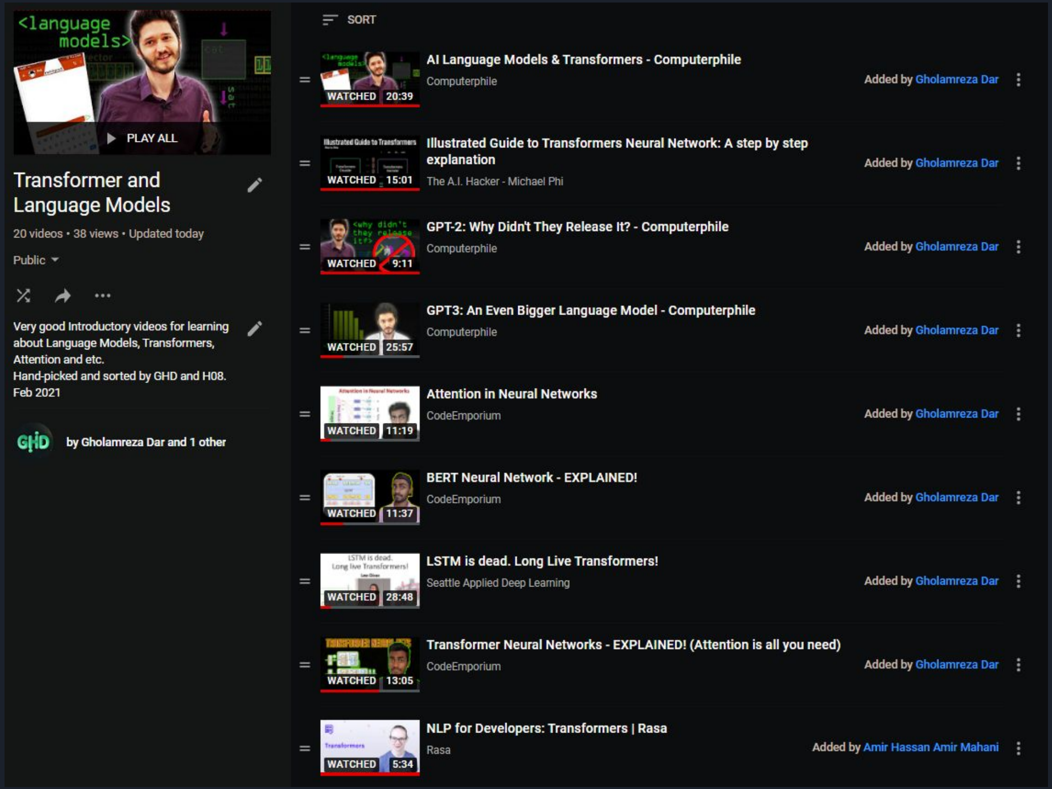
Introduction to Transformers

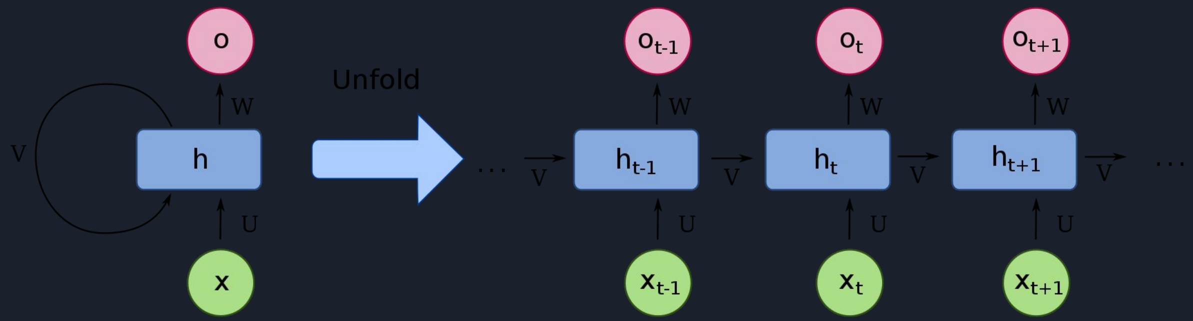
Gholamreza Dar, Amirhassan Amirmahani

YouTube Playlist



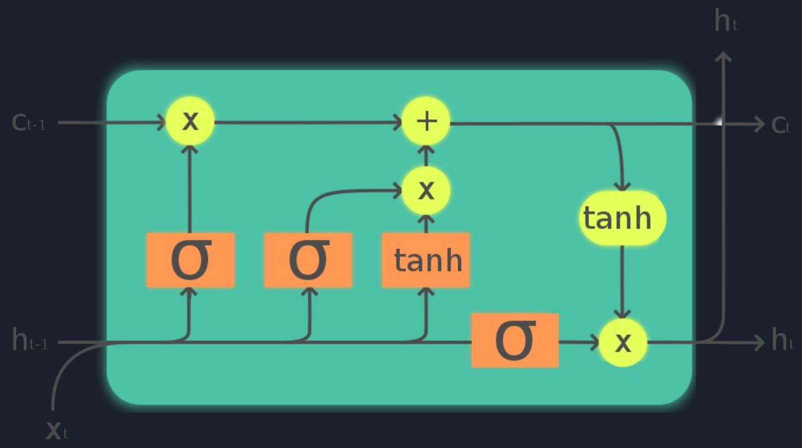
Recurrent neural network (1986)

- Language Translation
- Text Summarization
- Next Sentence Prediction



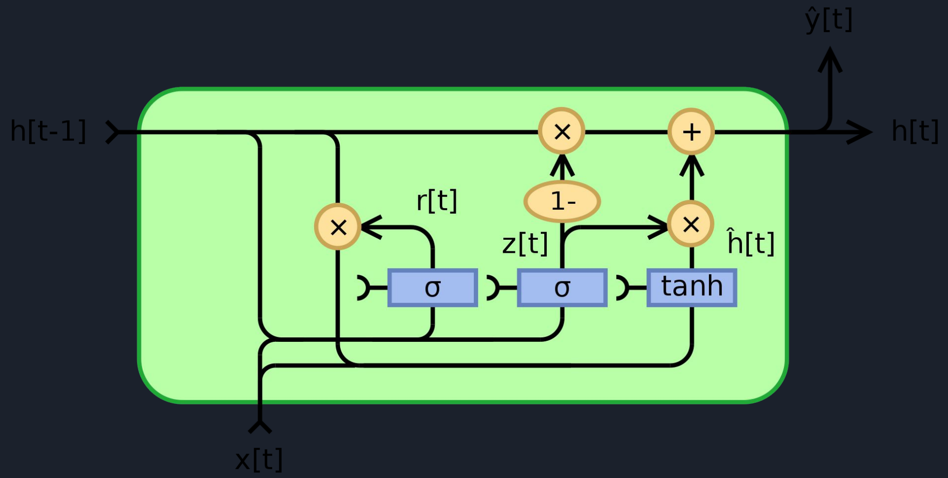
