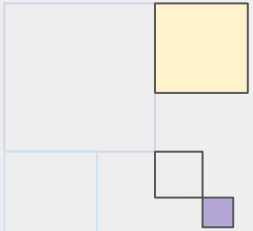
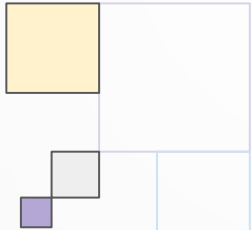
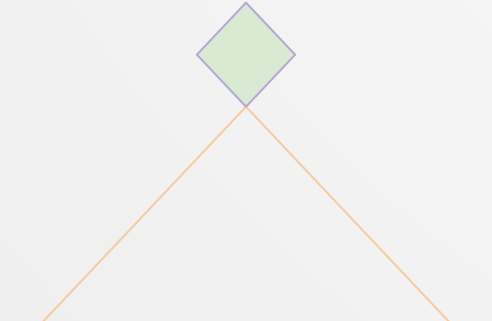




Deep Learning

Language Modelling



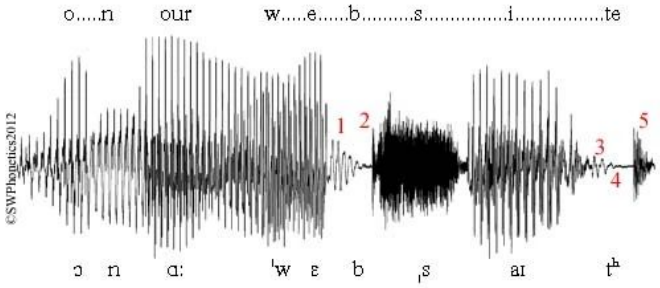
Bryan Cardenas
Caspar van Leeuwen
Monica Rotulo
Robert Jan Schlimbach

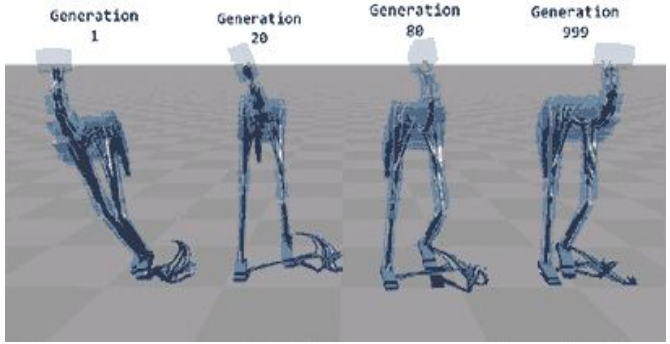
SURF

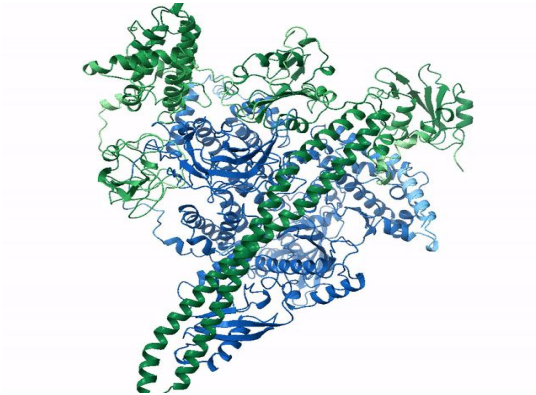
Sequential Data

Universe seems to be inherently sequential

We follow the arrow of time (no entropy reversals)



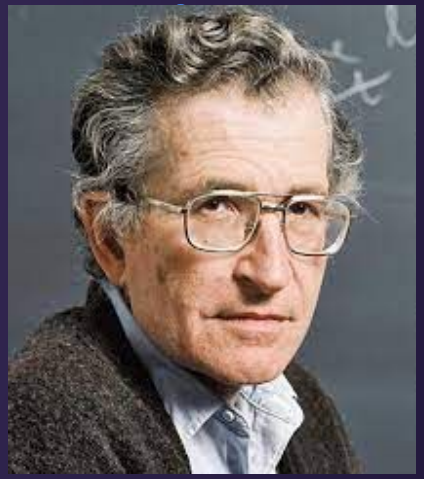




How Can we model this?

Is there one (sequential) model to rule them all?

Is modeling language difficult?



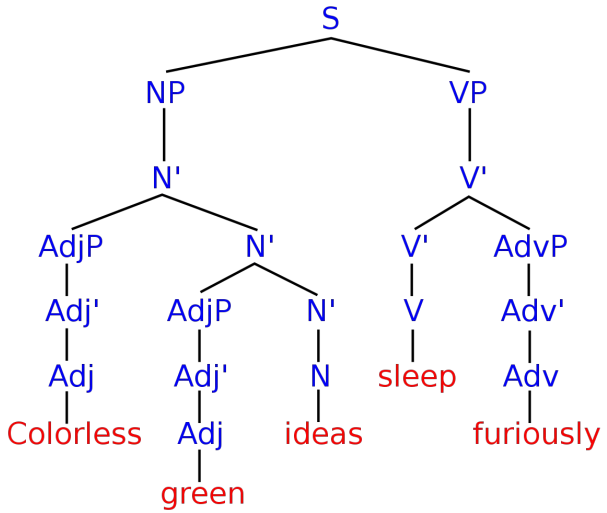
> Language is a process of free creation; its laws and principles are **fixed**, but the manner in which the principles of generation are used is free and **infinitely varied.**
>
> ~ Noam Chomsky

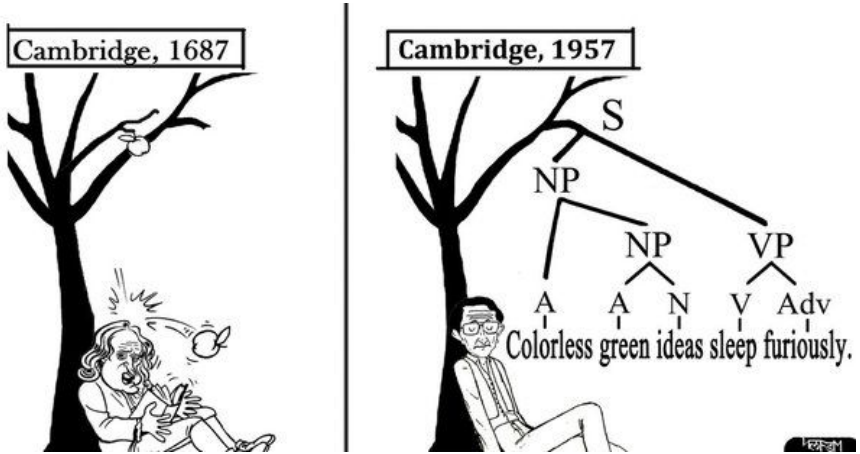
Sequential Data

Natural Language Modelling

Syntactic Structures

(**Chomsky**, N. 1957)





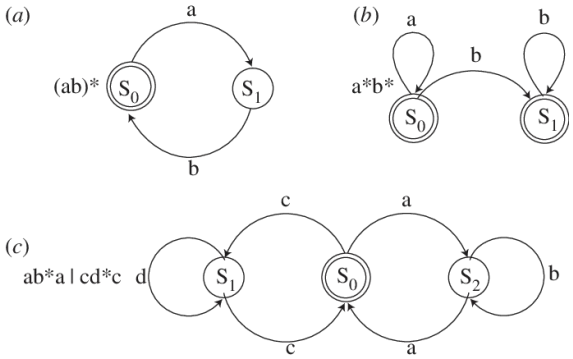
Sequential Data

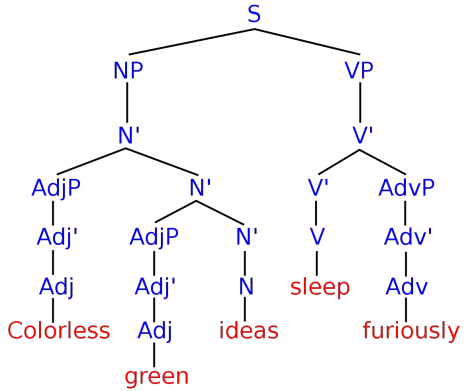
Natural Language Modelling

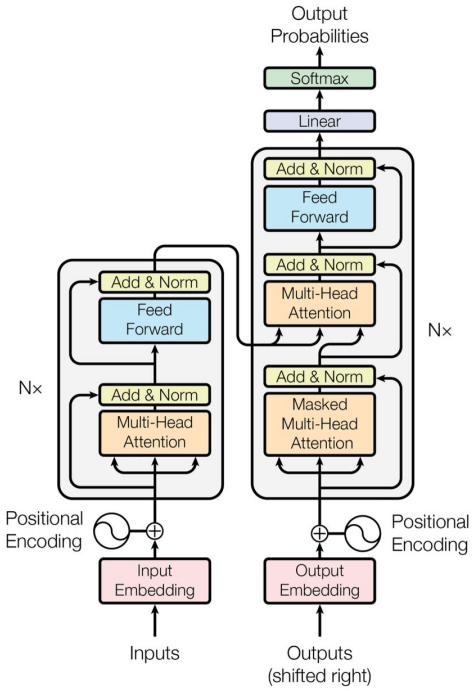
We have the innate ability for language (Universal Grammar)

Different ways to model language

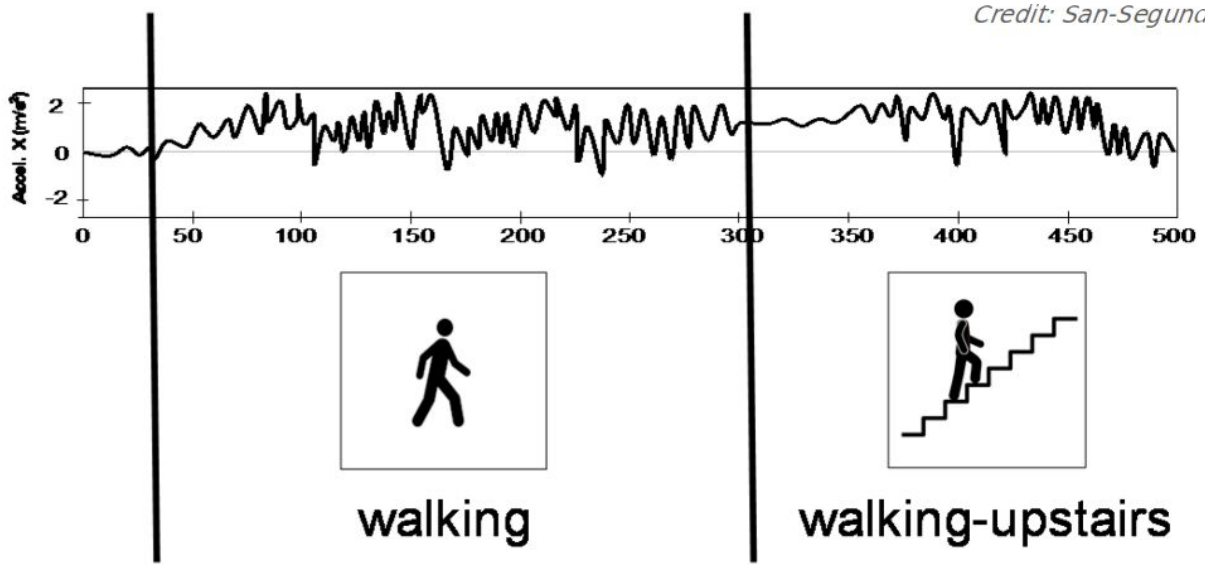
1. Finite State Automata
2. Generative Grammars
3. Deep Learning + Data







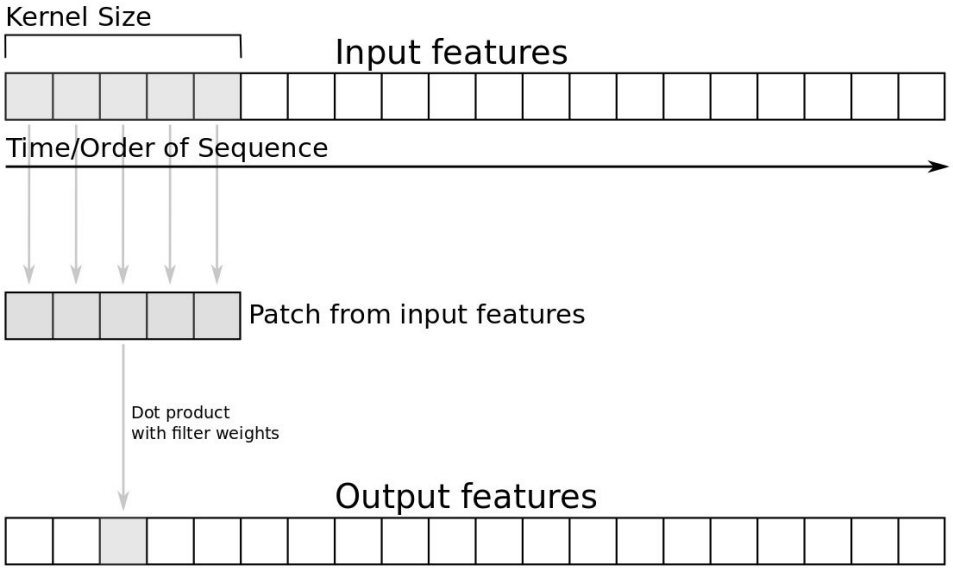
Human Activity Recognition



Human Activity Recognition

CNNs work here because they take into account **nearby** tokens

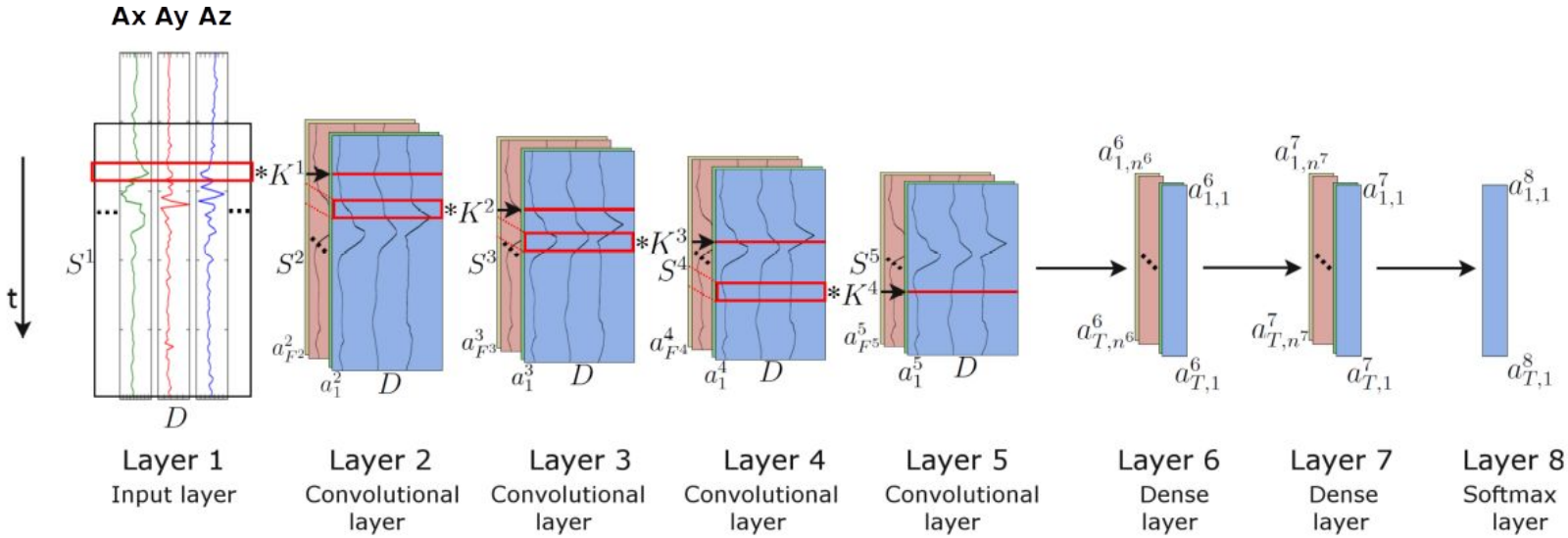
Use a CNN!



Human Activity Recognition

CNNs work here because they take into account **nearby** tokens

Use a CNN!



How to model Language?

$$P(w_1, w_2, \ \ldots \ w_n) = P(w_1)P(w_2 \mid w_1)P(w_3 \mid w_1, w_2) \ \ldots \ P(w_n \mid w_1, \ \ldots \ w_{n-1})$$

LM objective:

Model the probability of the next word given the previous context

How to model Language?

$$P(w_1, w_2, \ \ldots \ w_n) = P(w_1)P(w_2 \mid w_1)P(w_3 \mid w_1, w_2) \ \ldots \ P(w_n \mid w_1, \ \ldots \ w_{n-1})$$

Use a CNN? **NOPE**

LM objective:

Model the probability of the next word given the previous context

SURF

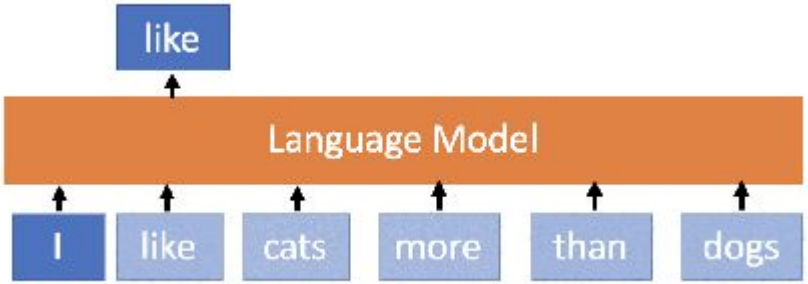
How to model Language?

$$P(w_1, w_2, \ \ldots \ w_n) = P(w_1)P(w_2 \mid w_1)P(w_3 \mid w_1, w_2) \ \ldots \ P(w_n \mid w_1, \ \ldots \ w_{n-1})$$

Use a CNN? NOPE

LM objective:

Model the probability of the next word given the previous context



The cat

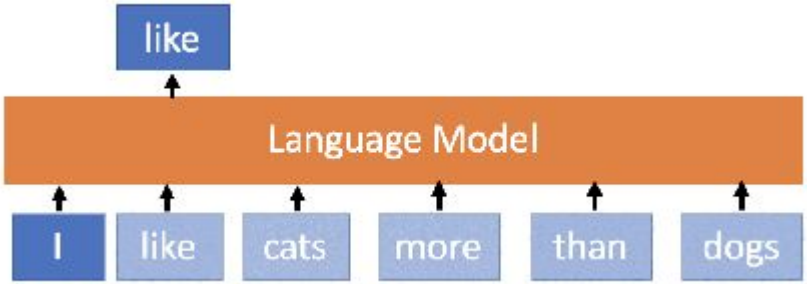
How to model Language?

$$P(w_1, w_2, \ \ldots \ w_n) = P(w_1)P(w_2 \mid w_1)P(w_3 \mid w_1, w_2) \ \ldots \ P(w_n \mid w_1, \ \ldots \ w_{n-1})$$

Use a CNN? NOPE

LM objective:

Model the probability of the next word given the previous context

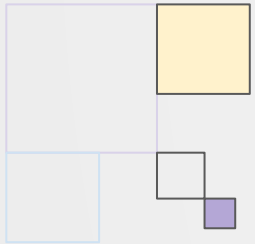


The cat

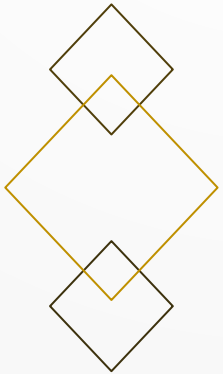
CNN's context size is too small and does **not *remember*** past context

How to even represent language as a continuous signal?

