| the | 0.8 | 0.1 | 0.05 | 0.05 |
| weather | 0.1 | 0.6 | 0.2 | 0.1 |
| is | 0.05 | 0.2 | 0.65 | 0.1 |
| great | 0.2 | 0.1 | 0.1 | 0.6 |

# To understand attention, we need to understand key, query, and value



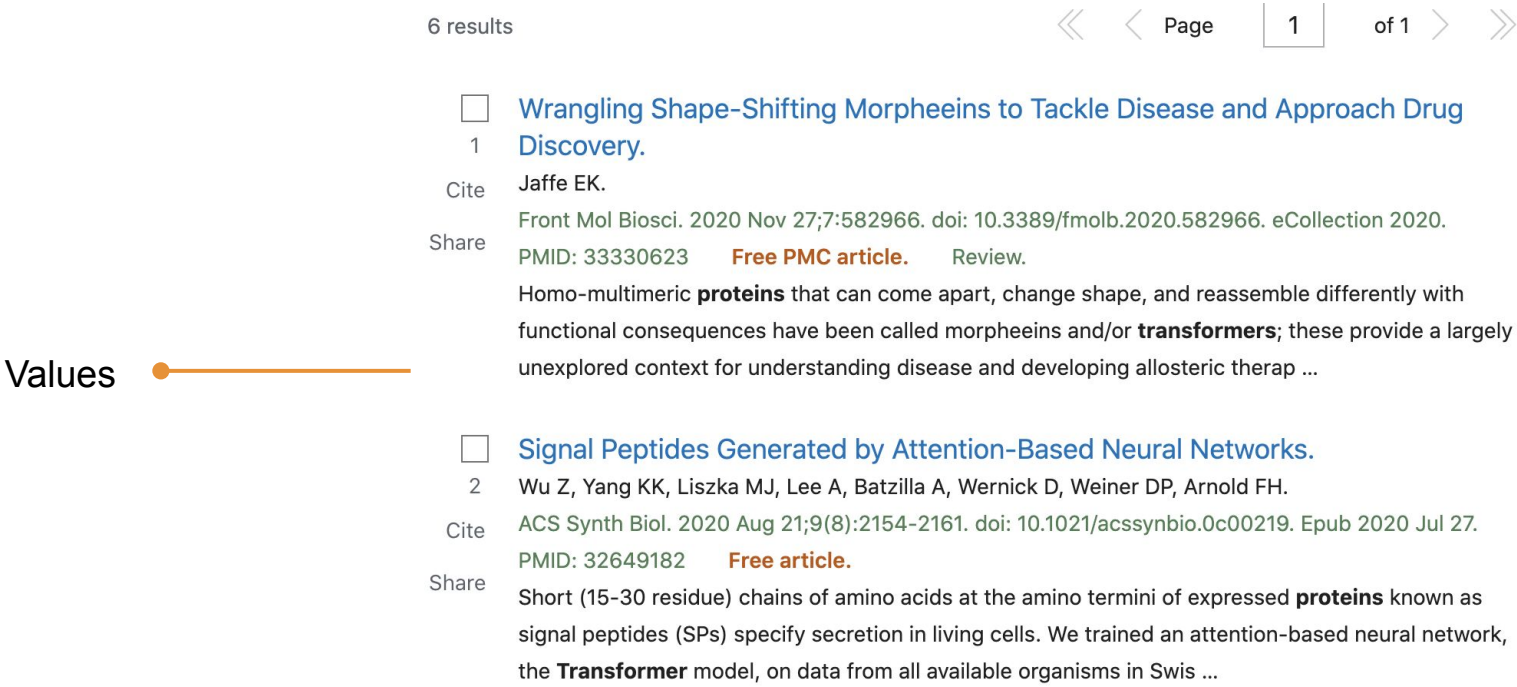



# Feature-based attentions

# Feature-based attentions

# Feature-based attentions

# Feature-based attentions



# To calculate attention, query, key, and value matrices are generated

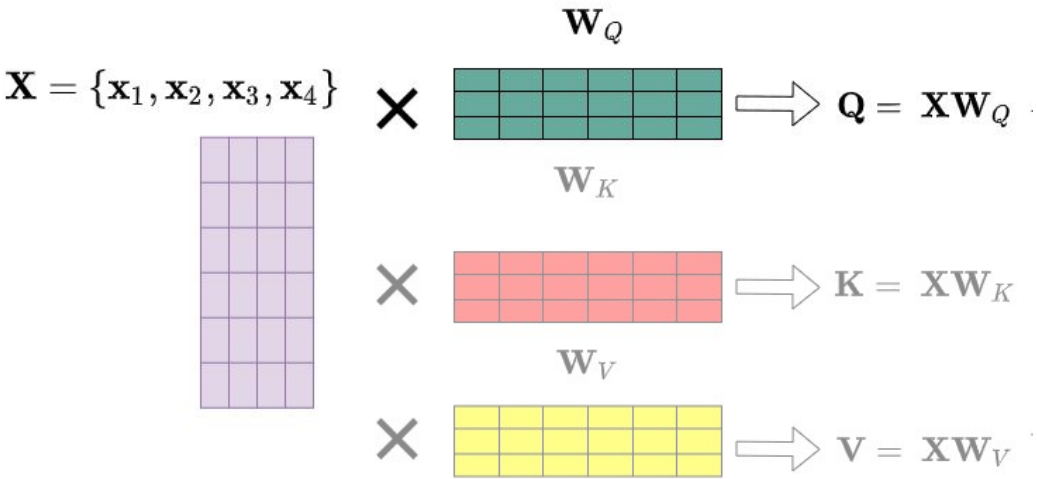



Image from AI Summer, Nicholas Adaloglu: Transformers

# To calculate attention, query, key, and value matrices are generated

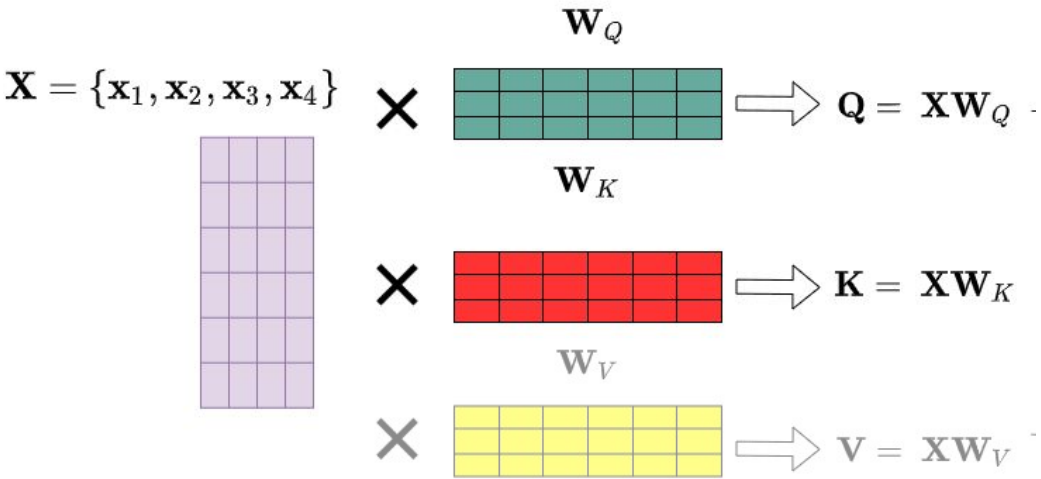



Image from AI Summer, Nicholas Adaloglu: Transformers

# To calculate attention, query, key, and value matrices are generated

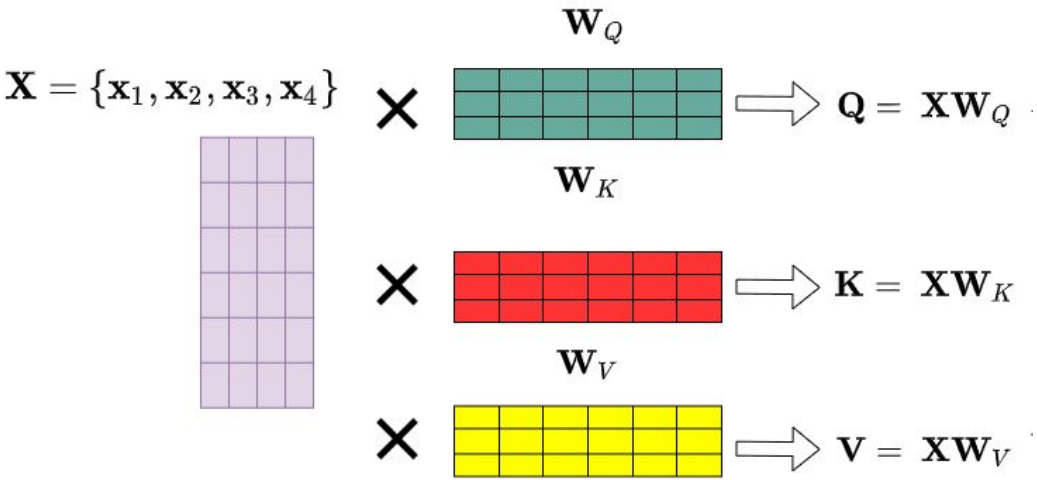