LSTM (1997)

RNNs using LSTM units partially solve the vanishing gradient problem



GRU (2014)

Better performance on smaller and less frequent datasets than **LSTM**.



Attention is all you need (2017) Cited by 18060

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

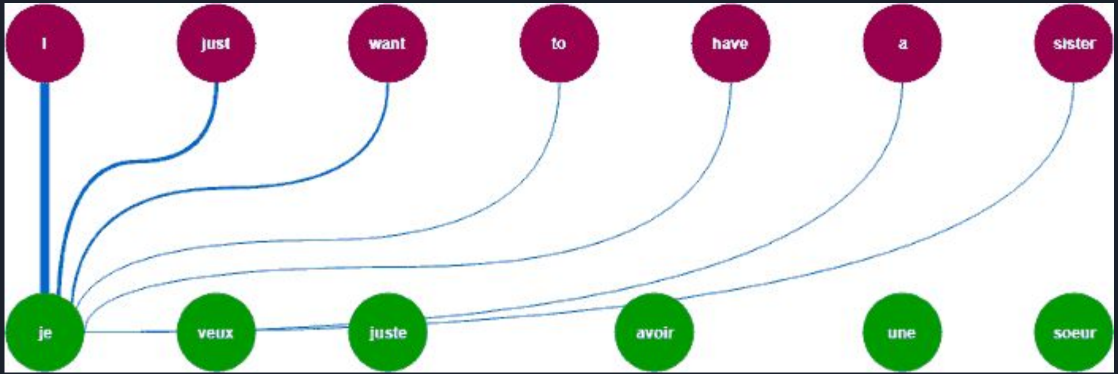
Llion Jones* Google Research llion@google.com