Language is very noise and heavily (long-distance) context dependent



The Transformer



High Performance Machine Learning Group

SURF

Transformers

Get rid of Recurrent Neural Networks

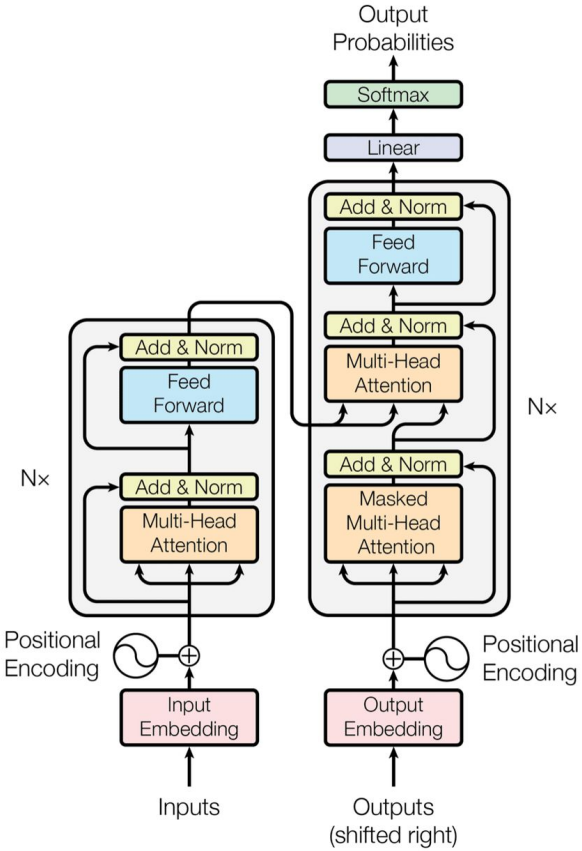
Instead of keeping one hidden state, **keep track of multiple**

What is the **importance** of every element in the input sequence with respect to it**self?**

Every word connected to every other word

More parallelization!

Self Attention



Transformers

Get rid of Recurrent Neural Networks

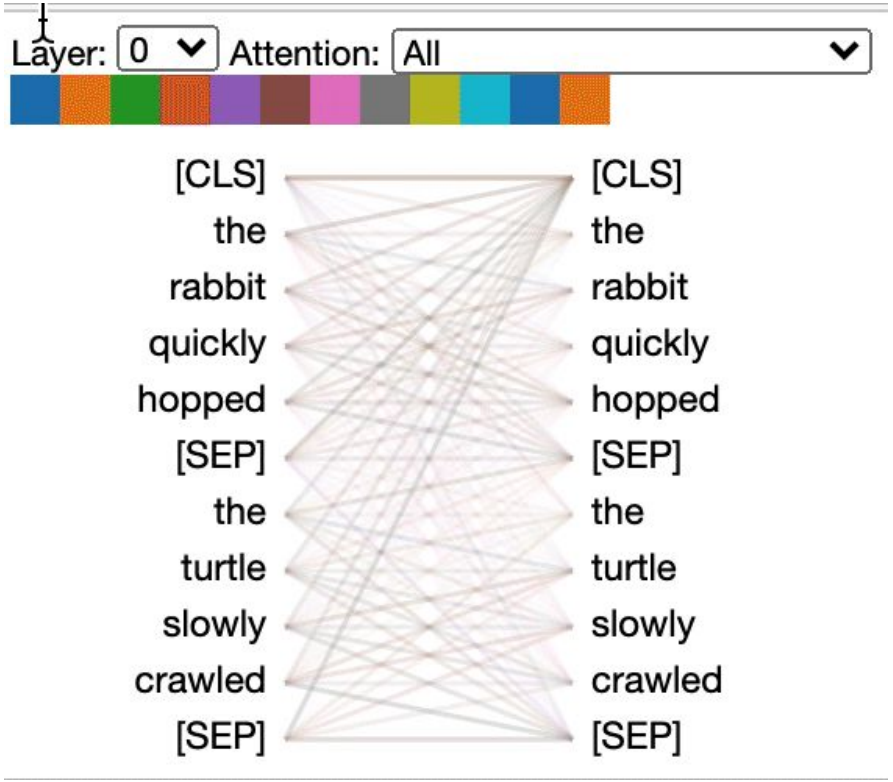
Instead of keeping one hidden state, **keep track of multiple**

What is the **importance** of every element in the input sequence with respect to it**self?**

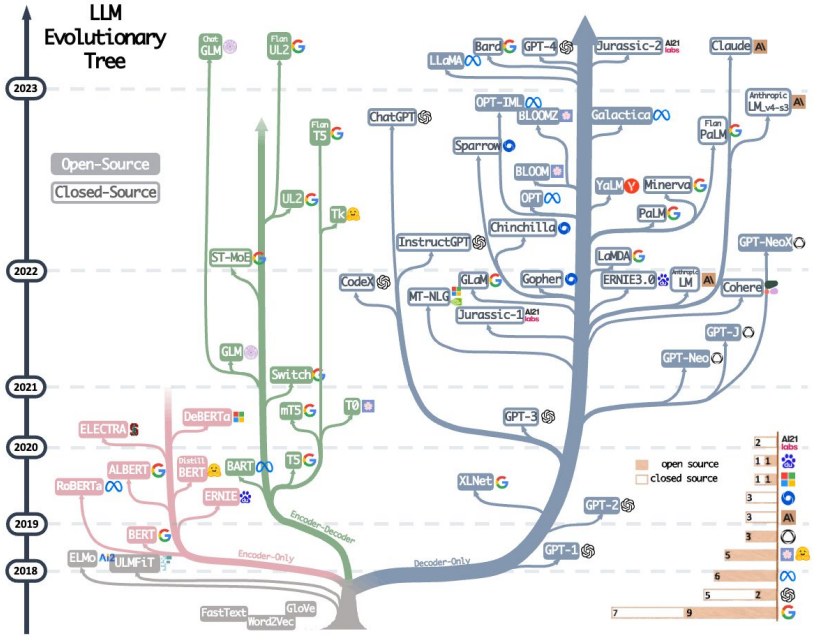
Every word connected to every other word

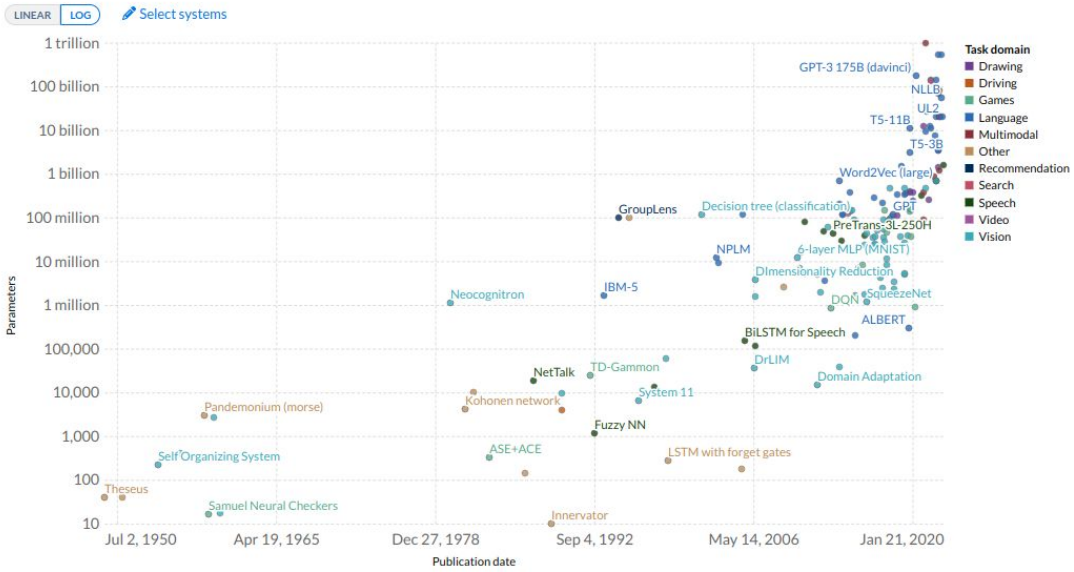
More parallelization!