Image from AI Summer, Nicholas Adaloglu: Transformers

# To calculate attention, query, key, and value matrices are generated

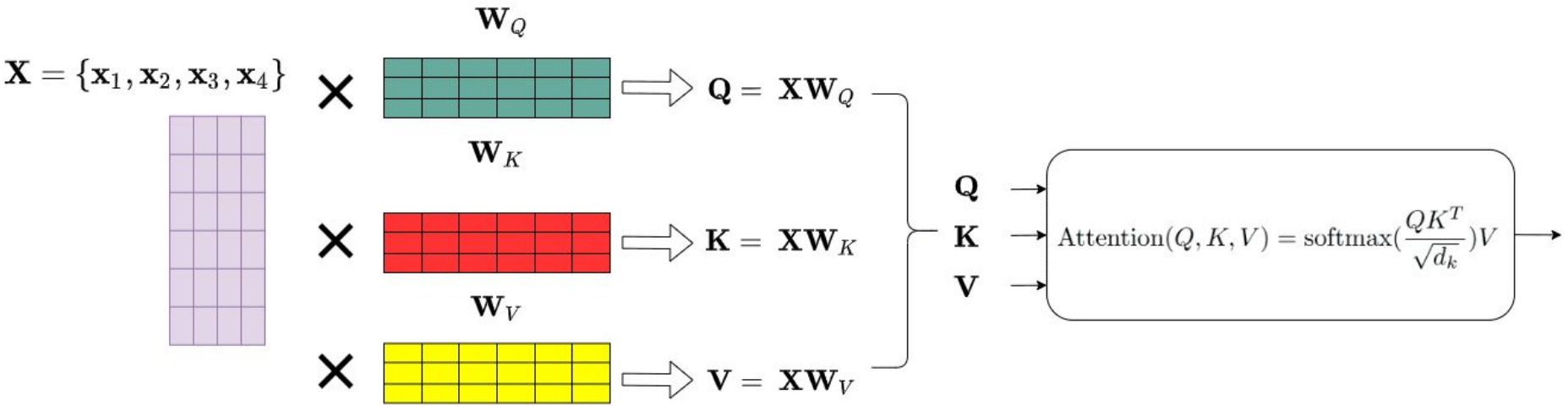



Image from AI Summer, Nicholas Adaloglu: Transformers

# Softmax takes in values and return a number between (0,1) → great for generating probabilities or as activation for multi categories

$$\sigma(\vec{z})_i = \frac{e^{z_i}}{\sum_{j=1}^{K} e^{z_j}}$$

$\sigma$ = softmax

$\vec{z}$ = input vector

$e^{z_i}$ = standard exponential function for input vector

$K$ = number of classes in the multi-class classifier

$e^{z_j}$ = standard exponential function for output vector

$e^{z_j}$ = standard exponential function for output vector

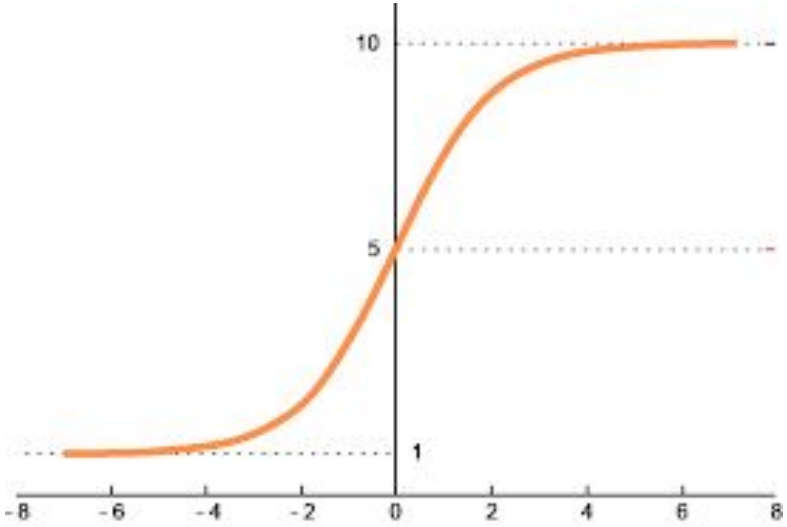
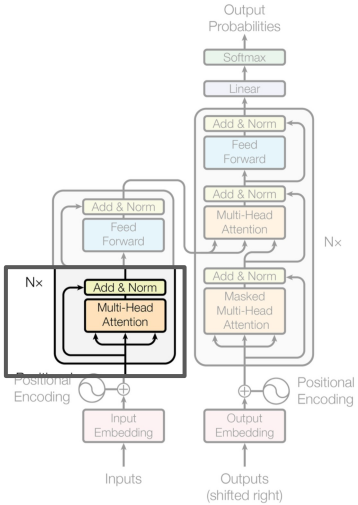
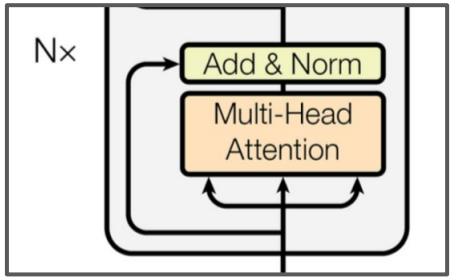