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

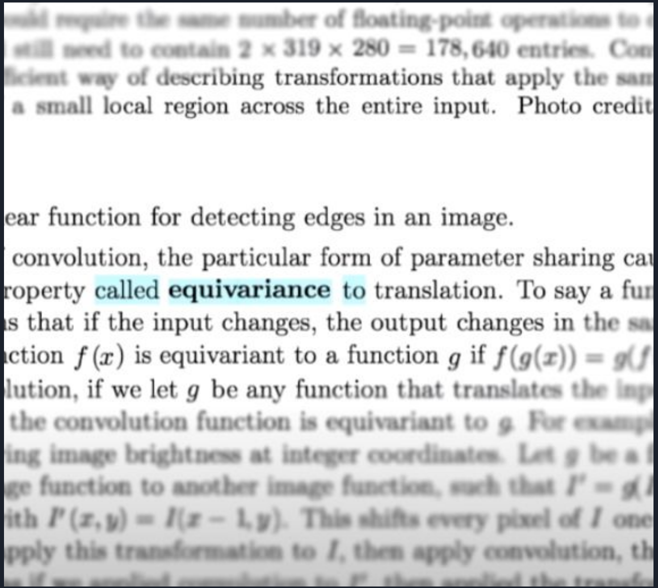
Łukasz Kaiser* Google Brain lukaszkaiser@google.com

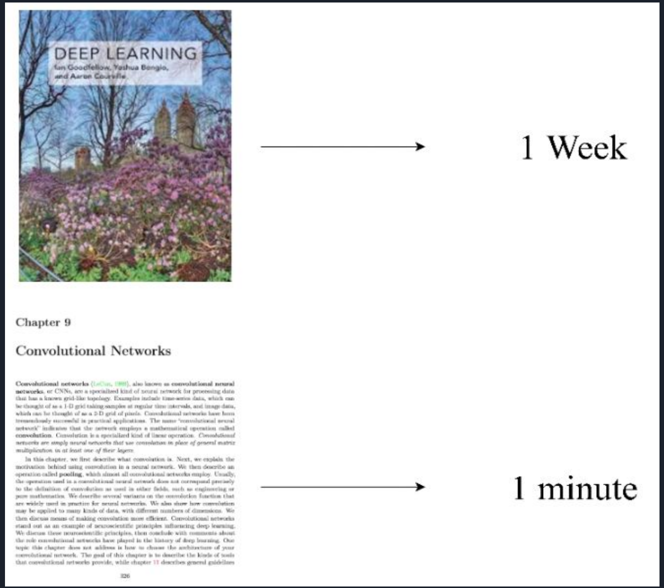
Illia Polosukhin* ‡ illia.polosukhin@gmail.com