Self Attention

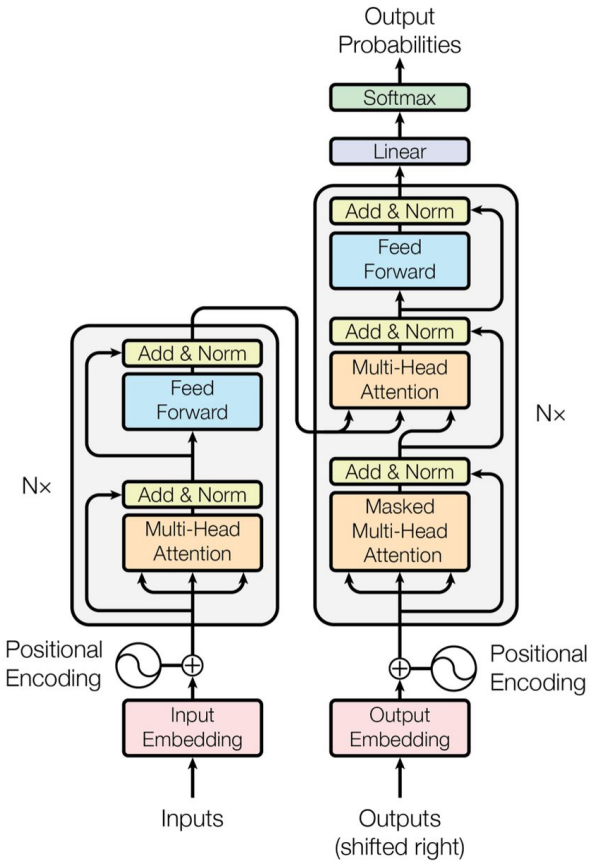


Cambrian Explosion of LMs

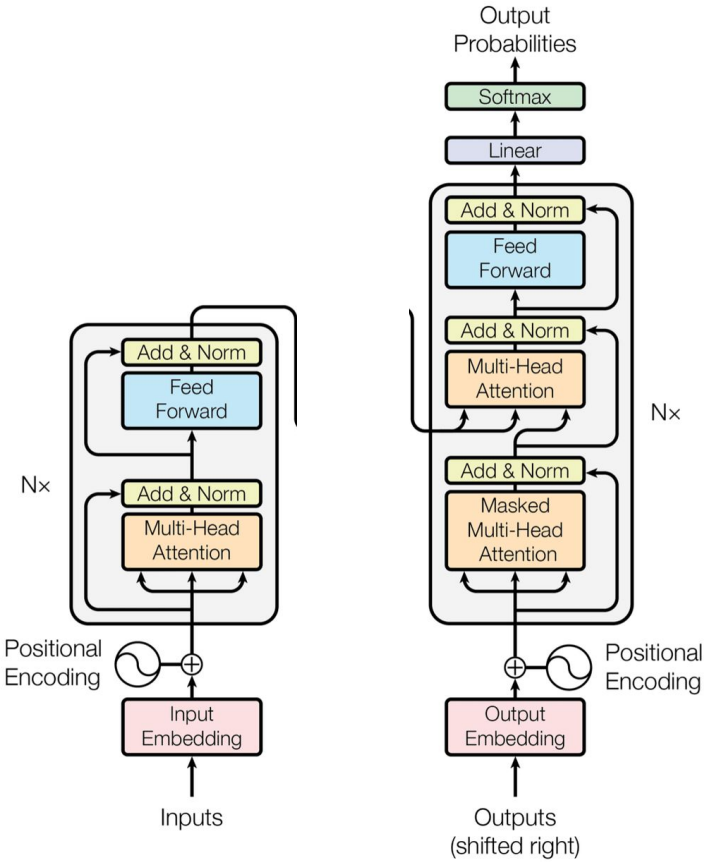




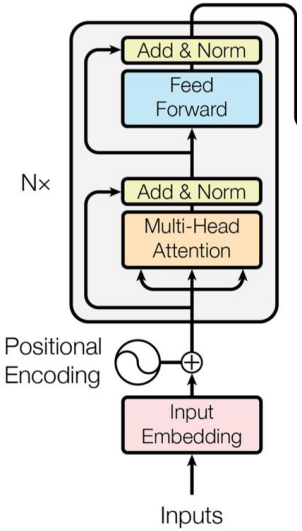
Dissection of the Transformer

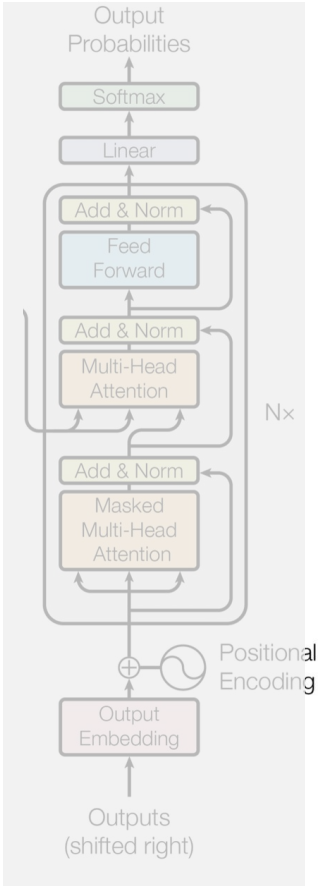


Dissection of the Transformer



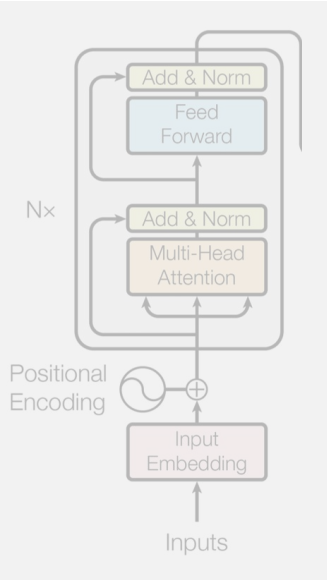
Dissection of the Transformer

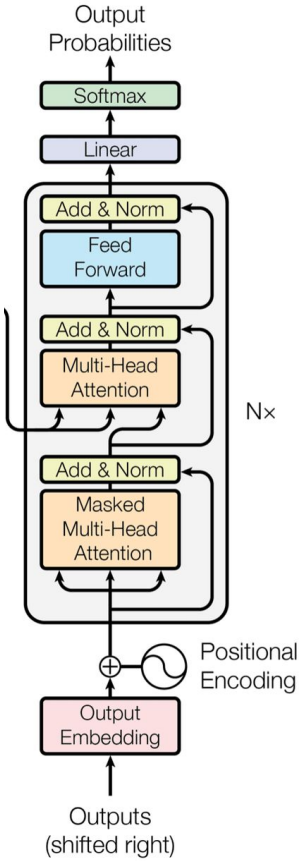




01. The Encoder

Dissection of the Transformer

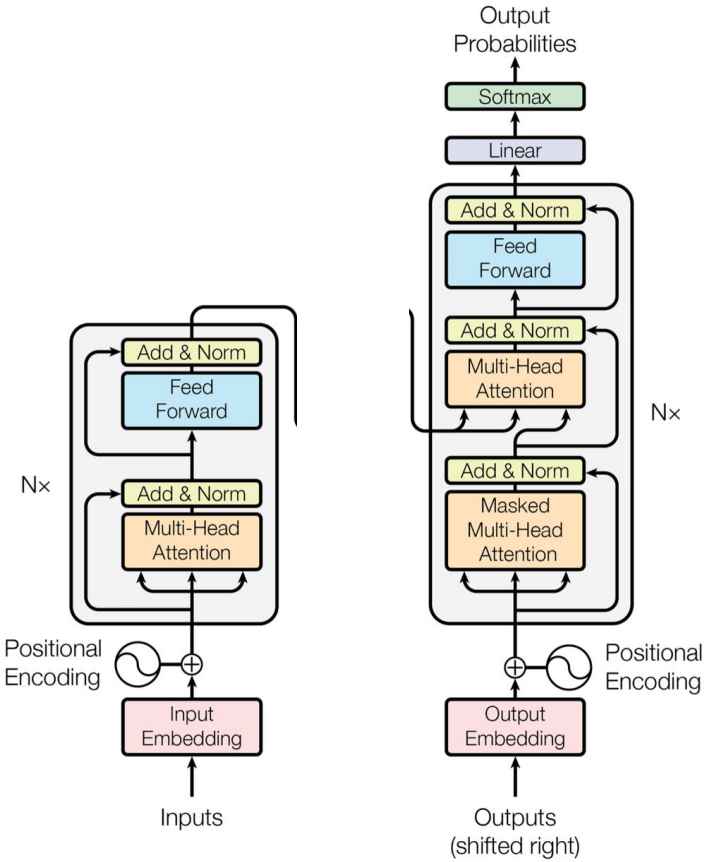




01. The Encoder

02. The Decoder

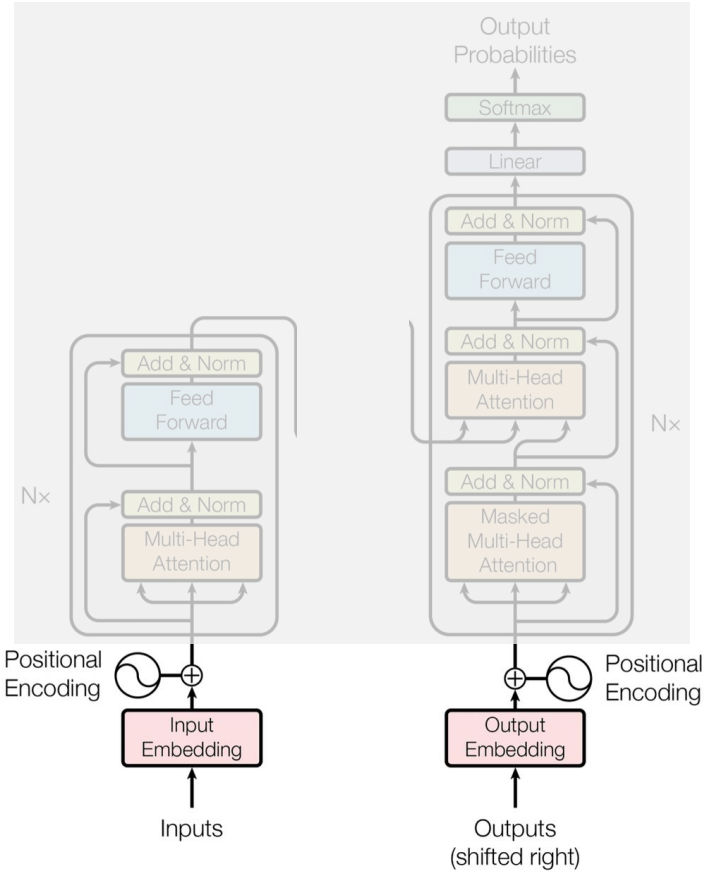
Dissection of the Transformer



01. The Encoder

02. The Decoder

Dissection of the Transformer

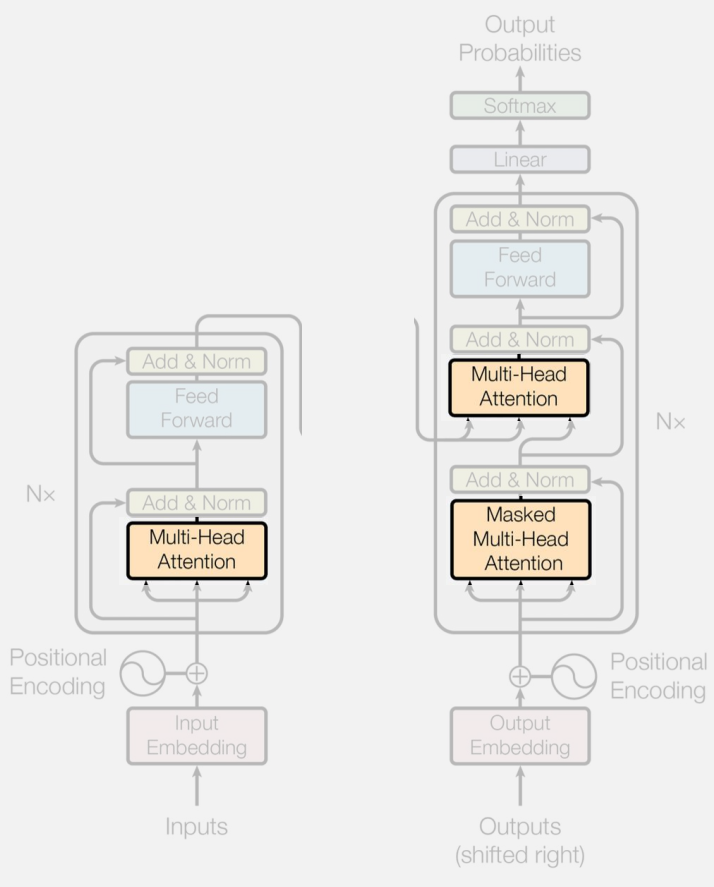


01. The Encoder

02. The Decoder

Input Embeddings

Dissection of the Transformer



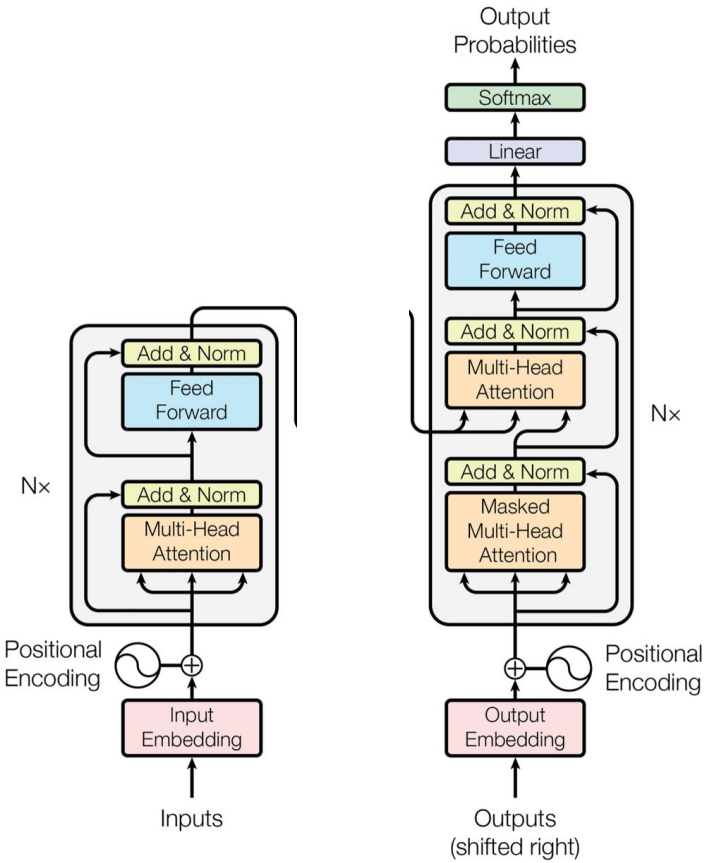
01. The Encoder

02. The Decoder

Self-Attention

Input Embeddings

Dissection of the Transformer



01. The Encoder

02. The Decoder

Self-Attention

Input Embeddings

Layer Normalization

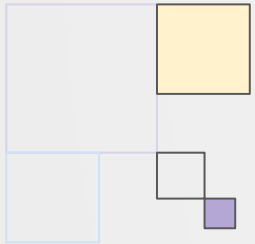
Skip Connections

Positional Encoding

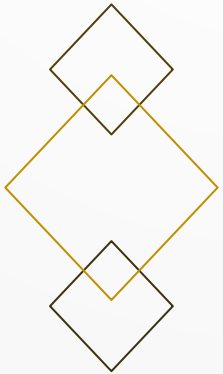
Byte Pair Encodings

Layer Stacking

Cross Attention



Neural Representations



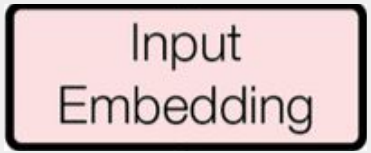
High Performance Machine Learning Group

SURF

How do we represent words?

Language is **discrete**!

> One cat ate a tuna, one cat ate a chicken, one weird cat ate a human.

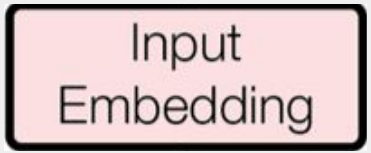


SURF

How do we represent words?

Language is **discrete**!

> One cat ate a tuna, one cat ate a chicken, one weird cat ate a human.

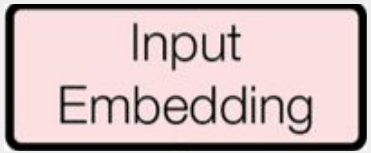


SURF

How do we represent words?

Language is **discrete**!

> One cat ate a tuna, one cat ate a chicken, one weird cat ate a human.



Vocabulary

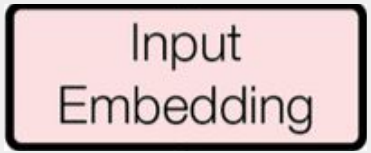
A vocabulary is the set of unique tokens in your language (dataset)

SURF

How do we represent words?

Language is **discrete**!

> One cat ate a tuna, one cat ate a chicken, one weird cat ate a human.



Vocabulary

A vocabulary is the set of unique tokens in your language (dataset)

SURF

How do we represent words?

Language is **discrete**!

> One cat ate a tuna, one cat ate a chicken, one weird cat ate a human.

Input Embedding

Vocabulary

A vocabulary is the set of unique tokens in your language (dataset)

one
cat
ate
tuna
chicken
weird
human
a

SURF

How do we represent words?

Language is **discrete**!

> One cat ate a tuna, one cat ate a chicken, one weird cat ate a human.

Input Embedding

Vocabulary

A vocabulary is the set of unique tokens in your language (dataset)

one	0
cat	1
ate	2
tuna	3
chicken	4
weird	5
human	6
a	7

How do we represent words?

Language is **discrete**!

> One cat ate a tuna, one cat ate a chicken, one weird cat ate a human.

Input Embedding

Vocabulary

A vocabulary is the set of unique tokens in your language (dataset)

SURF

How do we represent words?

Language is **discrete**!

> One cat ate a tuna, one cat ate a chicken, one weird cat ate a human.

Easy to **compose** and easy to **vectorize**

Input Embedding

Vocabulary

A vocabulary is the set of unique tokens in your language (dataset)

SURF

How do we represent words?

Language is **discrete**!

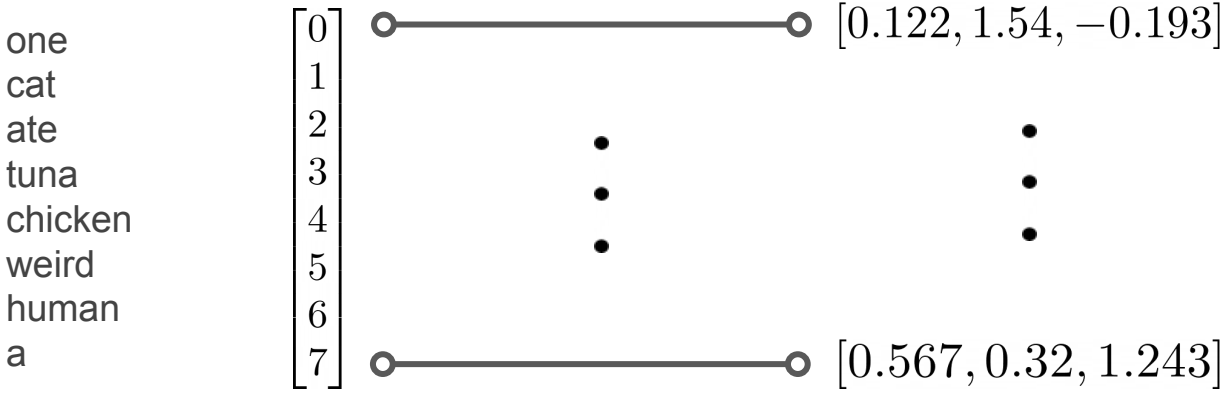
> One cat ate a tuna, one cat ate a chicken, one weird cat ate a human.

Easy to **compose** and easy to **vectorize**

Input Embedding

Vocabulary

A vocabulary is the set of unique tokens in your language (dataset)



SURF

How do we represent words?

Language is **discrete**!

> One cat ate a tuna, one cat ate a chicken, one weird cat ate a human.

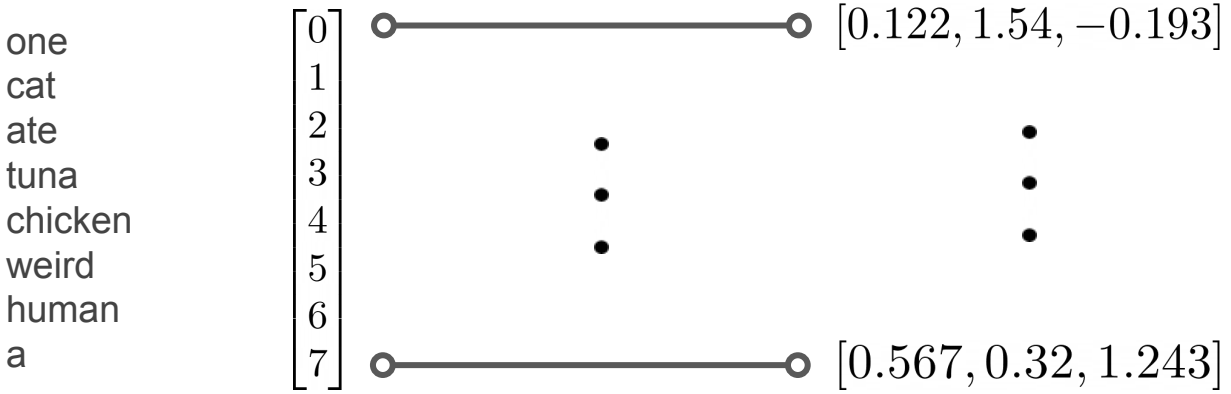
Easy to **compose** and easy to **vectorize**

Input Embedding

Vocabulary

A vocabulary is the set of unique tokens in your language (dataset)

Embedding



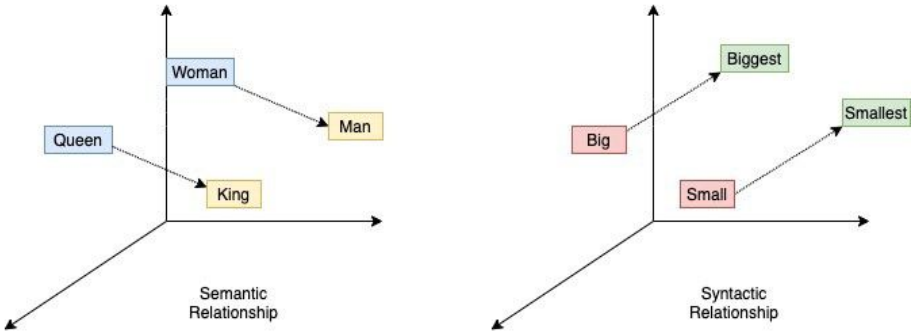
SURF

Neural Representation Learning

King $[0.122, 1.54, -0.193]$

Man $[0.567, 0.32, 1.243]$

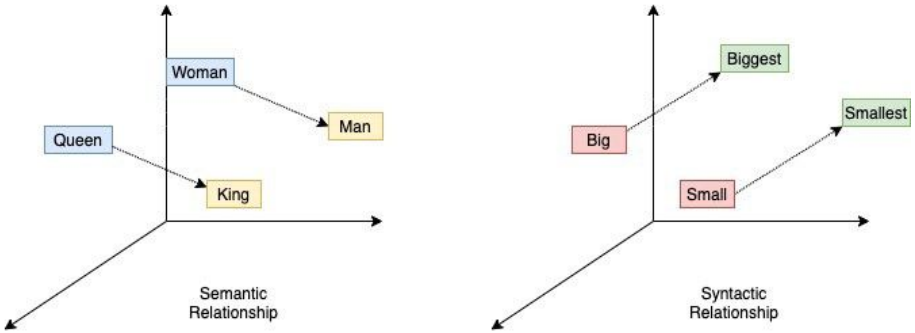
Neural Representation Learning



King $[0.122, 1.54, -0.193]$

Man $[0.567, 0.32, 1.243]$

Neural Representation Learning



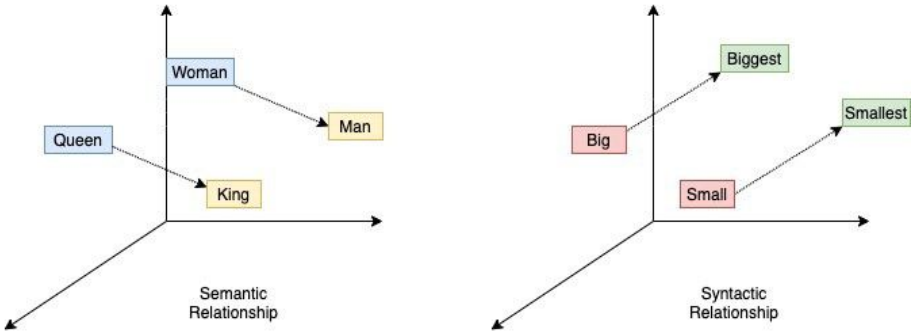
King $[0.122, 1.54, -0.193]$

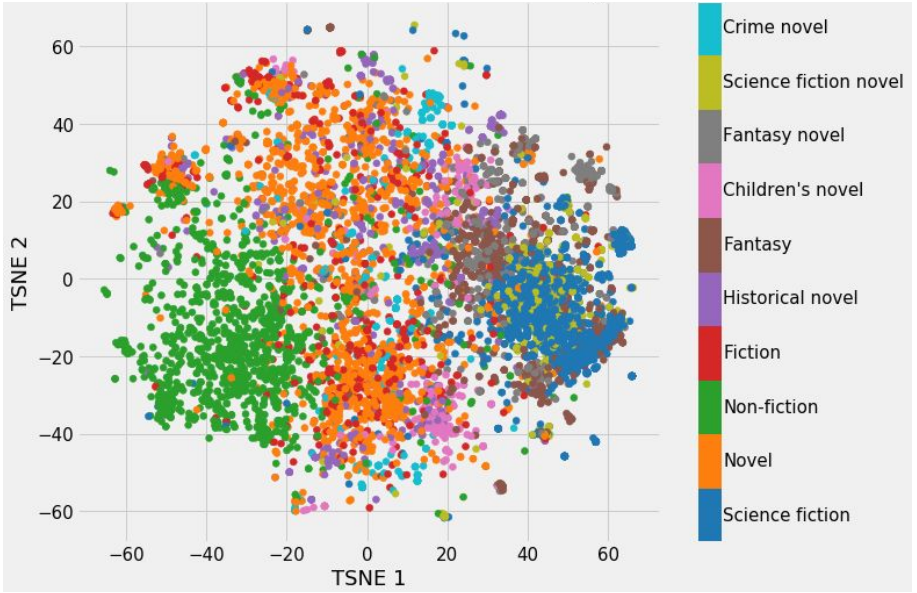
Man $[0.567, 0.32, 1.243]$

King - Man + Woman = Queen

Each word has a numerical vector representation learnt by a language model

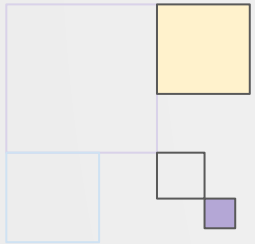
Neural Representation Learning



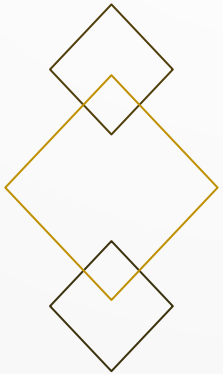


King - Man + Woman = Queen

Each word has a numerical vector representation learnt by a language model



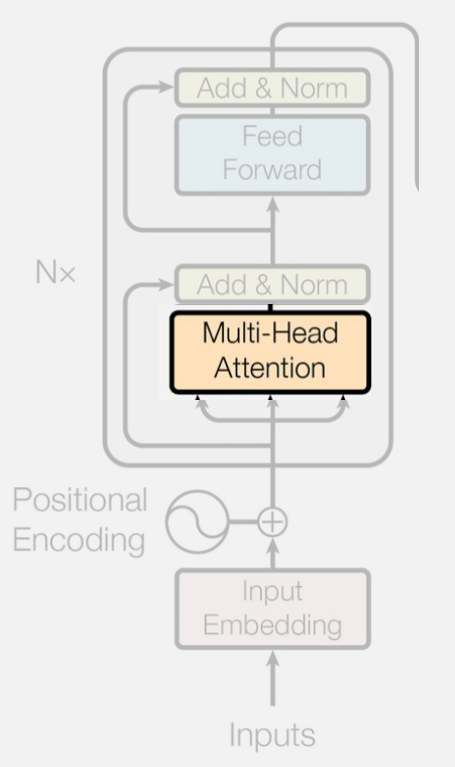
Self Attention



High Performance Machine Learning Group

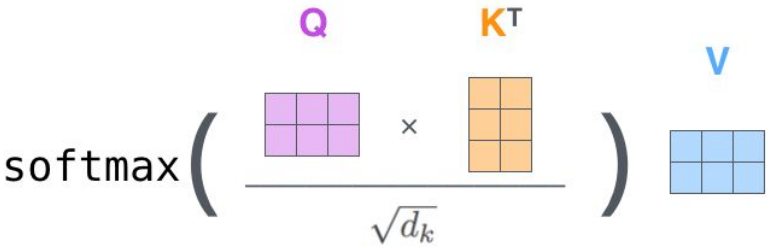
SURF

Self Attention



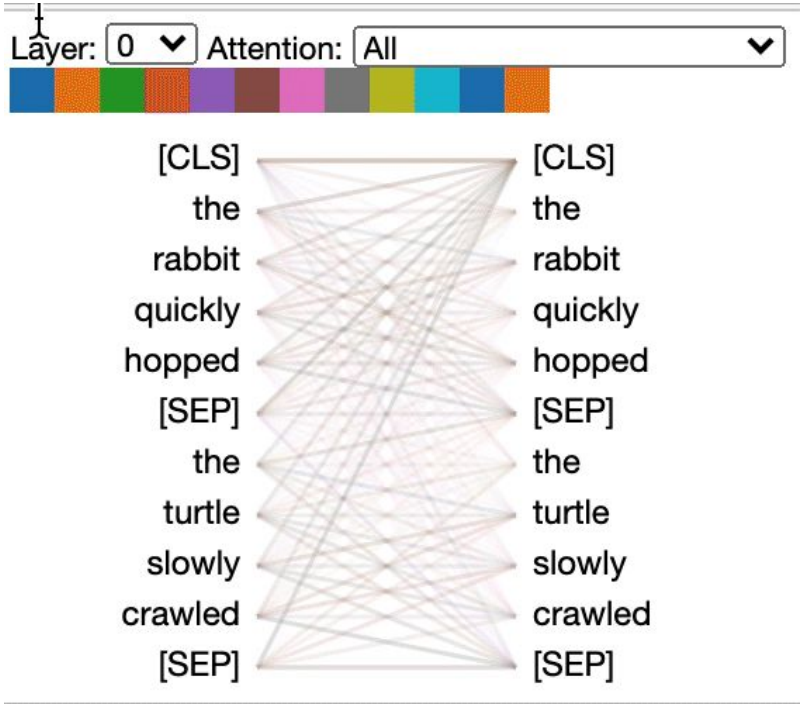
$$\text{softmax}\left(\frac{Q \times K^T}{\sqrt{d_k}}\right) V = Z$$