Image from AI Summer, Nicholas Adaloglu: Transformers

# The simplest attention is scaled dot product attention



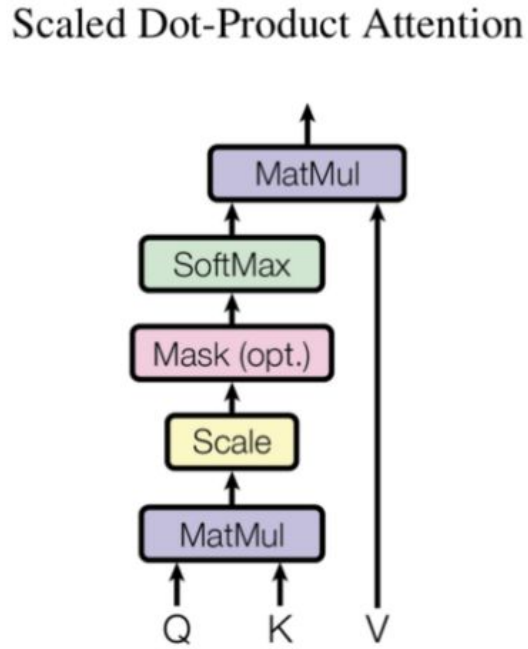
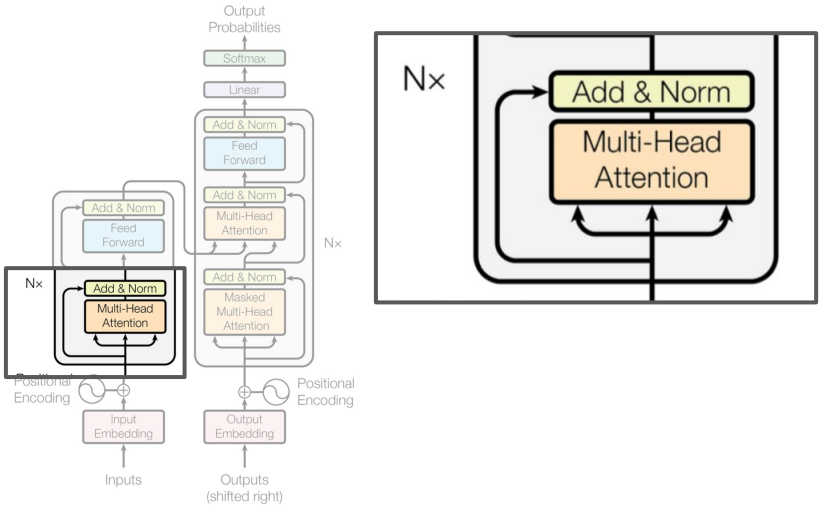




Image from medium: What is a transformer

# In most cases multi-head attentions are used







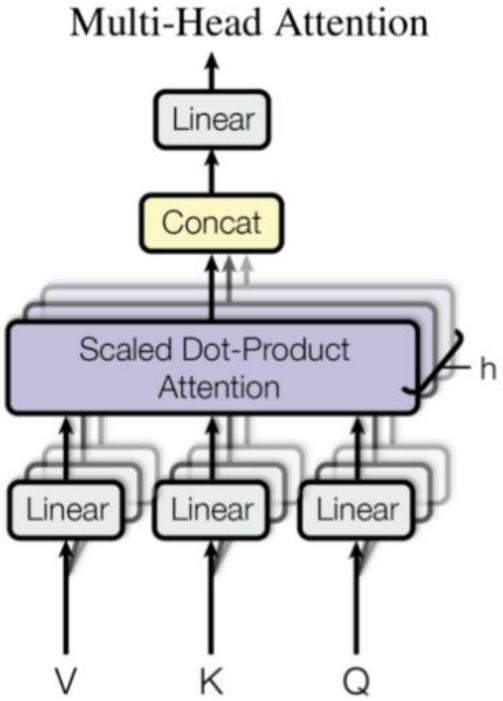



Image from medium: What is a transformer

# The attention layer is followed by short residual skip connections and a normalization layer

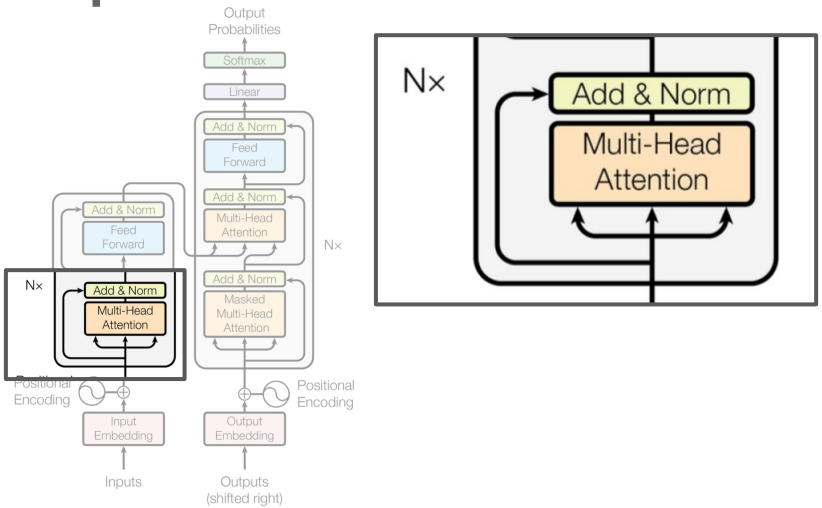
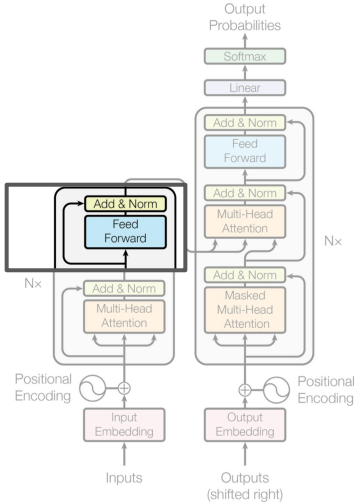


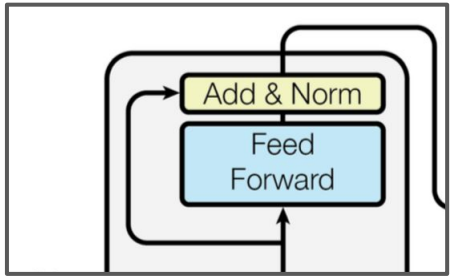




Image from AI Summer, Nicholas Adaloglu: Transformers

# The output of the attention is then fed into a feed forward layer



# The encoder is now complete!

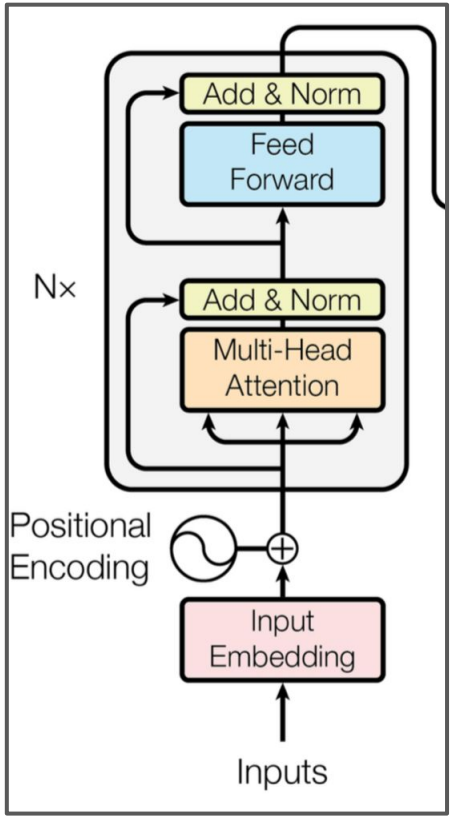



Image from medium: What is a transformer

# The decoder is in charge of creating the translation

The weather is great.