Attention in Neural networks

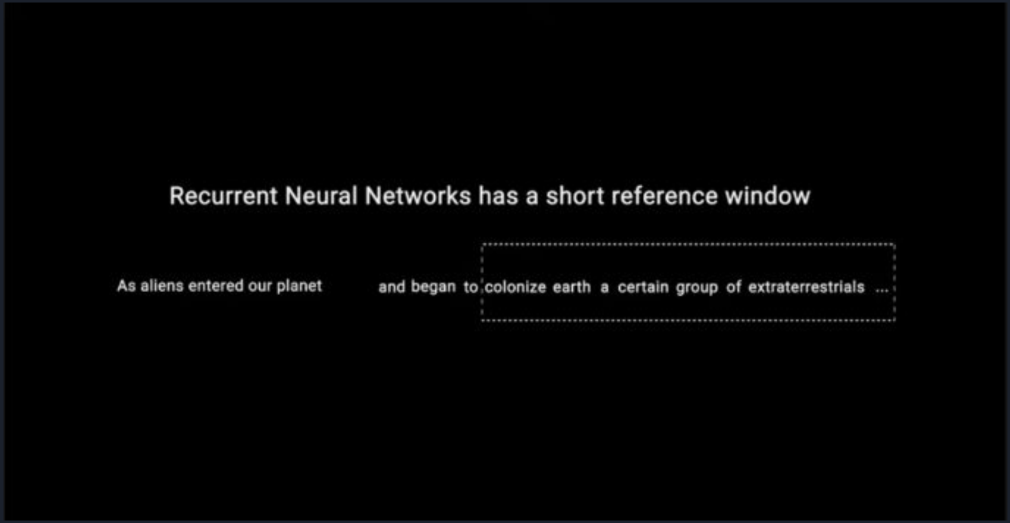


Example : Searching for “Equivariance” in a book



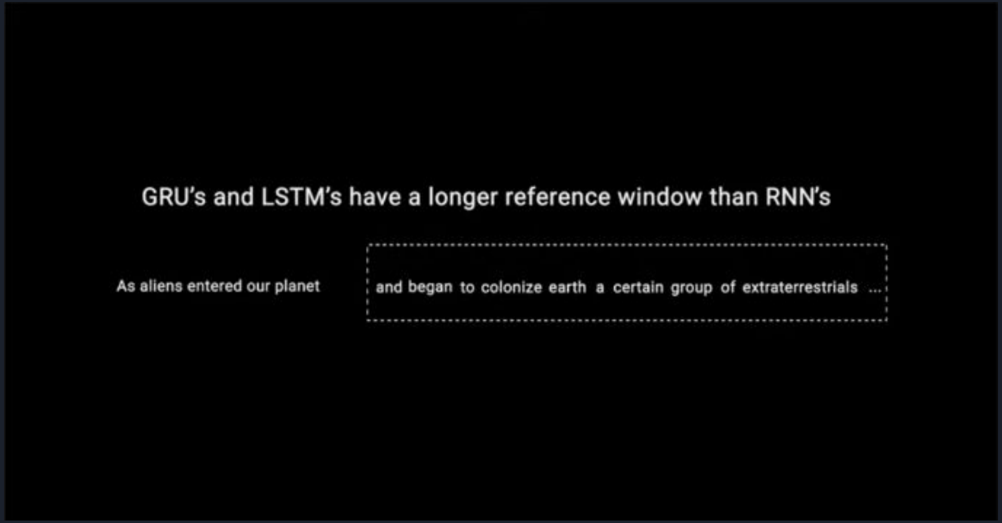


RNN vs LSTM vs Attention



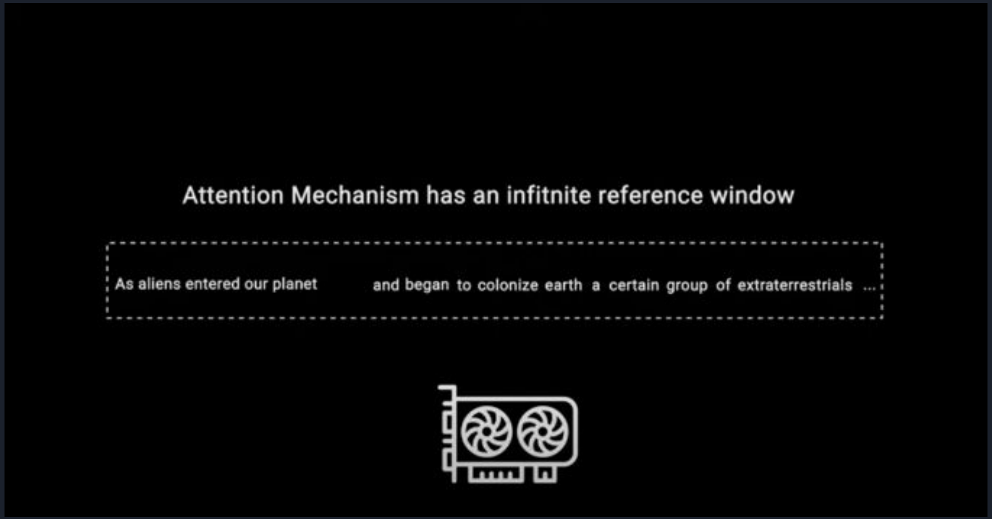
YT-Michael Phi

RNN vs **LSTM** vs Attention



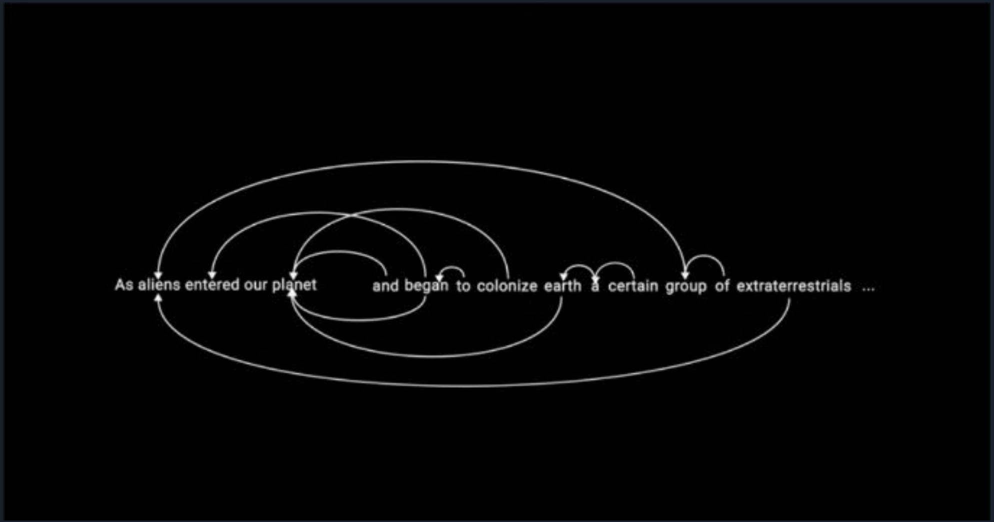
YT-Michael Phi

RNN vs LSTM vs **Attention**



YT-Michael Phi

RNN vs LSTM vs **Attention**



YT-Michael Phi

Attention example in images

A **bodybuilder** holding a dumbbell

YT - Computerphile