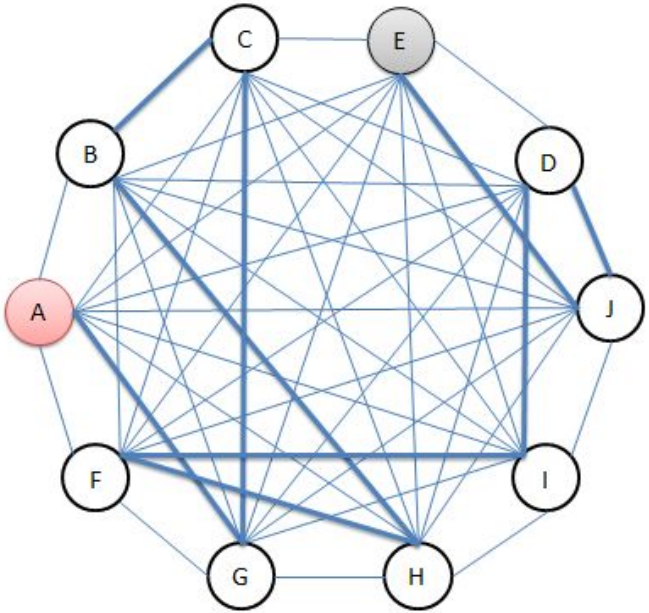
Self Attention



$$\text{softmax}(\frac{QK^T}{\sqrt{d_k}})V$$

Self Attention



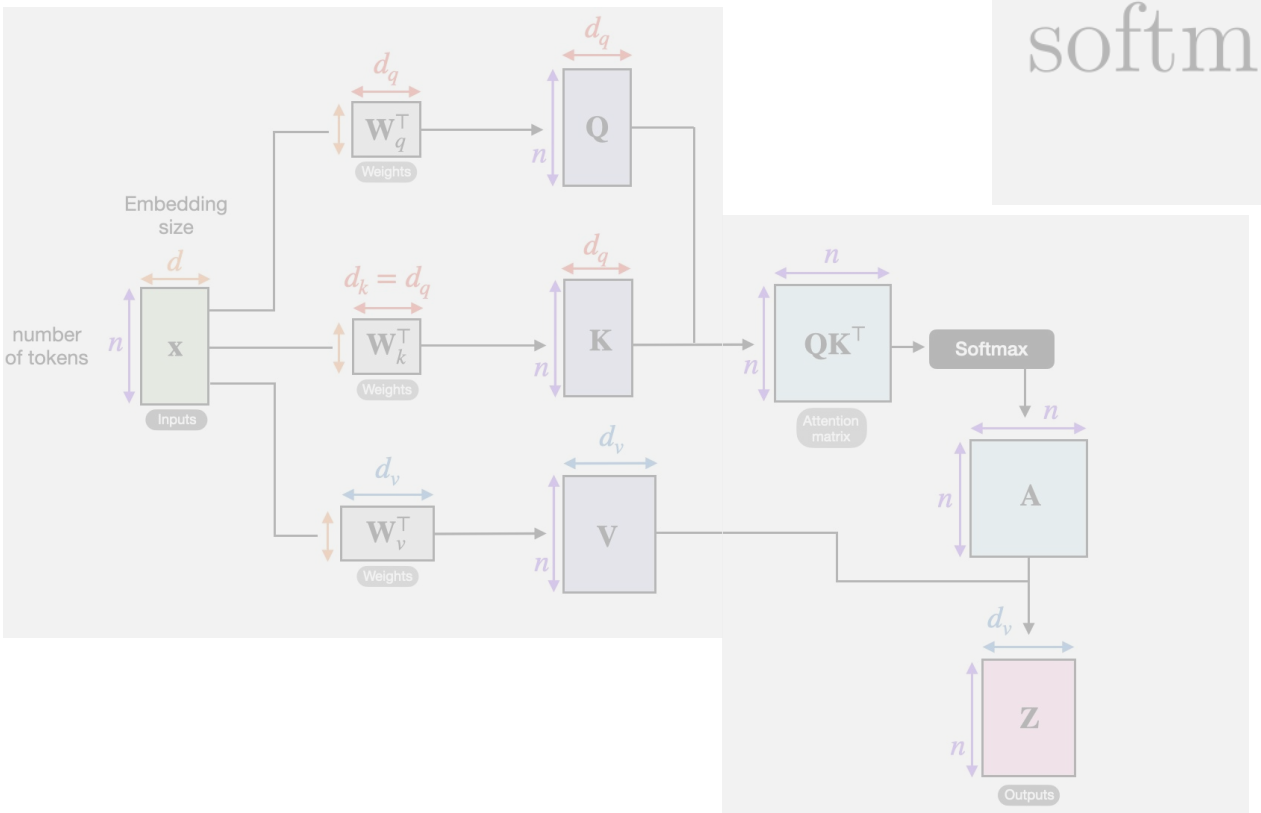


Self Attention

$$\text{softmax}(\frac{QK^T}{\sqrt{d_k}})V$$

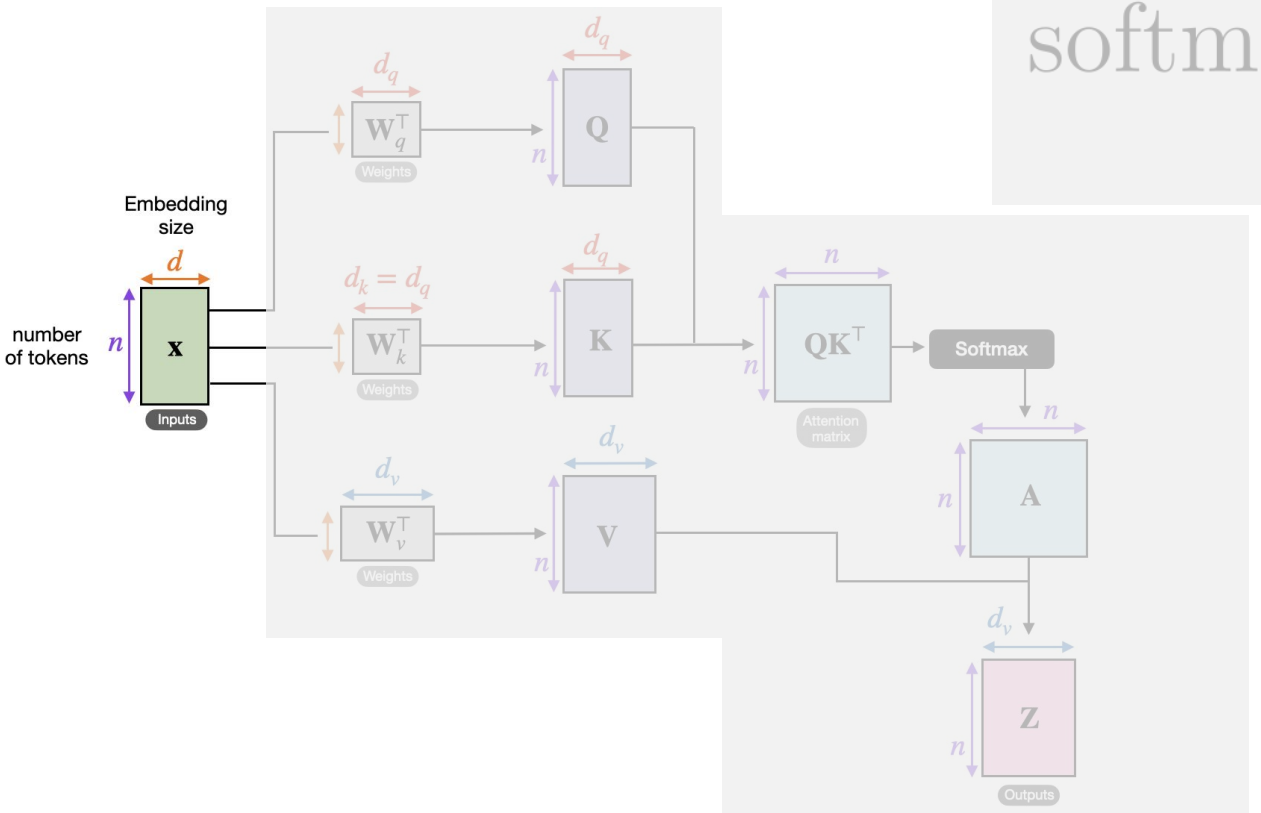
Self Attention

$$\text{softmax}(\frac{QK^T}{\sqrt{d_k}})V$$



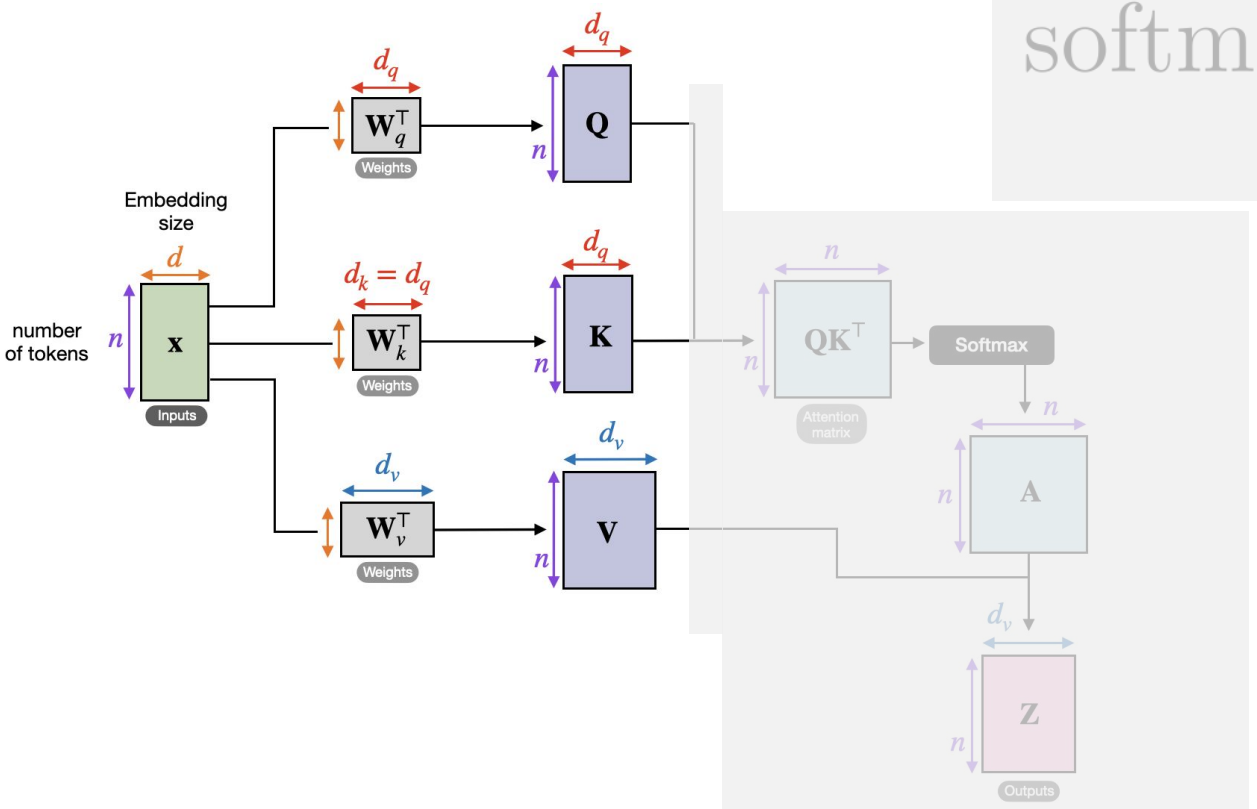
Self Attention

$$\text{softmax}(\frac{QK^T}{\sqrt{d_k}})V$$



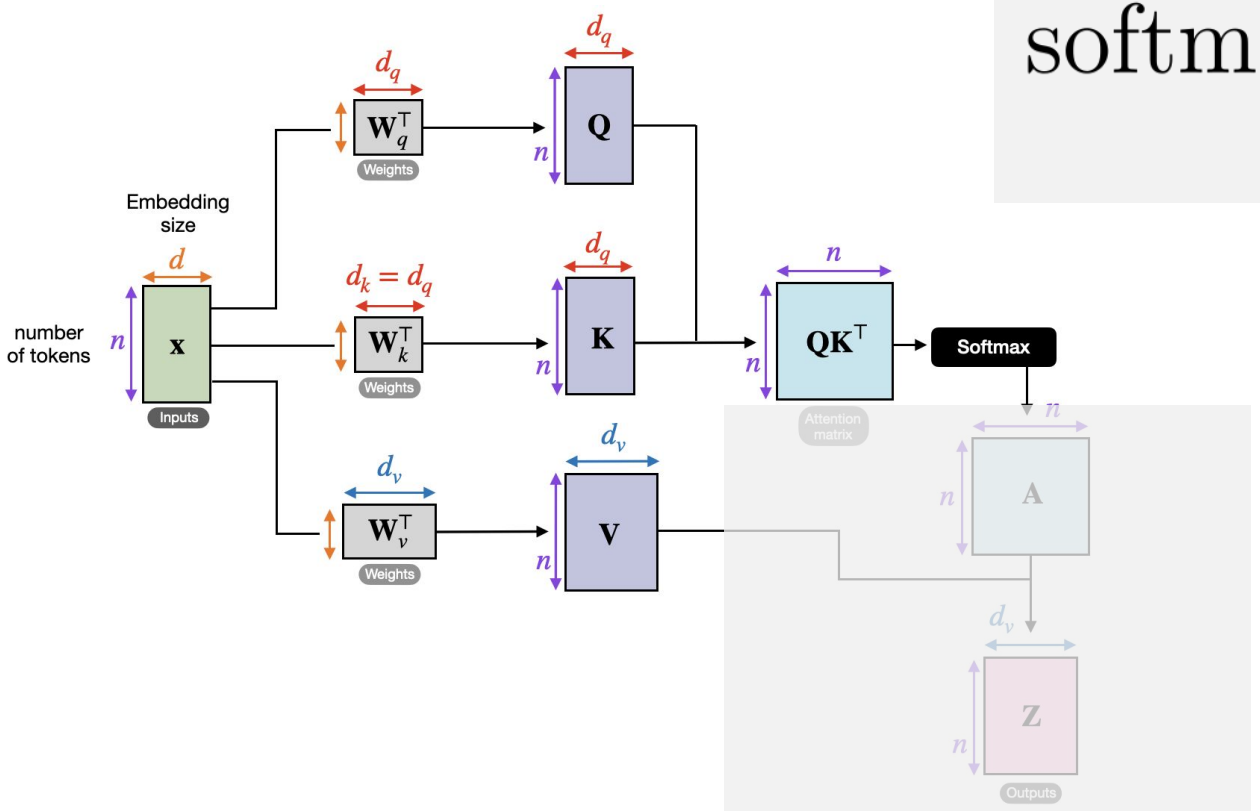
Self Attention

$$\text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V$$



Self Attention

$$\text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V$$



Self Attention

$$\text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V$$

$$Softmax(x) = \frac{e^{x_i}}{\sum_{j=0}^{K} e^{x_j}}$$

Self Attention

$$\text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V$$

$$\frac{e^{x_i}}{\sum_{j=0}^{K} e^{x_j}}$$

Self Attention

$$\text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V$$

$$\begin{bmatrix} 1.3 \\ 5.1 \\ 2.2 \\ 0.7 \\ 1.1 \end{bmatrix}$$

$$\frac{e^{x_i}}{\sum_{j=0}^{K} e^{x_j}}$$

SURF

Self Attention

$$\text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V$$

$$\begin{bmatrix} 1.3 \\ 5.1 \\ 2.2 \\ 0.7 \\ 1.1 \end{bmatrix}$$

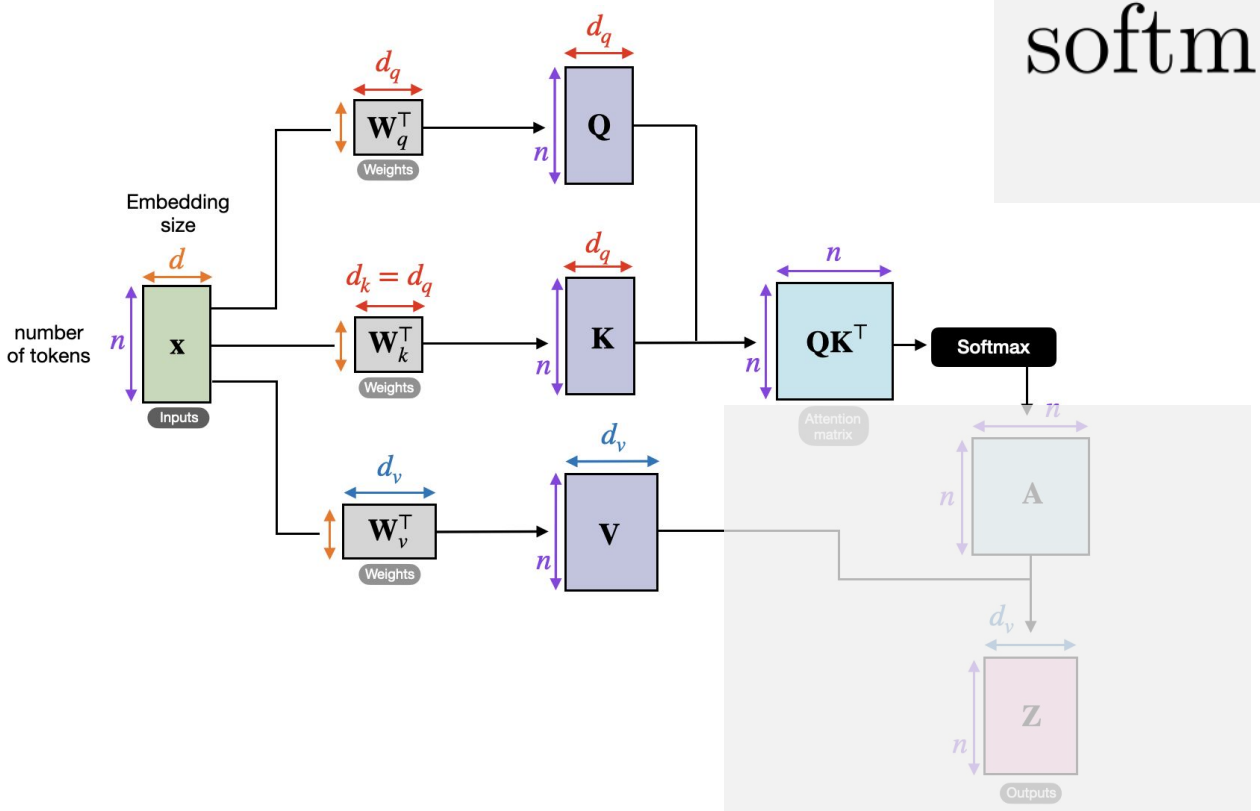
$$\frac{e^{x_i}}{\sum_{j=0}^{K} e^{x_j}}$$