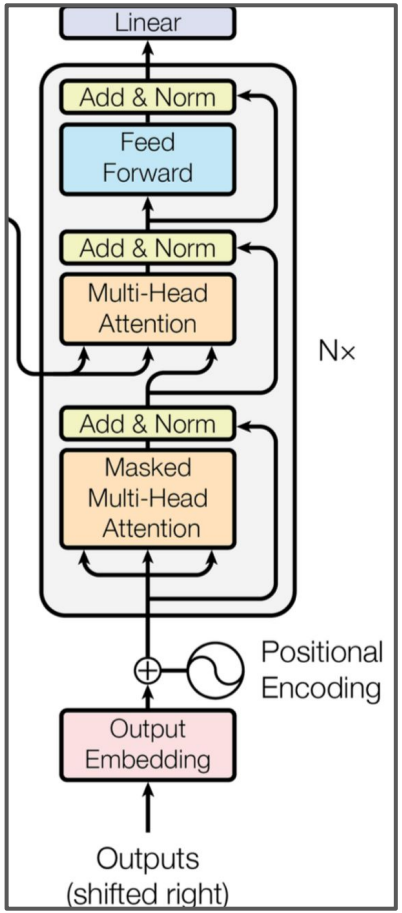



Image from medium: What is a transformer

# The decoder is in charge of creating the translation

The weather is great.

هوا

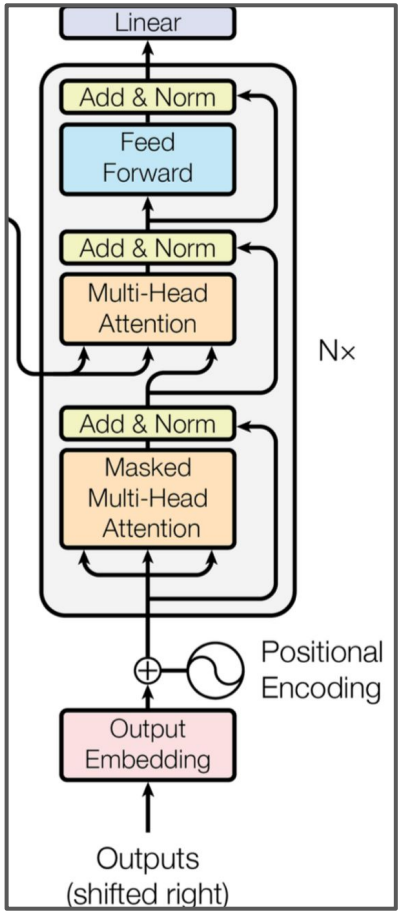



Image from medium: What is a transformer

# The decoder is in charge of creating the translation

The weather is great.

هوا عالی

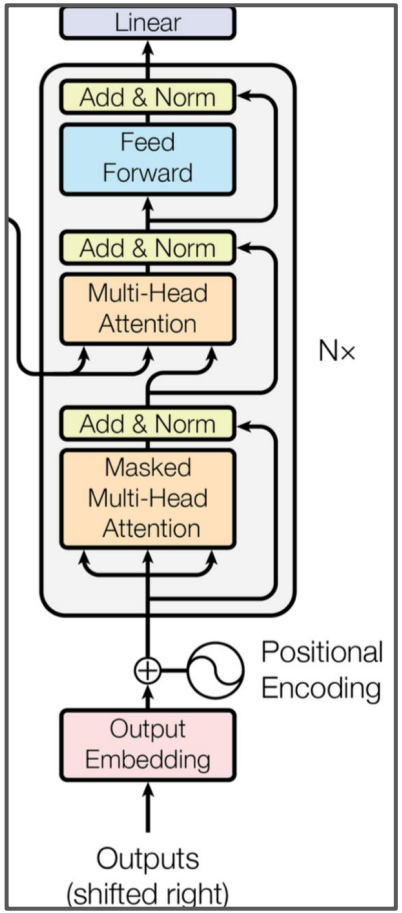



Image from medium: What is a transformer

# The decoder is in charge of creating the translation

The weather is great.

هوا عالی است

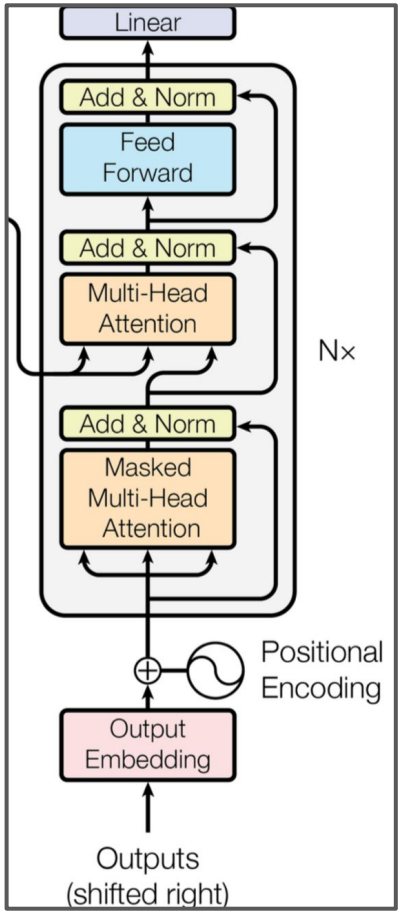



Image from medium: What is a transformer

# The decoder is in charge of creating the translation

The weather is great.

هوا عالی است.

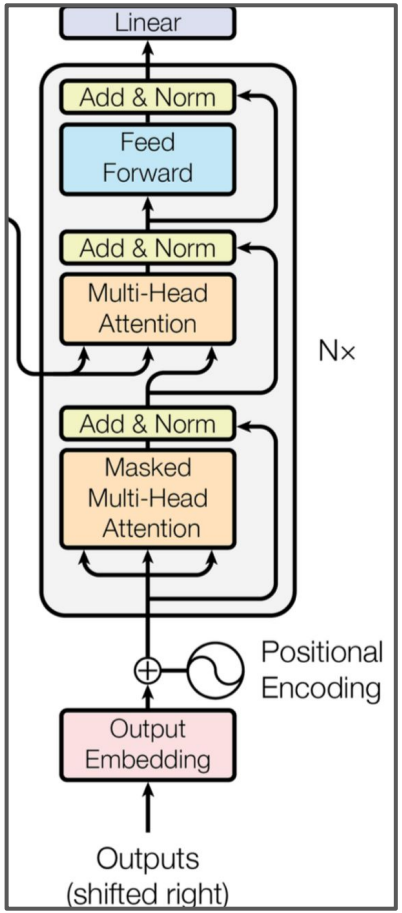
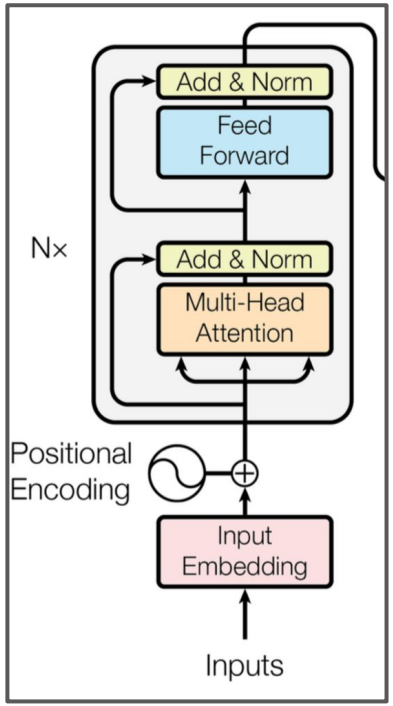