Microsoft Attention GANs

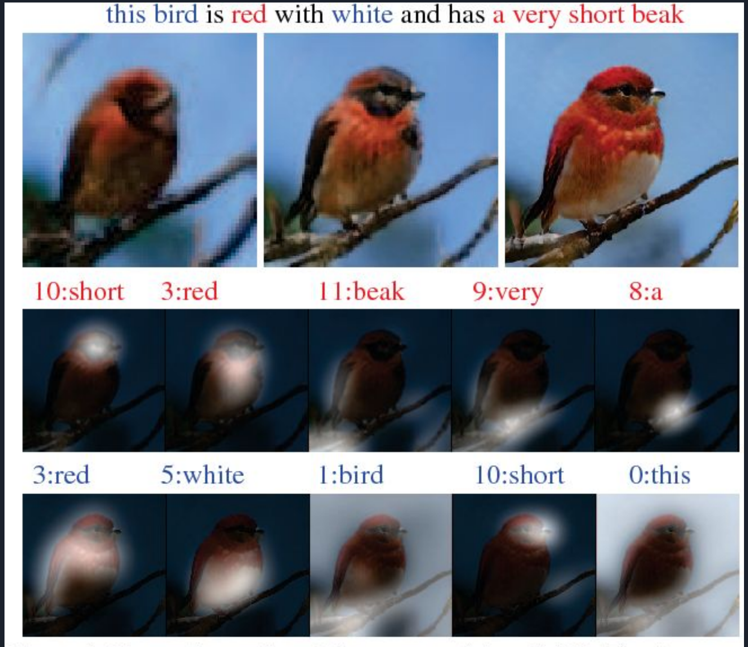


Figure 1. Example results of the proposed AttnGAN. The first row

Self Attention

Self-attention, sometimes called intra-attention is an attention mechanism relating different positions of a single sequence in order to compute a representation of the sequence.
Self-attention has been used successfully in a variety of tasks including reading comprehension, abstractive summarization, textual entailment and learning task-independent sentence representations