$$\begin{bmatrix} 0.02 \\ 0.90 \\ 0.05 \\ 0.01 \\ 0.02 \end{bmatrix}$$

SURF

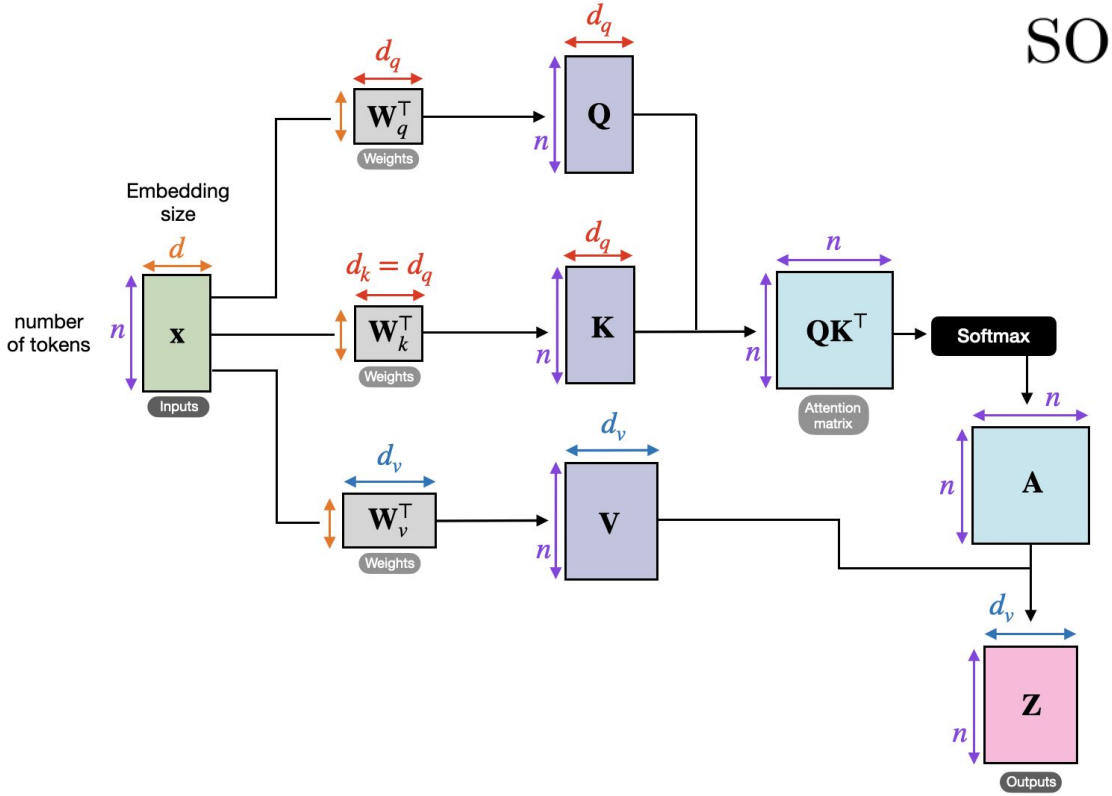
Self Attention

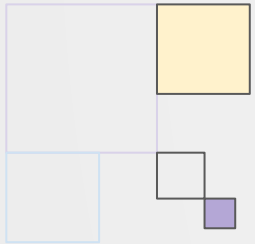
$$\text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V$$



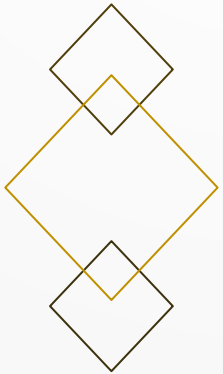
Self Attention

$$\text{softmax}(\frac{QK^T}{\sqrt{d_k}})V$$





Multi-Head Attention



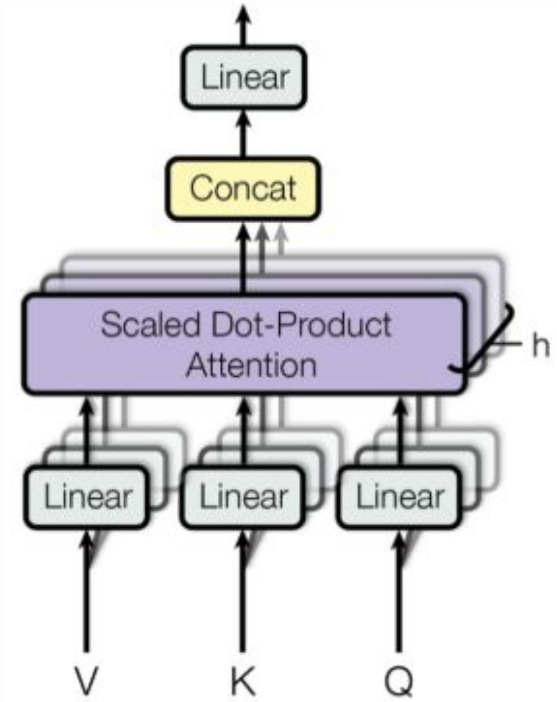
High Performance Machine Learning Group

SURF

The Multi-Headed Beast

Stack Self-Attention **heads** to extract multiple features

Voita, Elena et al. “Analyzing Multi-Head Self-Attention: Specialized Heads Do the Heavy Lifting, the Rest Can Be Pruned.”

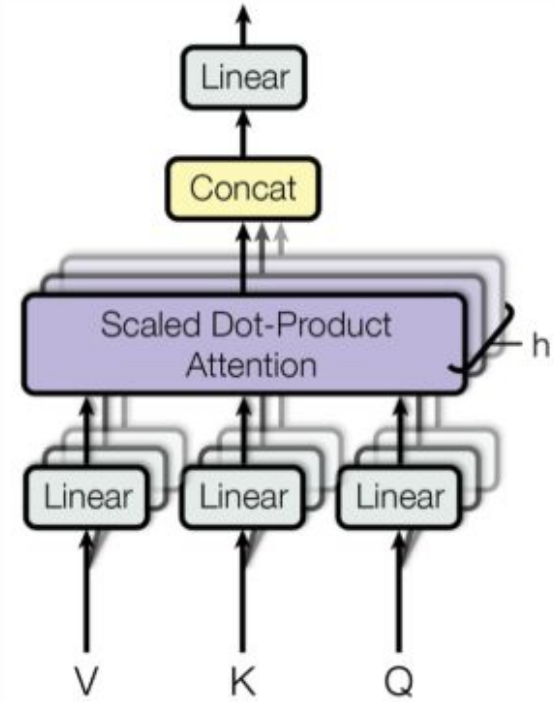


The Multi-Headed Beast

Stack Self-Attention **heads** to extract multiple features

> **Analyzing Multi-Head Self-Attention: Specialized Heads Do the Heavy Lifting, the Rest Can Be Pruned**

Voita, Elena et al. “Analyzing Multi-Head Self-Attention: Specialized Heads Do the Heavy Lifting, the Rest Can Be Pruned.”



The Multi-Headed Beast

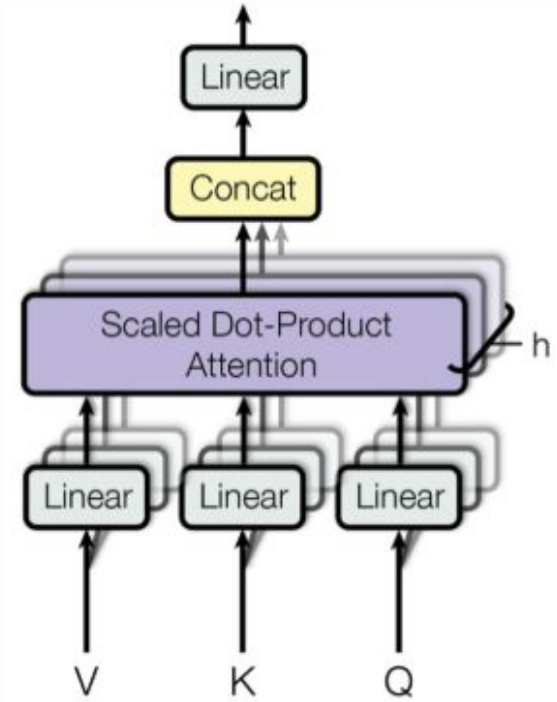
Stack Self-Attention **heads** to extract multiple features

> **Analyzing Multi-Head Self-Attention: Specialized Heads Do the Heavy Lifting, the Rest Can Be Pruned**

Each head could focus on

1. Syntax
2. Positions
3. Rare words
4. Grammatical Errors
5. N-grams

Voita, Elena et al. "Analyzing Multi-Head Self-Attention: Specialized Heads Do the Heavy Lifting, the Rest Can Be Pruned."



The Multi-Headed Beast

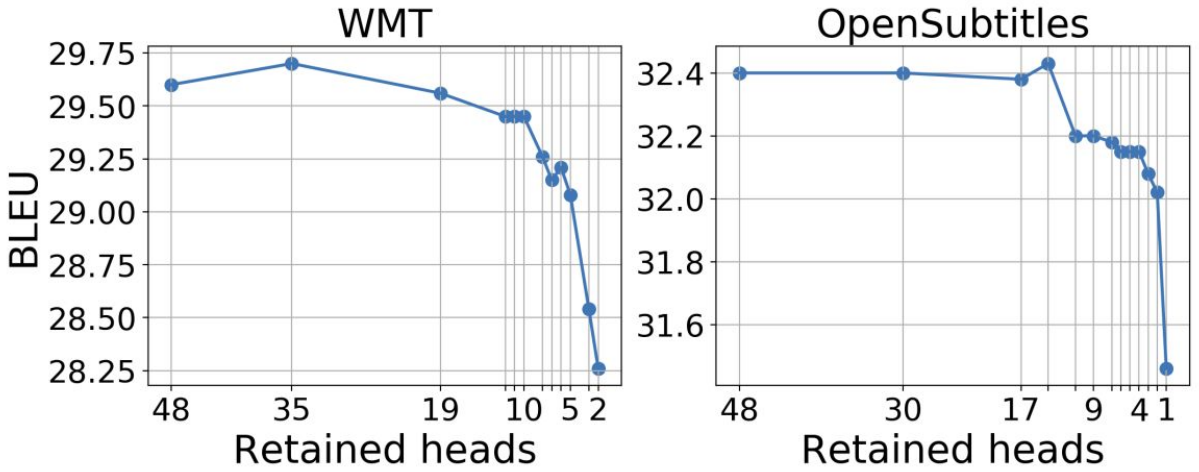
Stack Self-Attention **heads** to extract multiple features

> **Analyzing Multi-Head Self-Attention: Specialized Heads Do the Heavy Lifting, the Rest Can Be Pruned**

Each head could focus on

1. Syntax
2. Positions
3. Rare words
4. Grammatical Errors
5. N-grams

Voita, Elena et al. “Analyzing Multi-Head Self-Attention: Specialized Heads Do the Heavy Lifting, the Rest Can Be Pruned.”

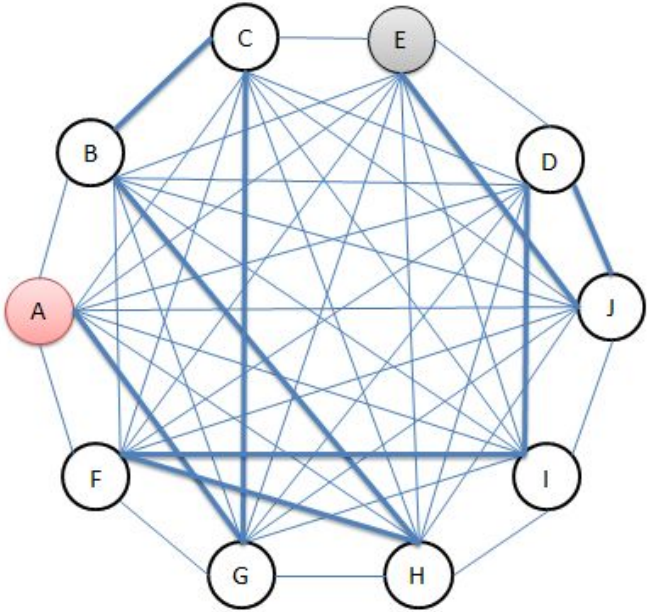


The Multi-Headed Beast

Stack Self-Attention **heads** to extract multiple features

> **Analyzing Multi-Head Self-Attention: Specialized Heads Do the Heavy Lifting, the Rest Can Be Pruned**

Transformers are **Permutation Invariant!**



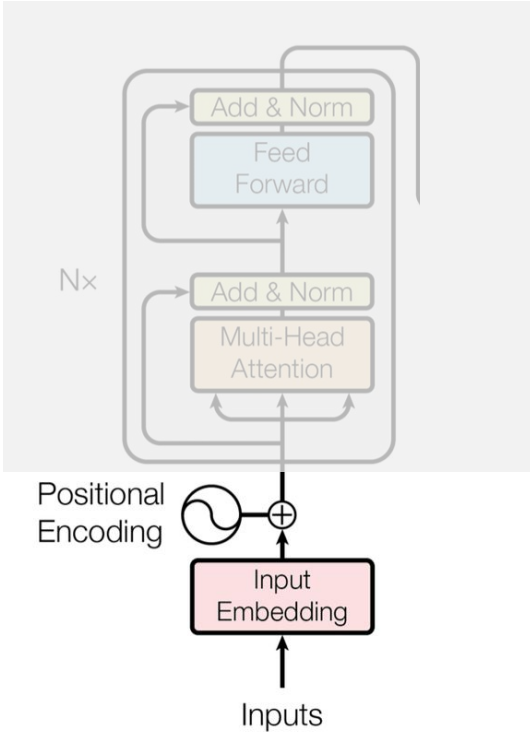
SURF

The Multi-Headed Beast

Stack Self-Attention **heads** to extract multiple features

> **Analyzing Multi-Head Self-Attention: Specialized Heads Do the Heavy Lifting, the Rest Can Be Pruned**

Transformers are **Permutation Invariant!**

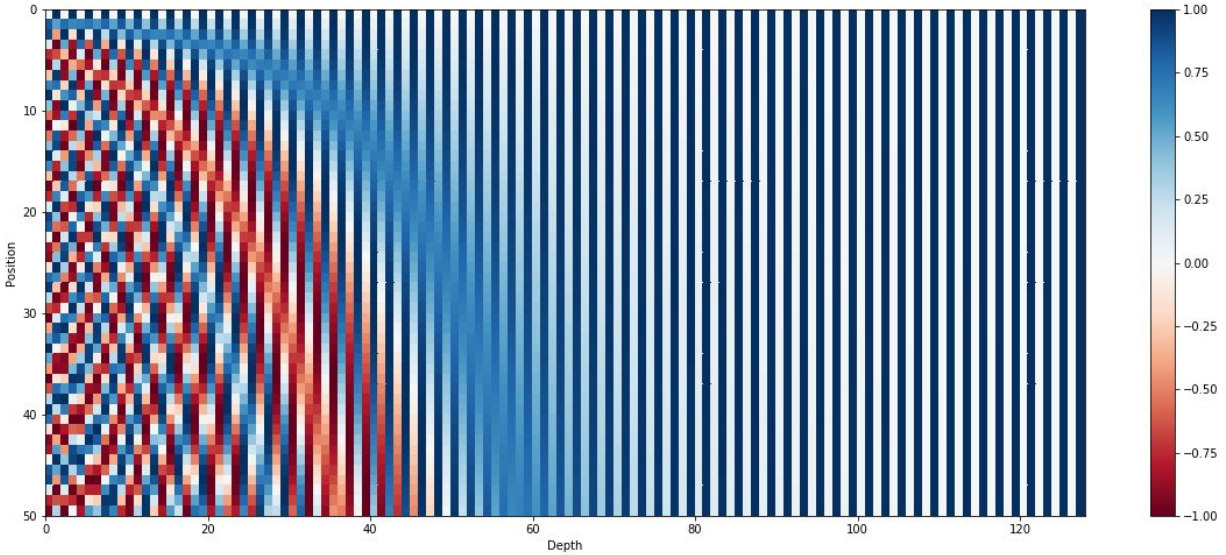


The Multi-Headed Beast

Stack Self-Attention **heads** to extract multiple features

> **Analyzing Multi-Head Self-Attention: Specialized Heads Do the Heavy Lifting, the Rest Can Be Pruned**

Transformers are **Permutation Invariant!**

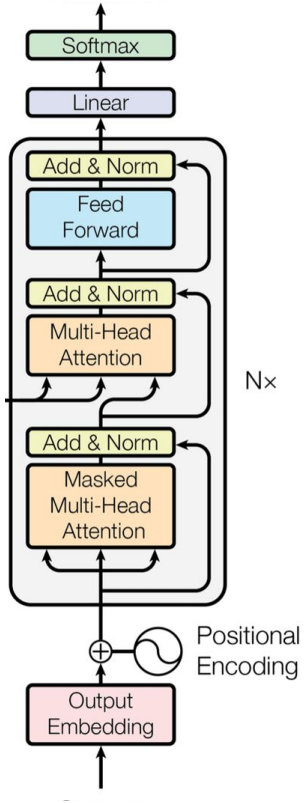


The Multi-Headed Beast

Stack Self-Attention **heads** to extract multiple features

> **Analyzing Multi-Head Self-Attention: Specialized Heads Do the Heavy Lifting, the Rest Can Be Pruned**

Transformers are **Permutation Invariant!**

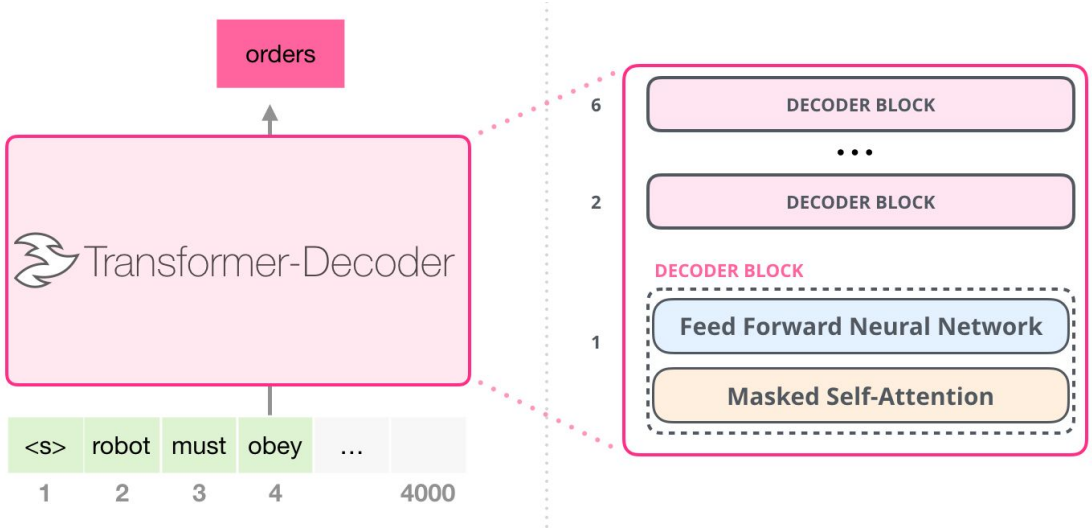


The Multi-Headed Beast

Stack Self-Attention **heads** to extract multiple features

> **Analyzing Multi-Head Self-Attention: Specialized Heads Do the Heavy Lifting, the Rest Can Be Pruned**