Image from medium: What is a transformer

# The decoder works in a way similar to the encoder



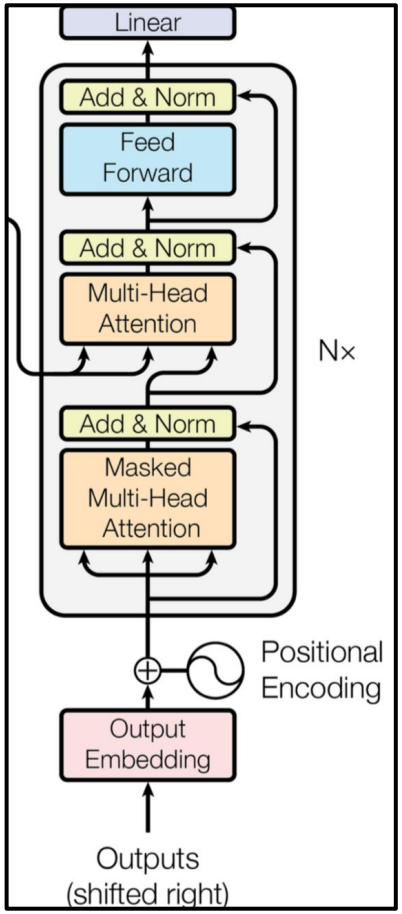
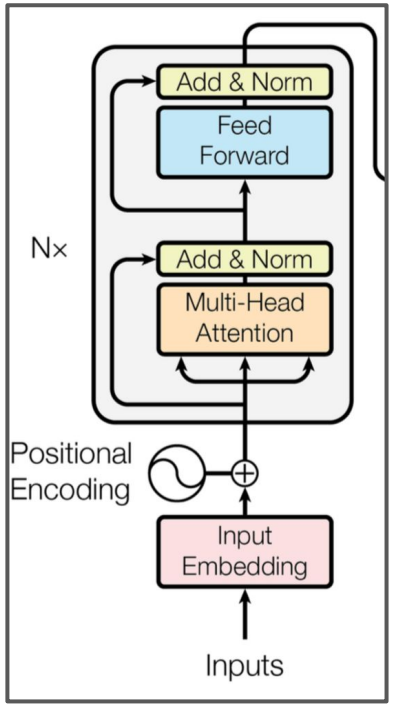




Image from medium: What is a transformer

# The decoder works in a way similar to the encoder



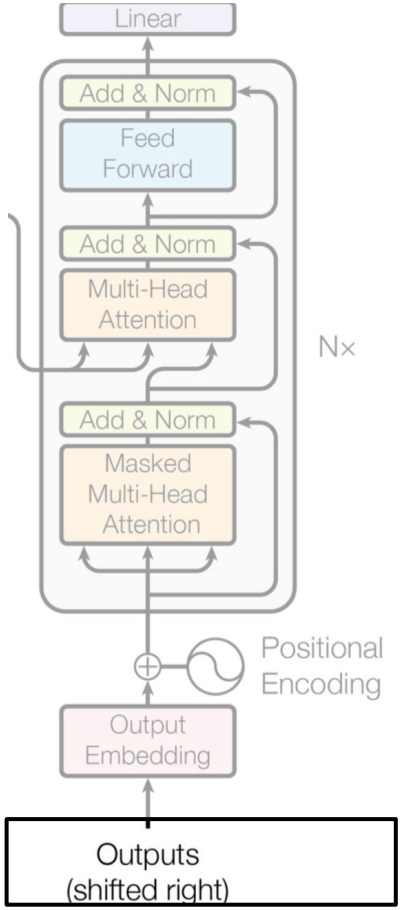
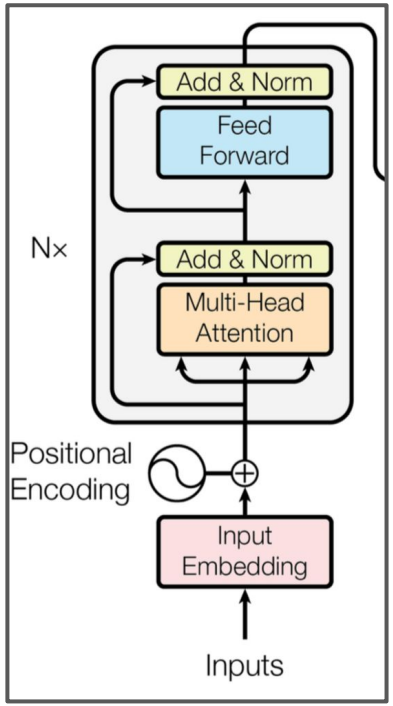




Image from medium: What is a transformer

# The decoder works in a way similar to the encoder



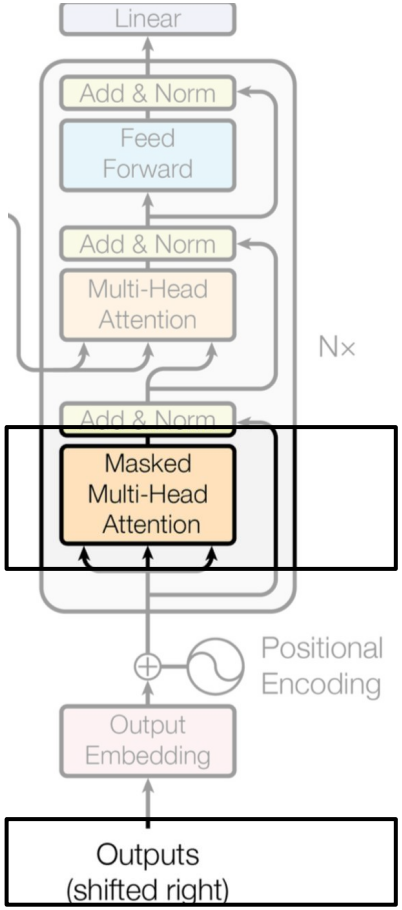
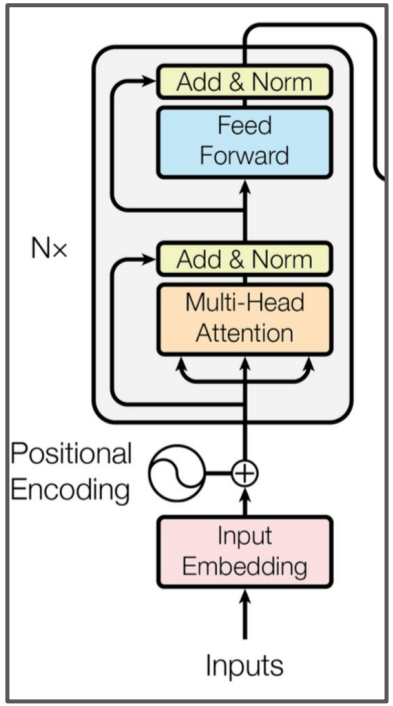