Transformers are **Permutation Invariant!**

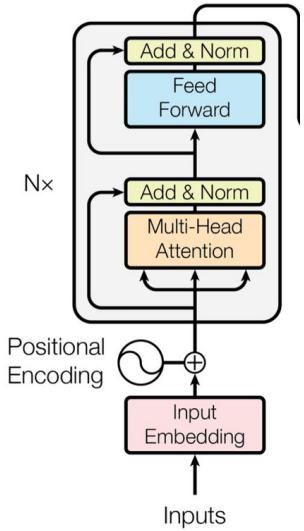


SURF

Dissection of the Transformer

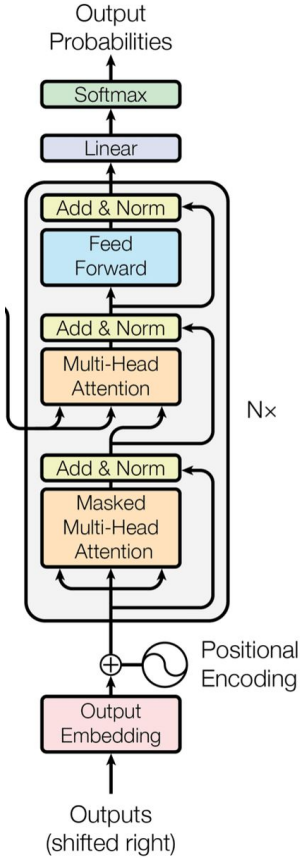
The Encoder

BERT

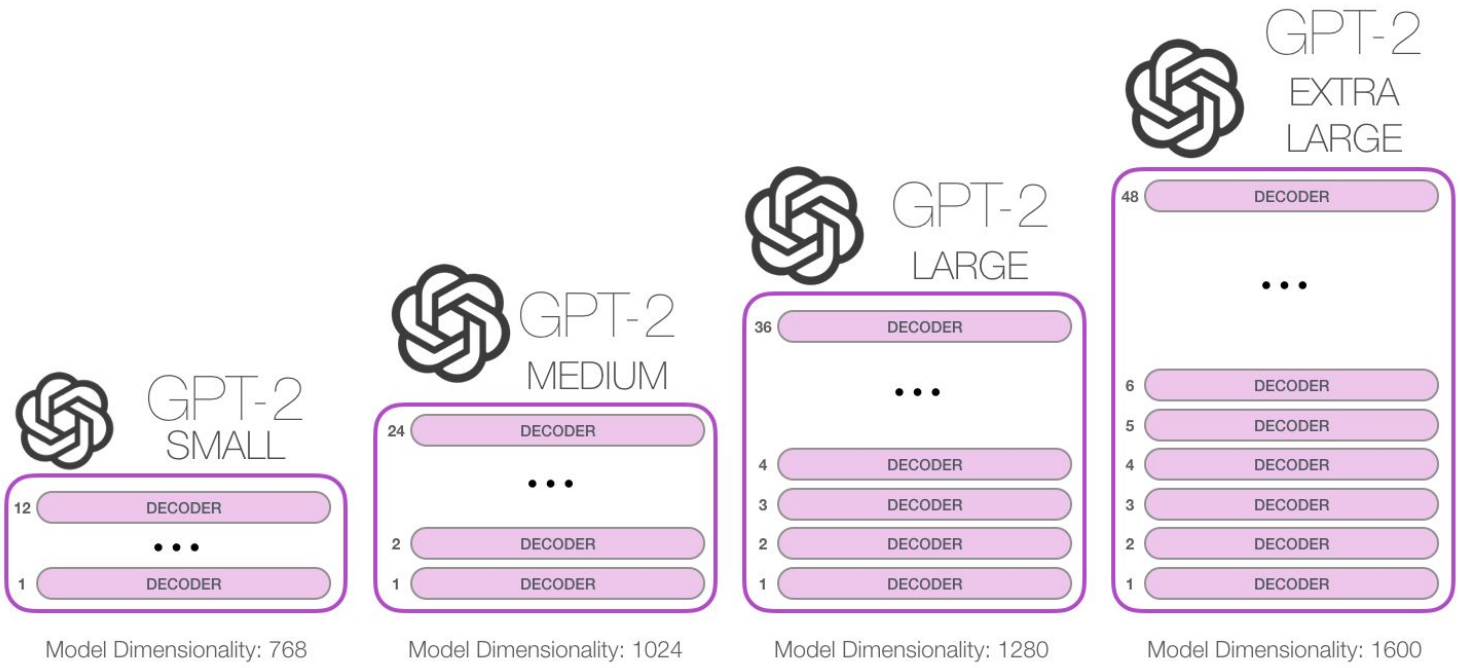


The Decoder

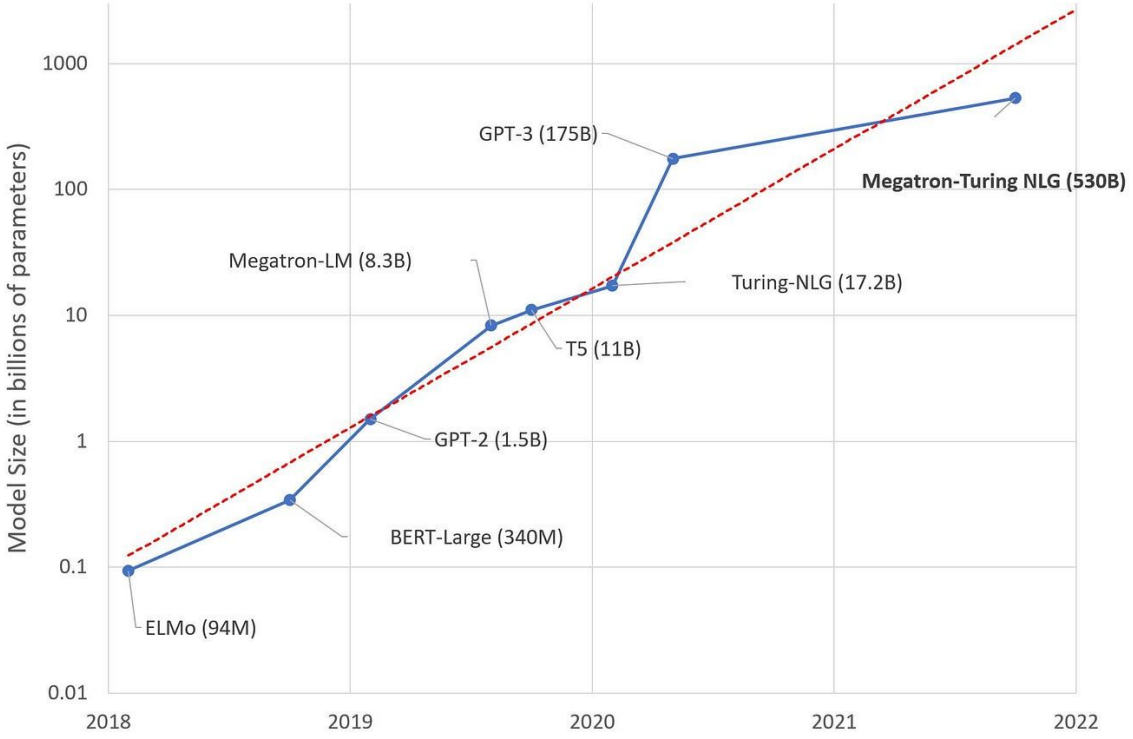
GPT

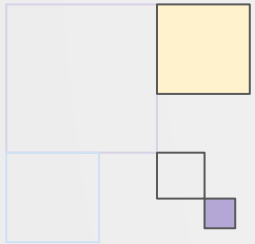


Generative Pre-Trained Transformers

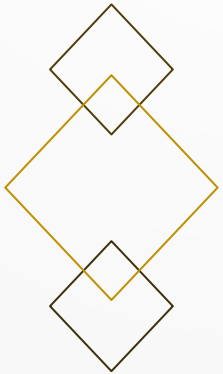


Generative Pre-Trained Transformers





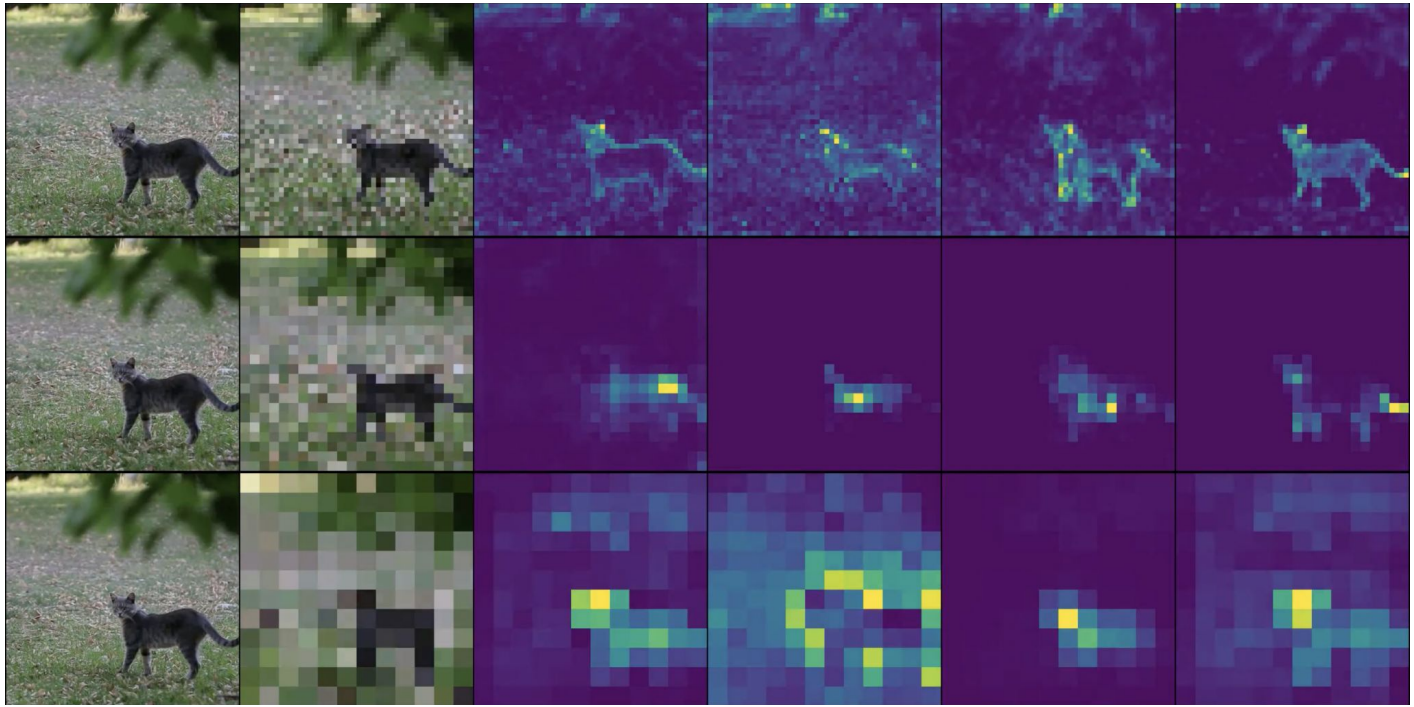
Transformer Cases



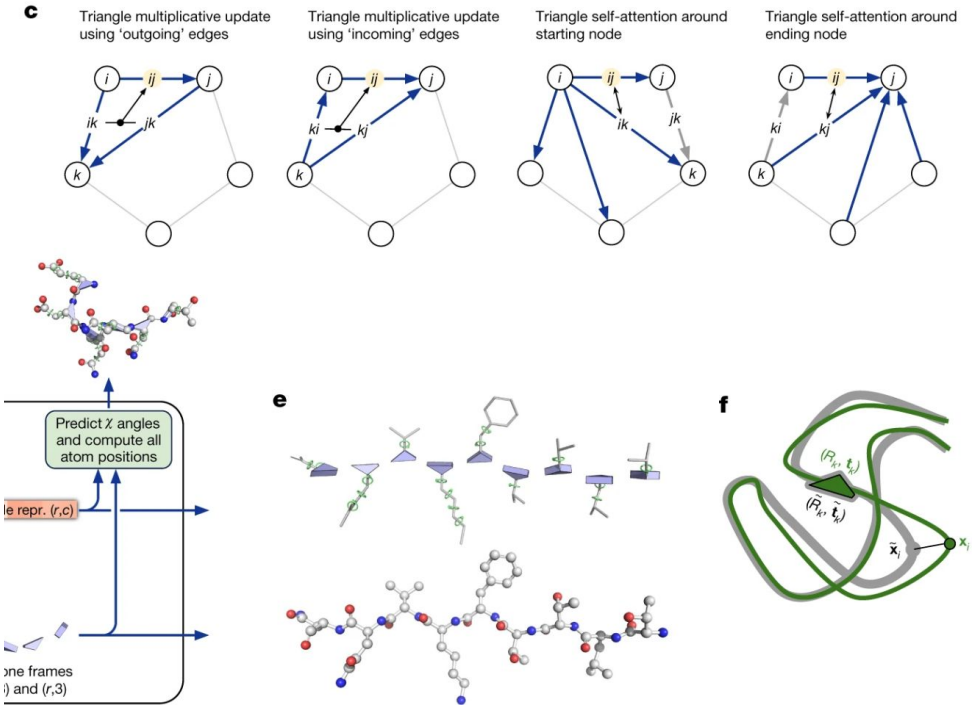
High Performance Machine Learning Group

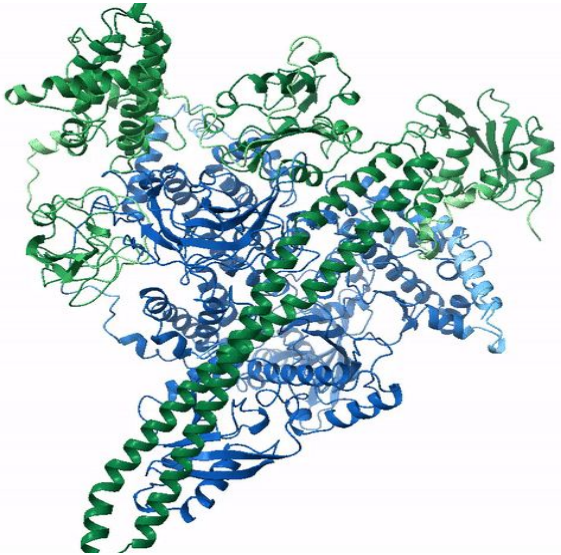
SURF

Transformers for other modalities: ViT



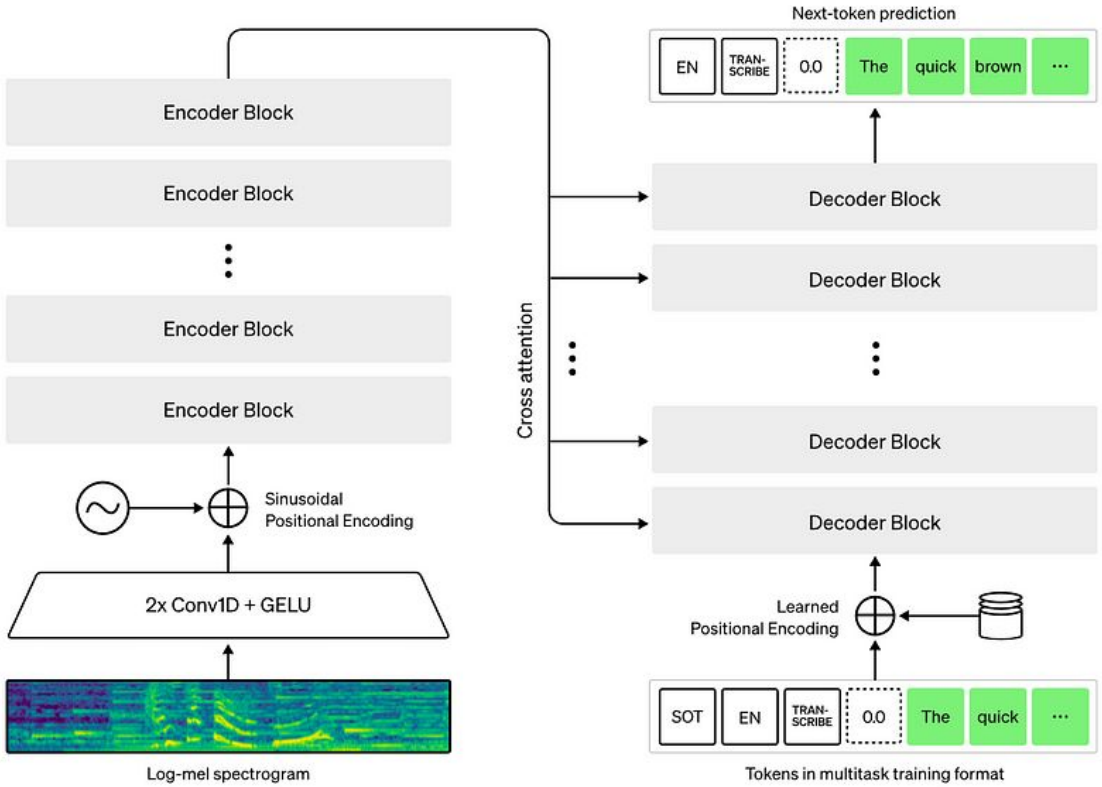
Transformers for other modalities: EvoFormer





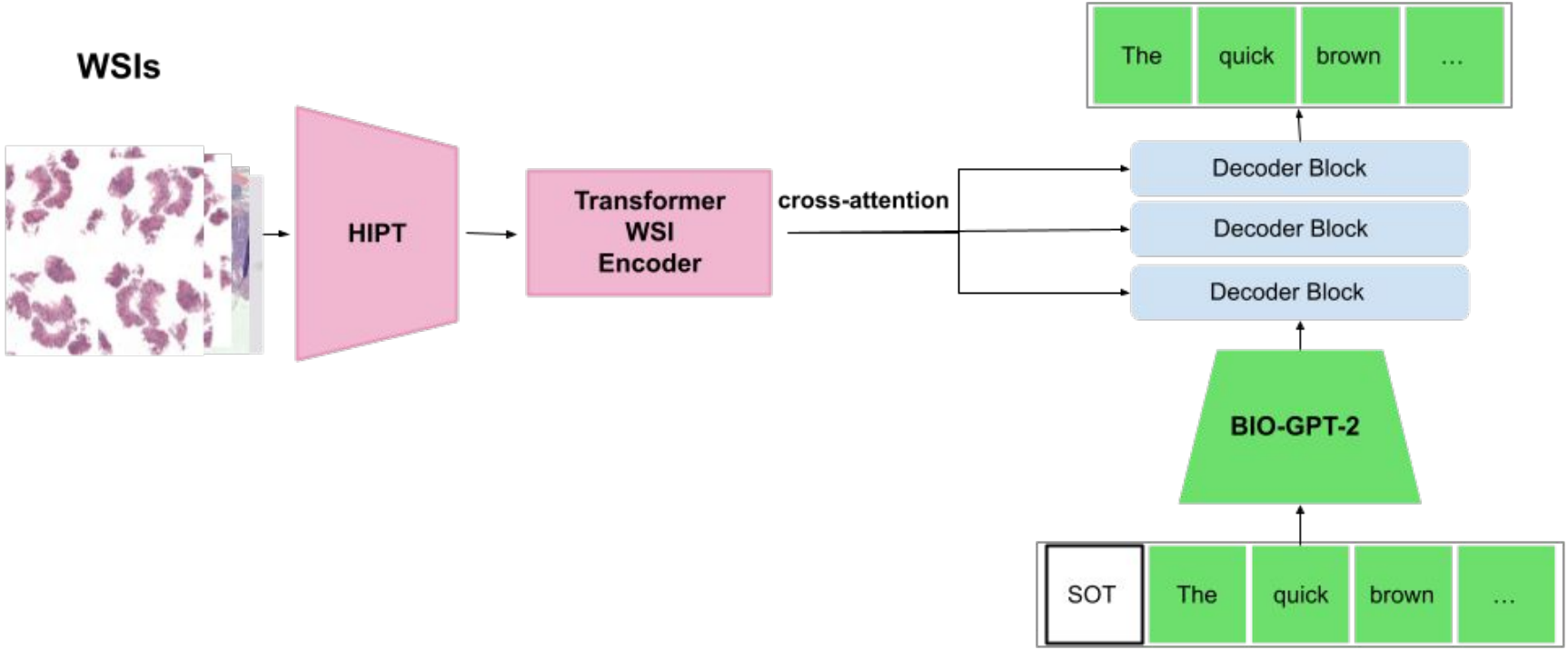
https://www.nature.com/articles/s41586-021-03819-2

Transformers for other modalities: Whisper

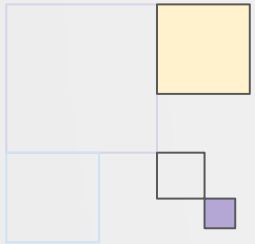


https://openai.com/research/whisper

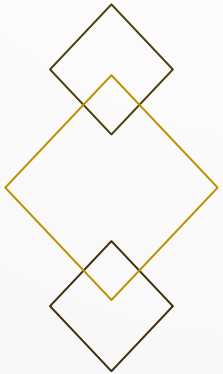
Transformers for other modalities: Multi-Modal Histopathology



Caption generation from histopathology whole-slide images using pre-trained transformers
BC Guevara, N Marini, S Marchesin, W Aswolinskiy, et al.