Image from medium: What is a transformer

# The decoder works in a way similar to the encoder



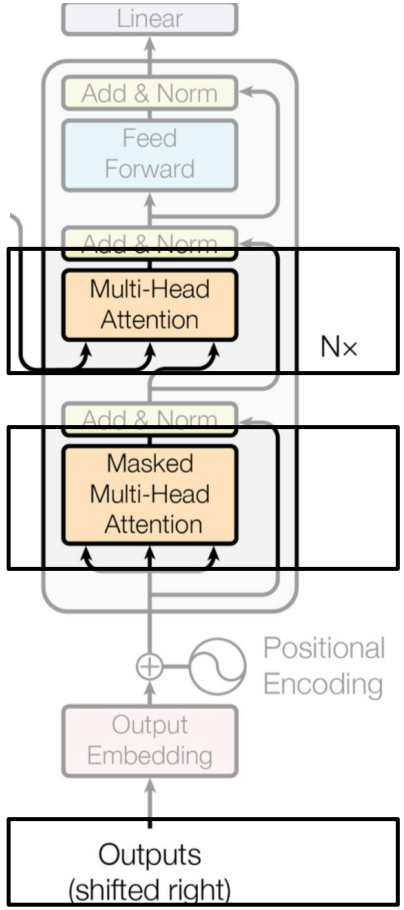




Image from medium: What is a transformer

# Shifting outputs …

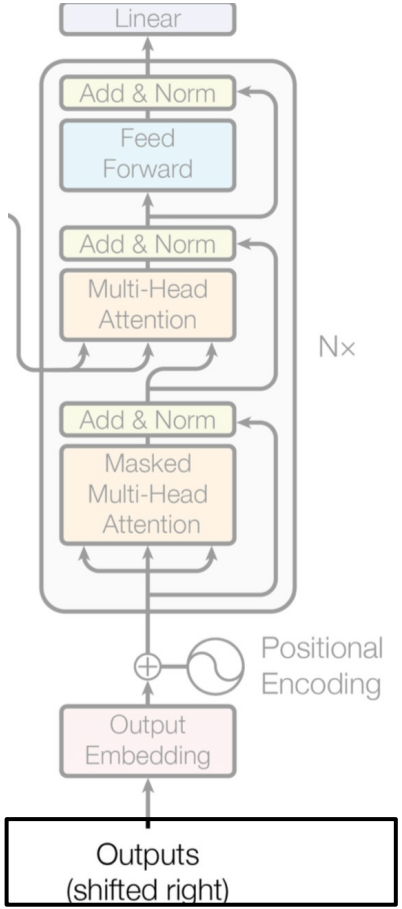



Image from medium: What is a transformer

# Shifting outputs prevent network from copying the decoder input

The weather is great.

هوا ⟵ هوا

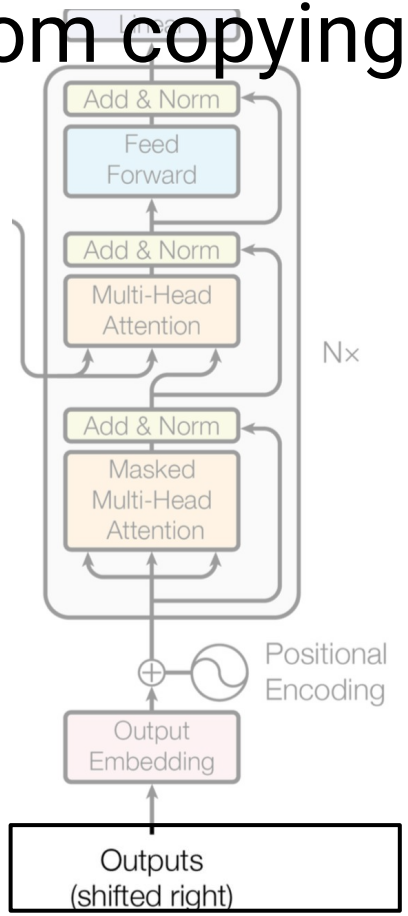



Image from medium: What is a transformer

# Shifting outputs prevent network from copying the decoder input

The weather is great.

هوا ← هوا

B ← هوا

| | |
|---|---|
| 14% | هوا |
| 12% | خورشید |
| 20% | من |
| 3% | درخت |
| ... | ... |

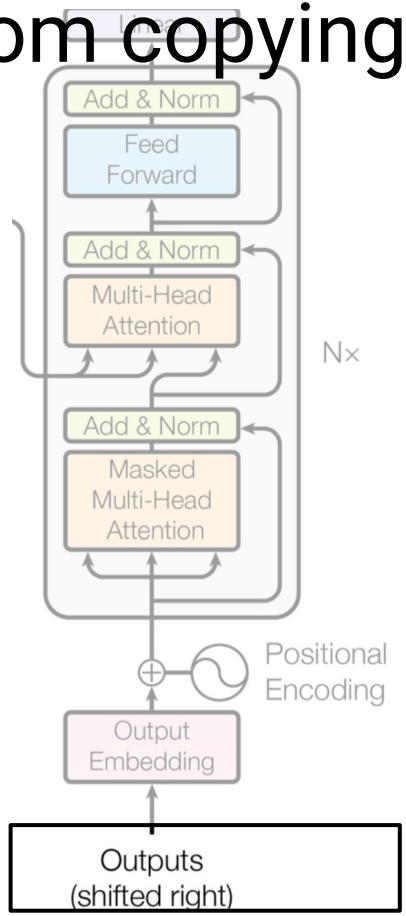



Image from medium: What is a transformer

# Masked layers ensure that we're only seeing information before the word to predict

$$\text{MaskedAttention}(\mathbf{Q}, \mathbf{K}, \mathbf{V}) = \text{softmax}\left(\frac{\mathbf{Q}\mathbf{K}^T + \mathbf{M}}{\sqrt{d_k}}\right)\mathbf{V}$$

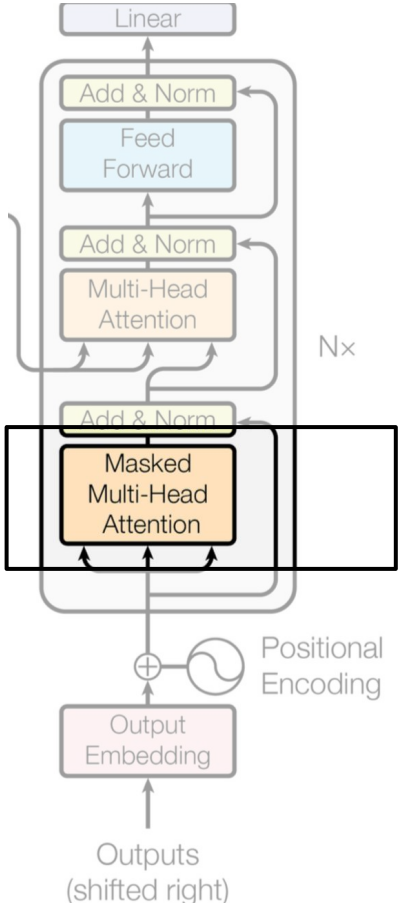



Image from AI Summer, Nicholas Adaloglu: Transformers

Image from medium: What is a transformer

# Encoder-decoder attention is where the magic happens

The output of encoder:

Keys and values

The decoder's latest sentence:

The query

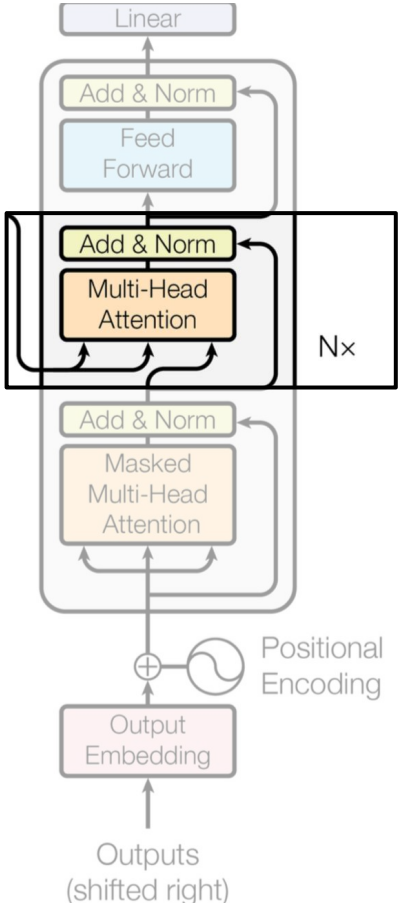



Image from medium: What is a transformer

# The final step is another linear layer and a softmax

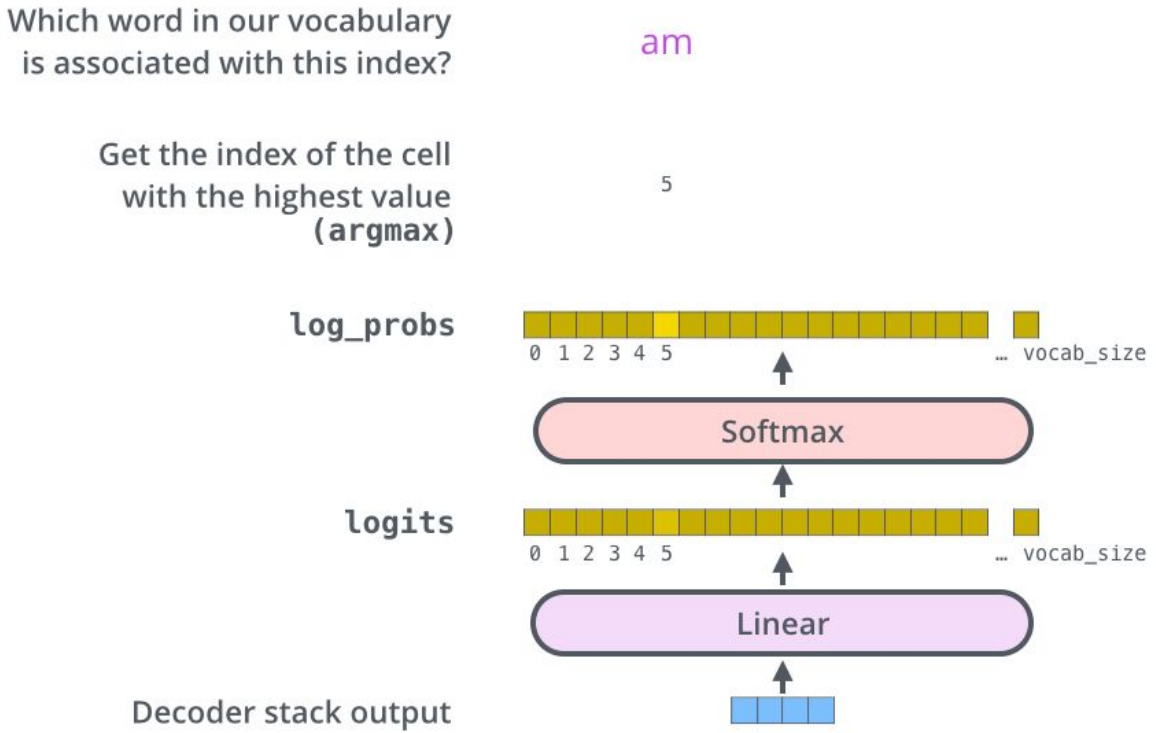
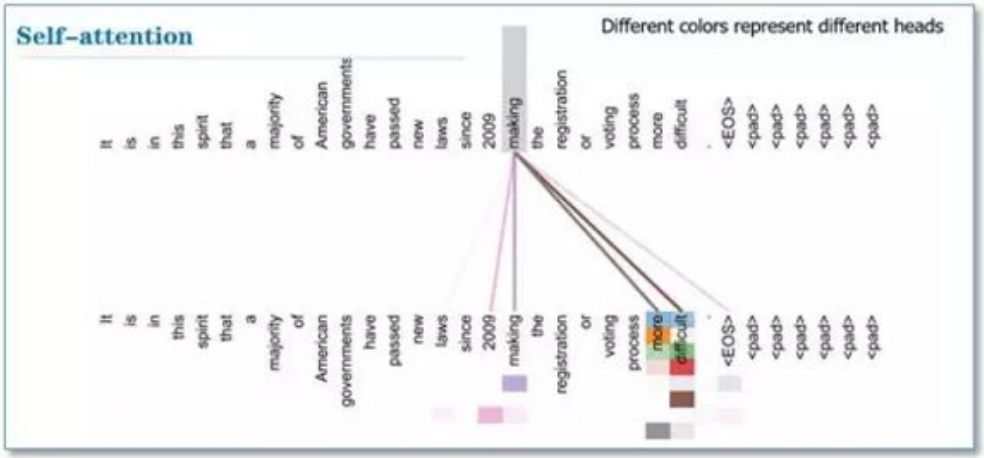



Image from Jay Alammar

# Basic architecture of a transformer



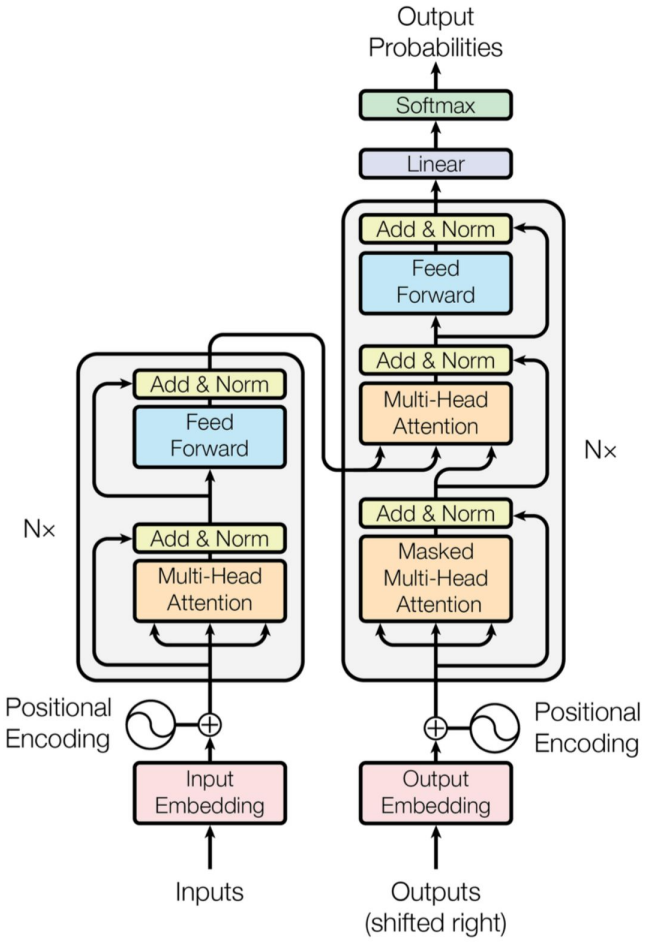




Image from Alibaba Clouder: Self-attention

Image from medium: What is a transformer

# Why do transformers work so well?

# Why do transformers work so well?

1. Distributed and independent representation at each block
2. Meaning heavily depends on the context
3. Multiple encoder and decoder units
4. Combination of high- and low-level information