Transformer Interpretations

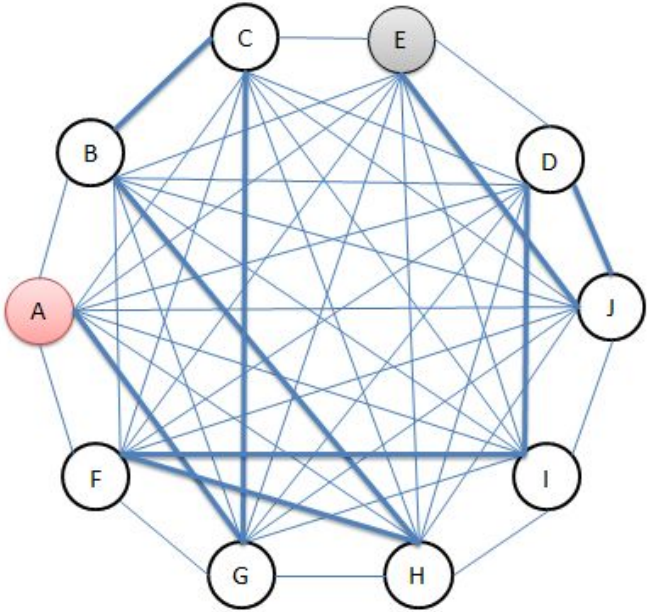


High Performance Machine Learning Group

SURF

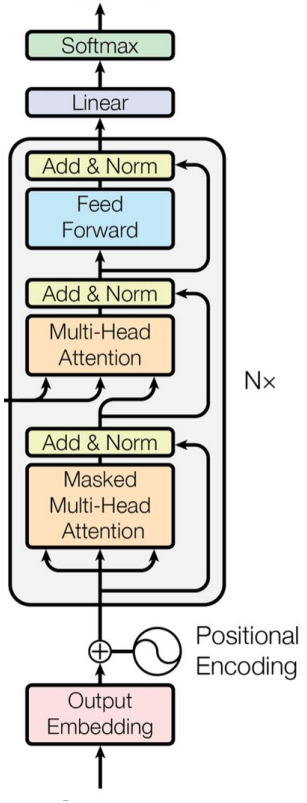
Fully Connected Weighted Graph

Lower the energy state that the network should ***remember***



Modify the input layer by layer

Lower the energy state that the network should ***remember***



https://transformer-circuits.pub/2021/framework/index.html

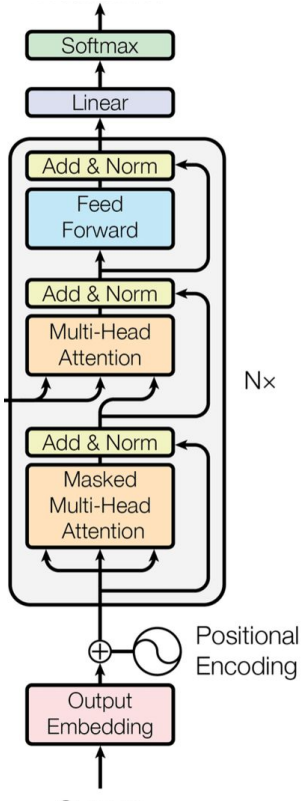
Belrose, Nora et al. “Eliciting Latent Predictions from Transformers with the Tuned Lens.”

Modify the input layer by layer

Lower the energy state that the network should ***remember***

Attention heads output a result added to the **residual stream.**

All components of a transformer **communicate** by **reading and writing** to different subspaces of the stream



https://transformer-circuits.pub/2021/framework/index.html

Belrose, Nora et al. “Eliciting Latent Predictions from Transformers with the Tuned Lens.”

Modify the input layer by layer

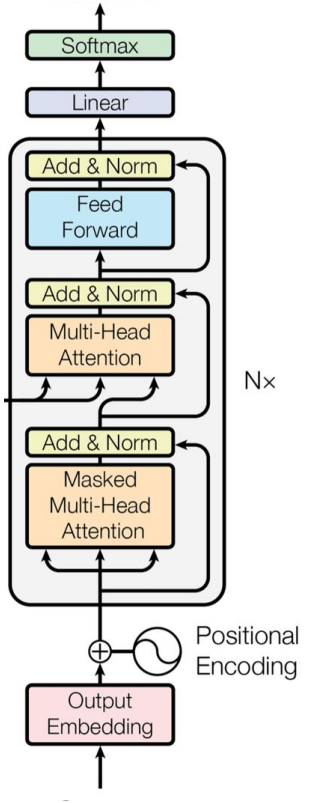
Lower the energy state that the network should ***remember***

Attention heads output a result added to the **residual stream.**

All components of a transformer **communicate** by **reading and writing** to different subspaces of the stream

Conveyor belt principle

$$(T)$$



https://transformer-circuits.pub/2021/framework/index.html

Belrose, Nora et al. “Eliciting Latent Predictions from Transformers with the Tuned Lens.”

Modify the input layer by layer

Lower the energy state that the network should ***remember***

Attention heads output a result added to the **residual stream.**

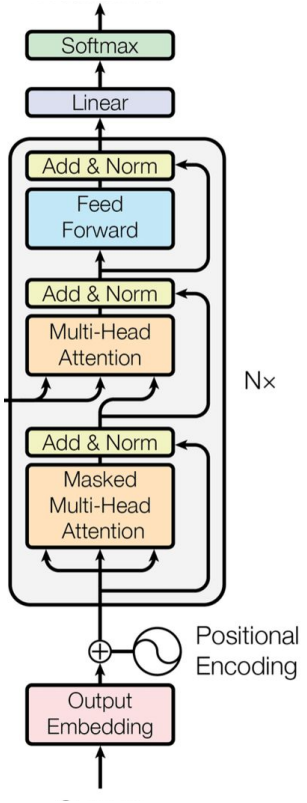
All components of a transformer **communicate** by **reading and writing** to different subspaces of the stream

Conveyor belt principle

$$(T, D)$$

$$(T, D)$$

$$(T)$$



https://transformer-circuits.pub/2021/framework/index.html

Belrose, Nora et al. “Eliciting Latent Predictions from Transformers with the Tuned Lens.”

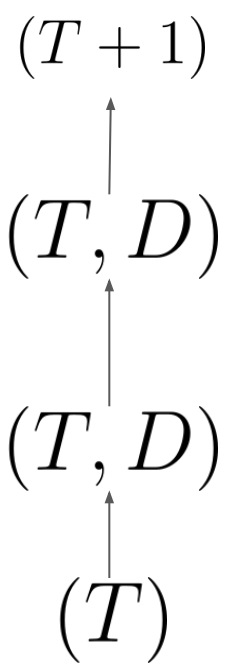
Modify the input layer by layer

Lower the energy state that the network should ***remember***

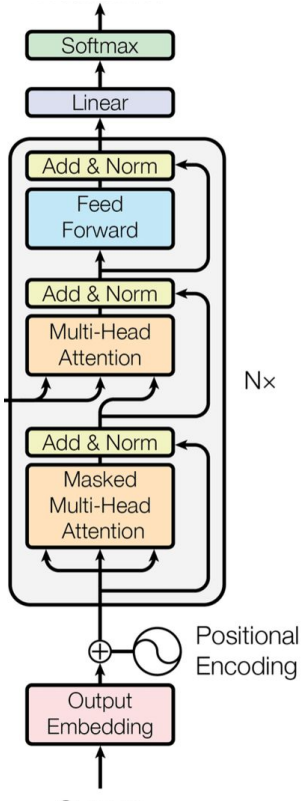
Attention heads output a result added to the **residual stream.**

All components of a transformer **communicate** by **reading and writing** to different subspaces of the stream

Conveyor belt principle



https://transformer-circuits.pub/2021/framework/index.html



Belrose, Nora et al. "Eliciting Latent Predictions from Transformers with the Tuned Lens."

SURF

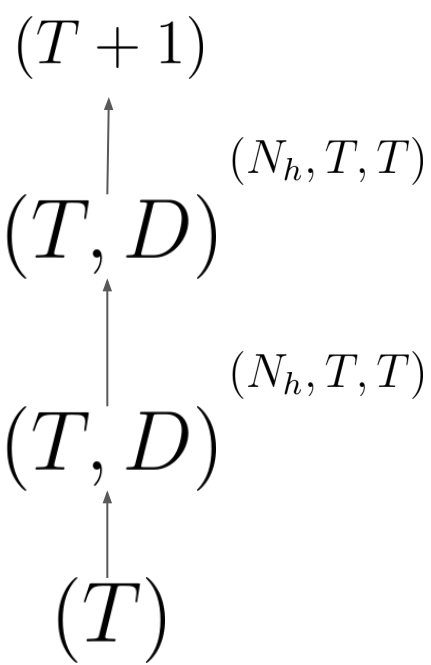
Modify the input layer by layer

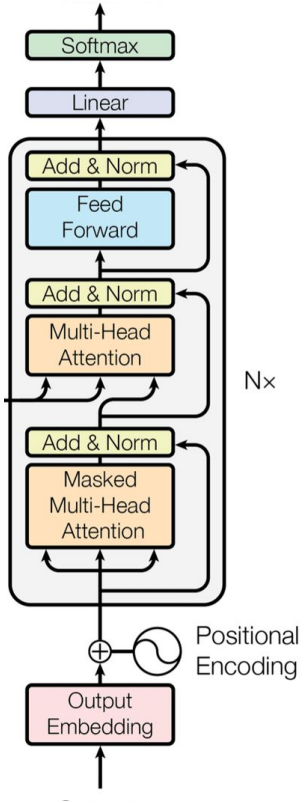
Lower the energy state that the network should ***remember***

Attention heads output a result added to the **residual stream.**

All components of a transformer **communicate** by **reading and writing** to different subspaces of the stream

Conveyor belt principle





https://transformer-circuits.pub/2021/framework/index.html

Belrose, Nora et al. “Eliciting Latent Predictions from Transformers with the Tuned Lens.”

SURF

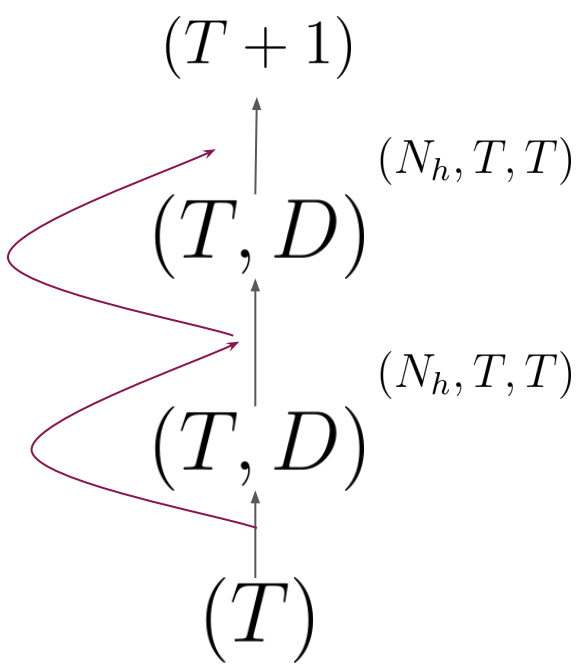
Modify the input layer by layer

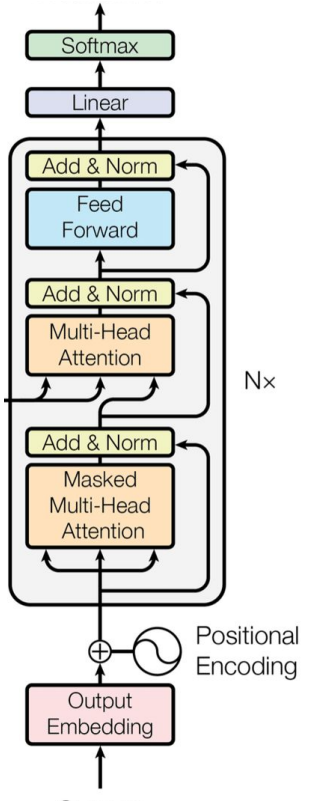
Lower the energy state that the network should ***remember***

Attention heads output a result added to the **residual stream.**

All components of a transformer **communicate** by **reading and writing** to different subspaces of the stream

Conveyor belt principle





https://transformer-circuits.pub/2021/framework/index.html

Belrose, Nora et al. "Eliciting Latent Predictions from Transformers with the Tuned Lens."

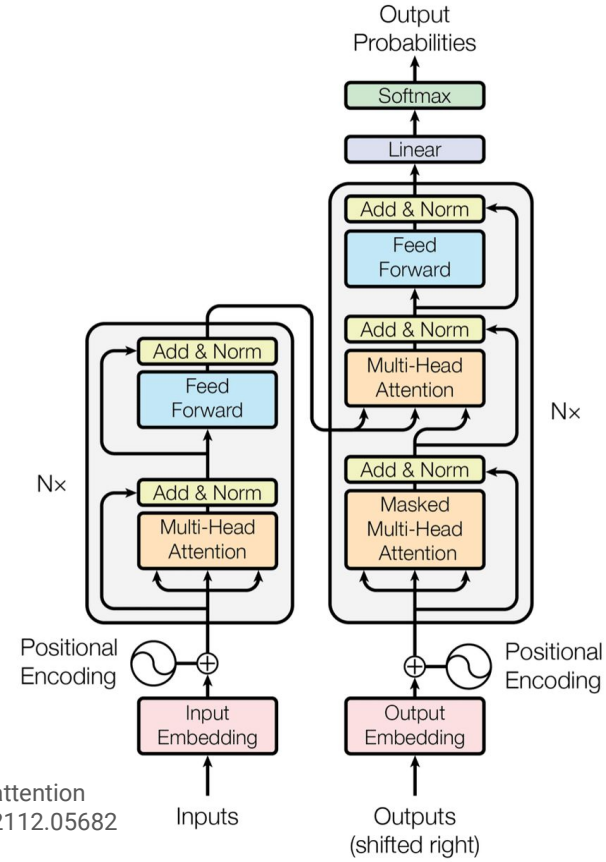
SURF

Final Words

Transformer training is quite slow

We need a lot of data

And **a lot** of parameters



Rabe, Markus N. and Charles Staats. "Self-attention Does Not Need O(n2) Memory." *ArXiv* abs/2112.05682 (2021): n. pag.

Final Words

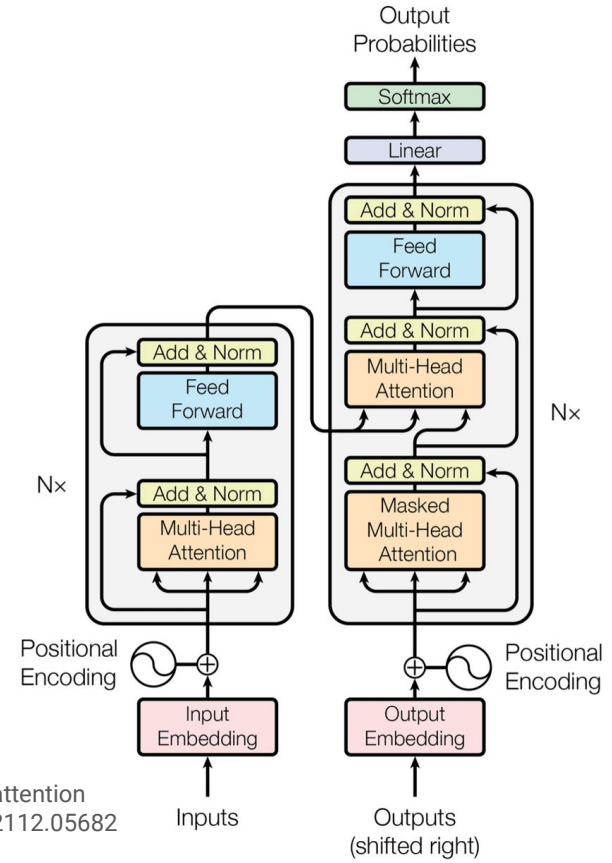
Transformer training is quite slow

We need a lot of data

And **a lot** of parameters

$O(N^2)$ in compute and memory

$$O(N^2)$$



Rabe, Markus N. and Charles Staats. "Self-attention Does Not Need O(n2) Memory." *ArXiv* abs/2112.05682 (2021): n. pag.

SURF

Final Words

Transformer training is quite slow

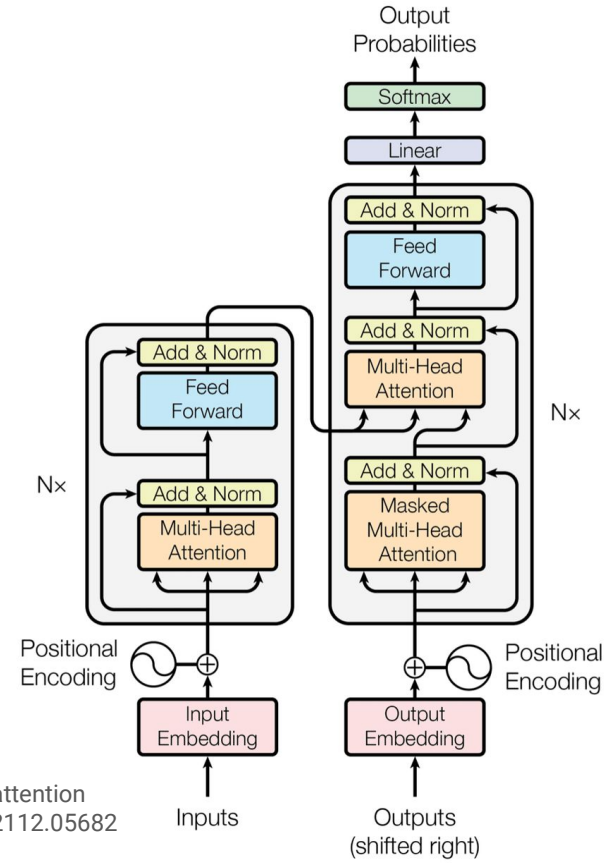
We need a lot of data

And **a lot** of parameters

$O(N^2)$ in compute and memory

But, does not have to be $O(N^2)$
In memory!

$$O(N^2)$$



Rabe, Markus N. and Charles Staats. "Self-attention Does Not Need O(n2) Memory." *ArXiv* abs/2112.05682 (2021): n. pag.

Final Words

Transformer training is quite slow

We need a lot of data

And **a lot** of parameters

$O(N^2)$ in compute and memory

But, does not have to be $O(N^2)$ In memory!

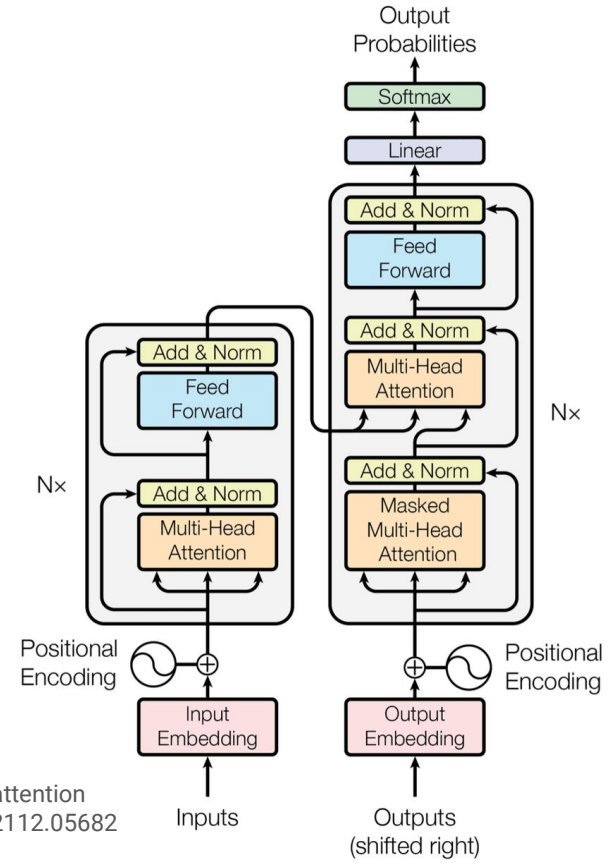
We can do $O(\sqrt{N})$

Partition the attention and compute it **chunk by chunk**

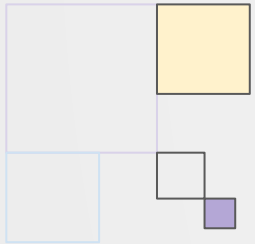
Uses more compute but orders of magnitude less memory usage

$$O(N^2)$$

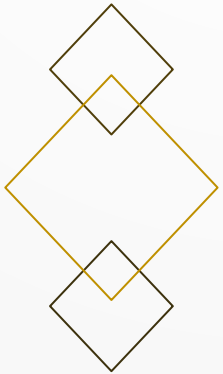
$$O(\sqrt{N})$$



Rabe, Markus N. and Charles Staats. "Self-attention Does Not Need O(n2) Memory." *ArXiv* abs/2112.05682 (2021): n. pag.



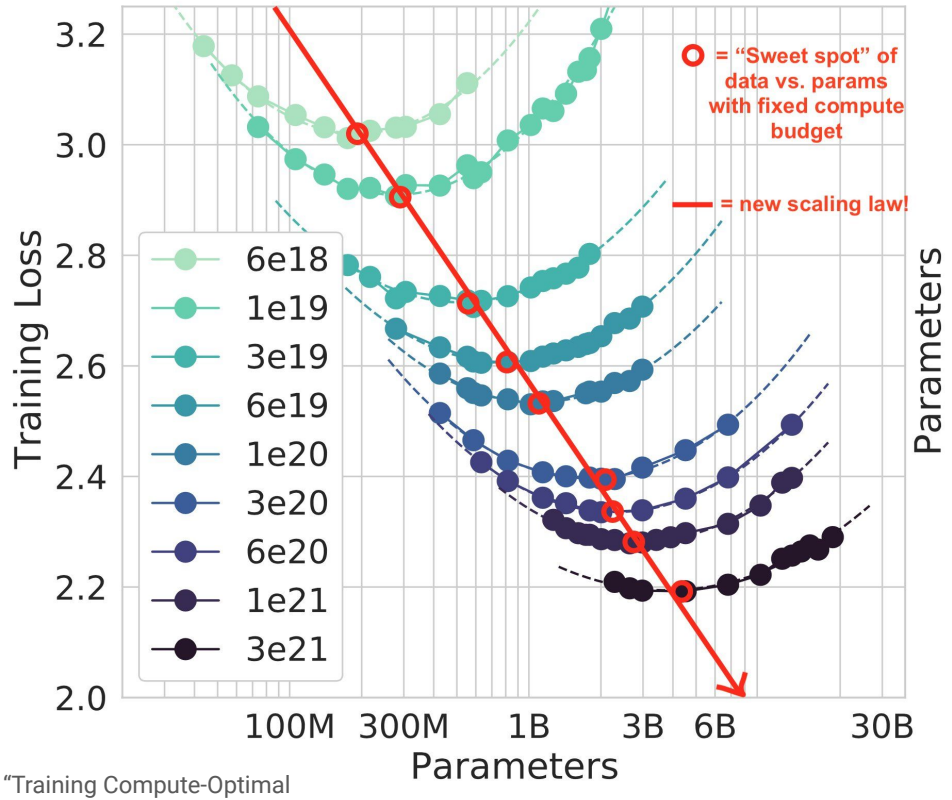
Chinchilla Scaling Laws



High Performance Machine Learning Group

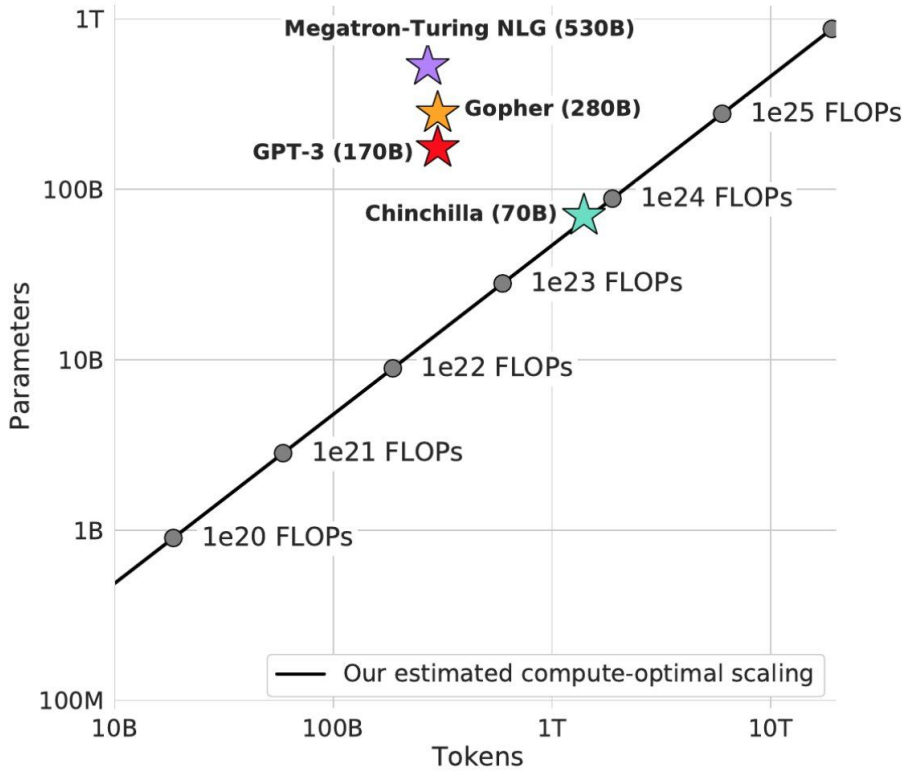
SURF

Chinchilla Scaling Laws

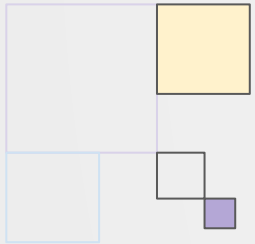


Hoffmann, Jordan et al. “Training Compute-Optimal Large Language Models.” *ArXiv* abs/2203.15556 (2022): n. pag.

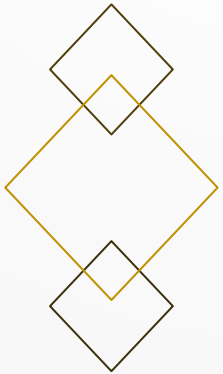
Chinchilla Scaling Laws



Hoffmann, Jordan et al. "Training Compute-Optimal Large Language Models." *ArXiv* abs/2203.15556 (2022): n. pag.

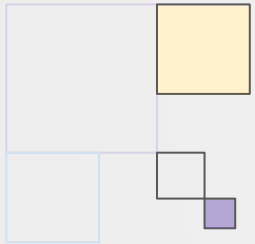


Transformer Demo

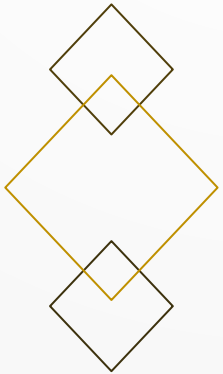


High Performance Machine Learning Group

SURF



Scaling Transformers



High Performance Machine Learning Group

SURF

Whose Revolution?

01.

Data Amount & Availability

Data generated by our devices continuously

02.

Compute Power

Easier available in the cloud
Frameworks to work with accelerated hardware

03.

Research & Funds

More funds to researchers to experiment without limits

04.

More Parameters

Make networks bigger
Feed more information

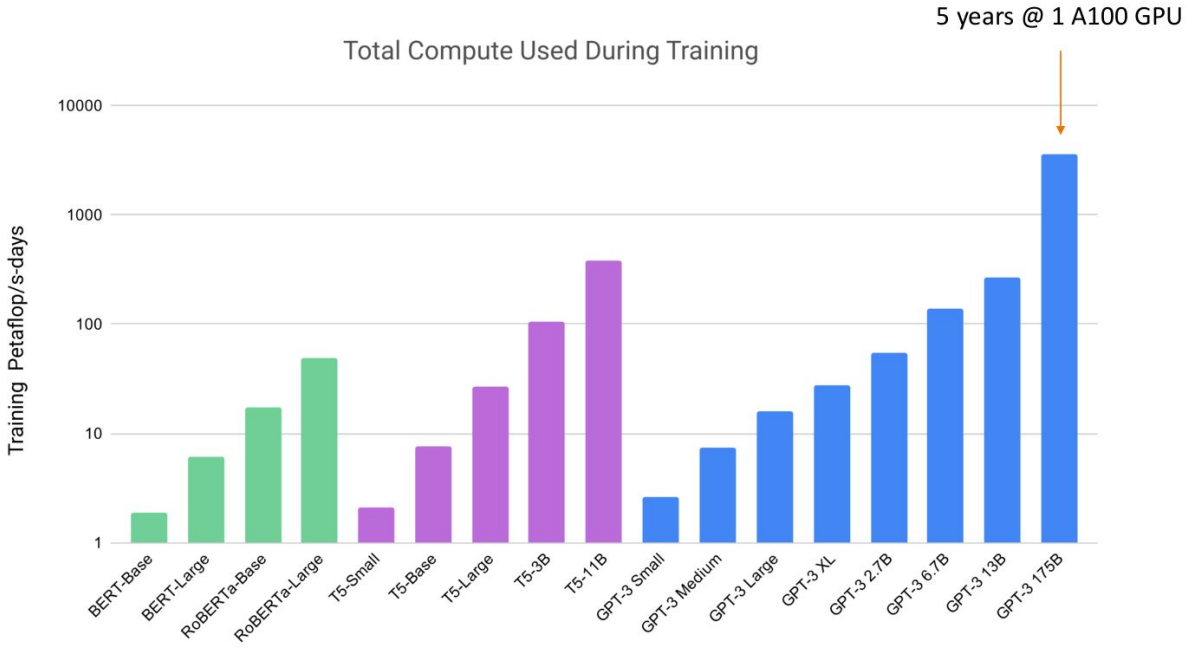
Transformer Explosion

GPT-3 Trained on ~350 Billion tokens

GPT-3 has 175B parameters!

OpenAI used a cluster of V100 GPUs

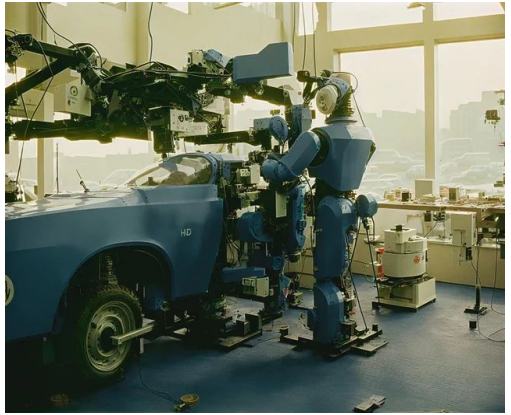
~Newest iterations trained on 1024 A100s for 34 days

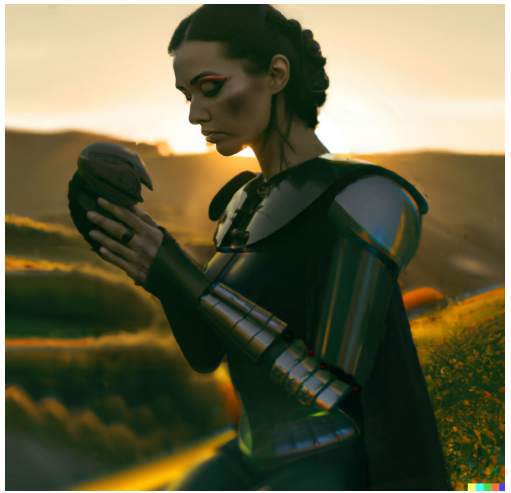


Stable Diffusion & DALLE

Trained on 500M+ image and text pairs

Stability AI used over 4000 A100 GPUs





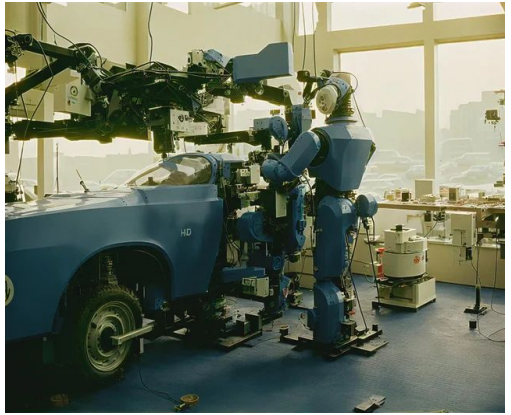
Stable Diffusion & DALLE

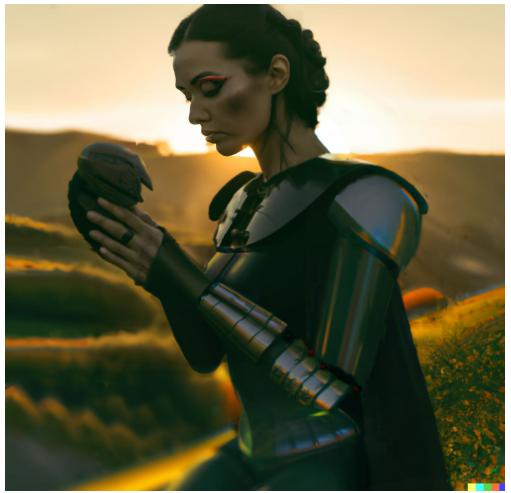
Trained on 500M+ image and text pairs

Stability AI used over 4000 A100 GPUs

Training Large Language Models can consume a few hundreds or even thousands of MWh of energy

A GPU can consume around 400W at peak or more





Stable Diffusion & DALLE

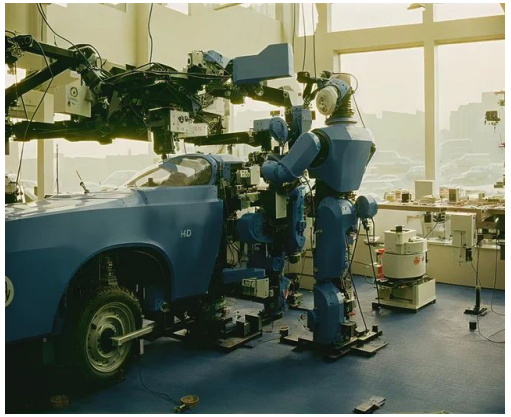
Trained on 500M+ image and text pairs

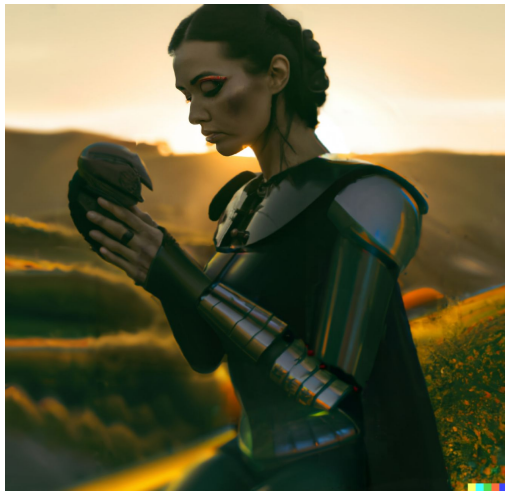
Stability AI used over 4000 A100 GPUs

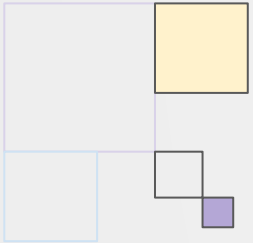
Training Large Language Models can consume a few hundreds or even thousands of MWh of energy

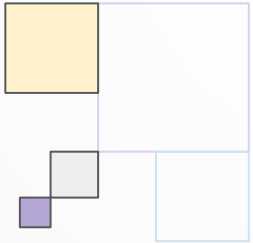
A GPU can consume around 400W at peak or more

Brain is vastly more efficient consuming around 10W



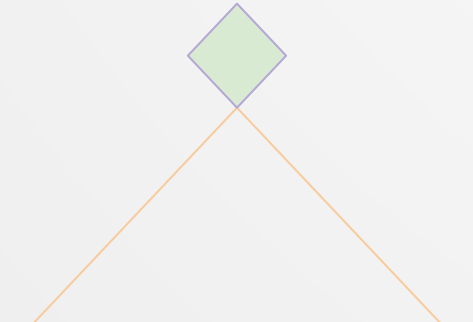






Fin

Introduction Series



Bryan Cardenas
Thomas van Osch

SURF