# Famous transformers

GPT3

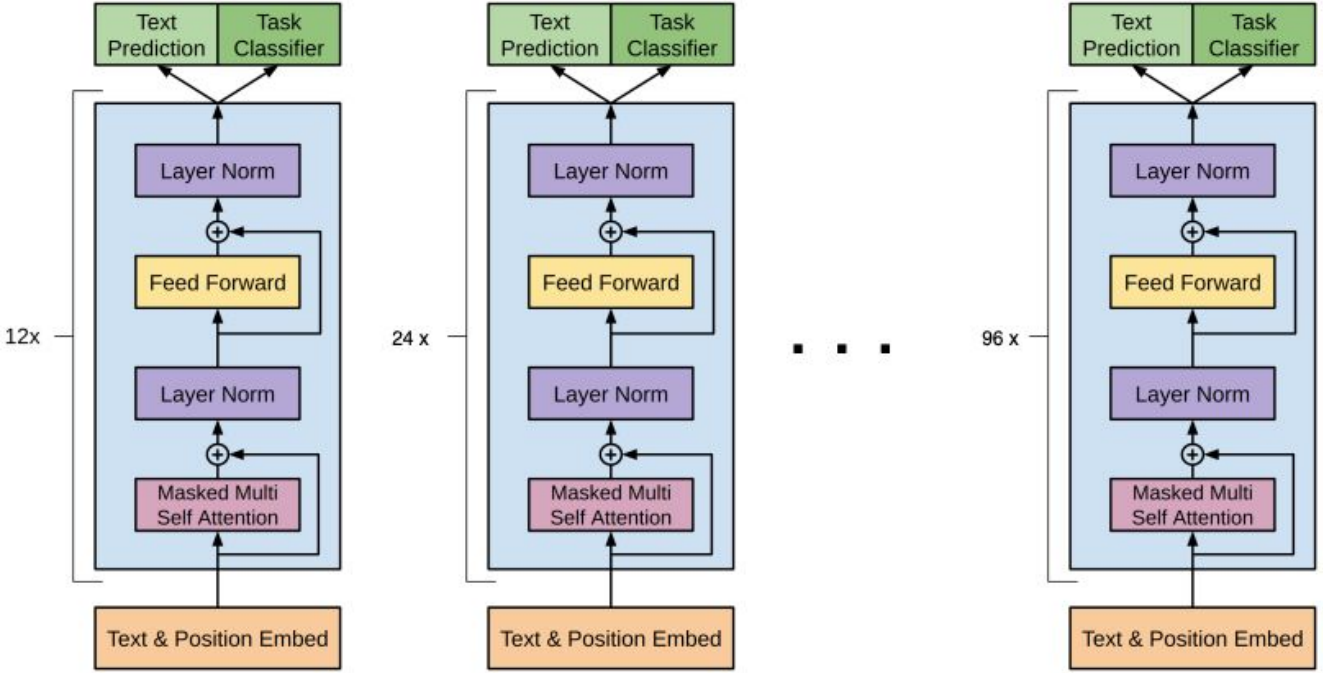



Image from dzlab: GPT3 explained

# Famous transformers

BERT
(Bidirectional Encoder Representations from Transformers)

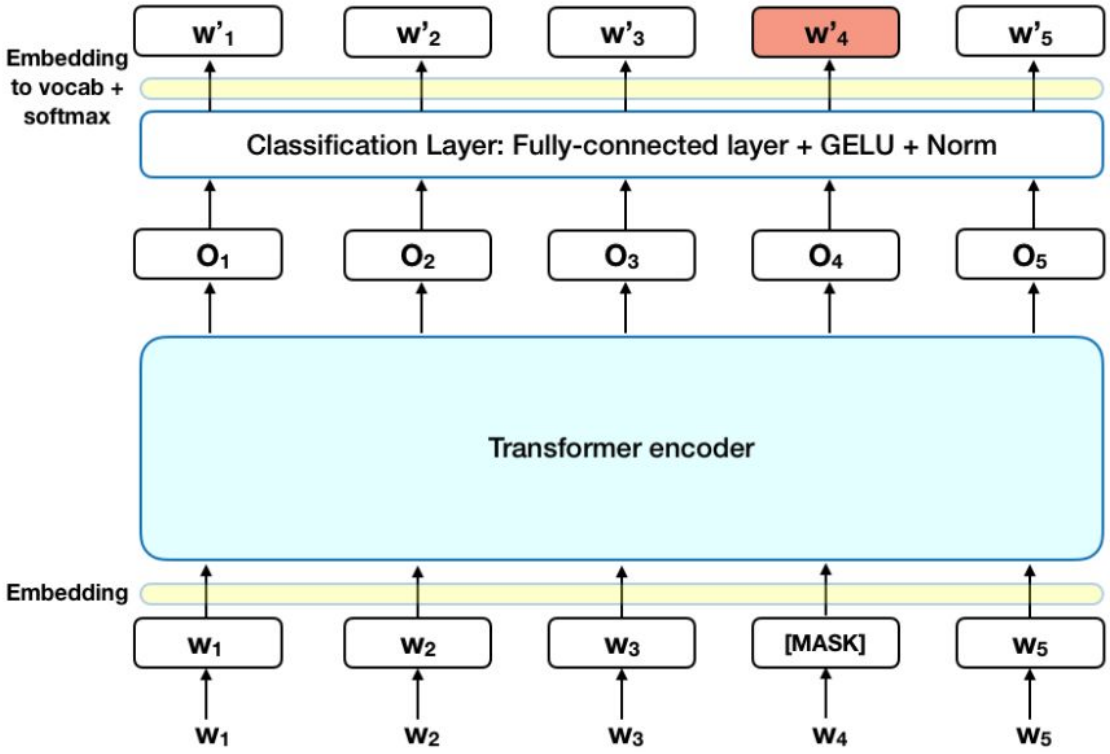



Image from Rani Horev, TDS: BERT explained

# Next lecture:

## *AlphaFold2 and DL in protein engineering*

**Kristine Deibler** · 1st
Computational Design Scientist at Novo Nordisk

**Nikhil Naik** · 2nd
Senior Research Manager | Machine learning, Computer Vision, NLP, AI for Biology

**Layne Price** · 1st
Sr Machine Learning Scientist at Amazon

**Jack Maguire** · 1st
Senior Scientist at Genentech