Transformers

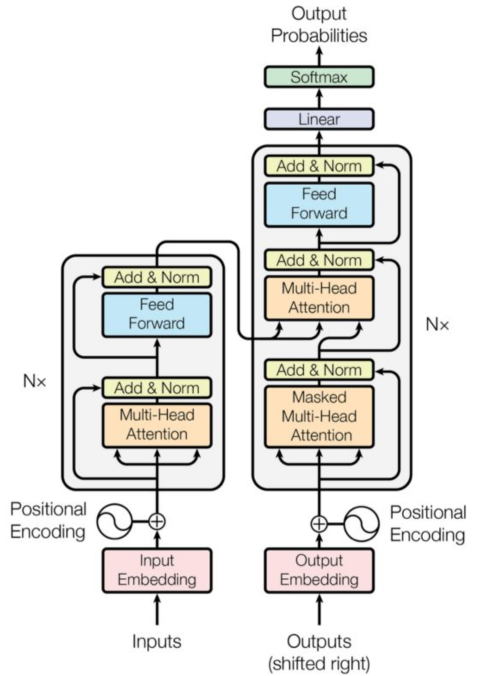
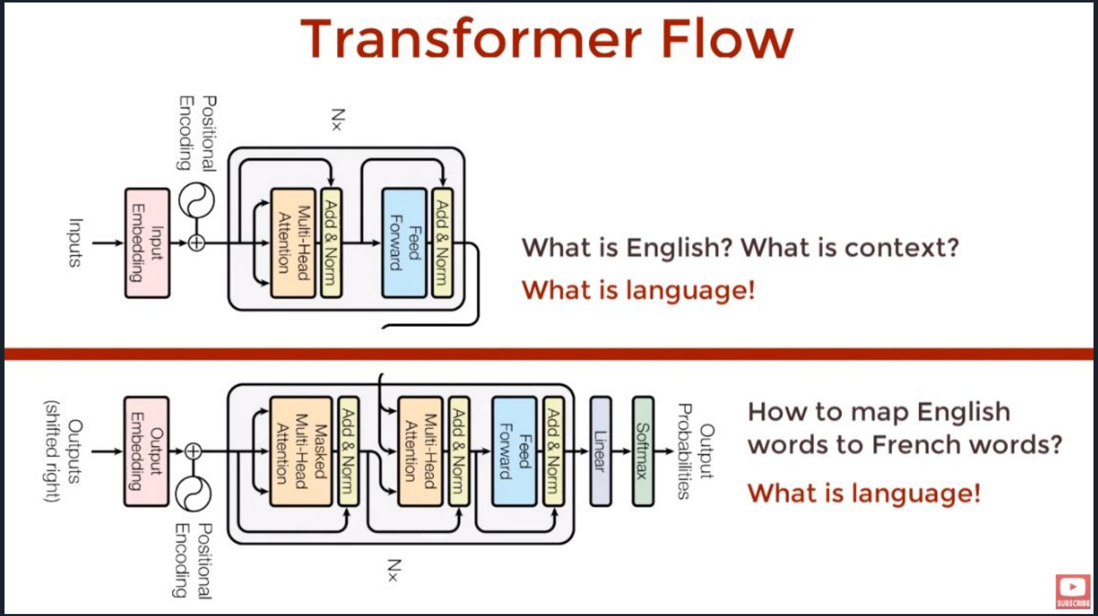


Figure 1: The Transformer - model architecture.

Encoder and Decoder

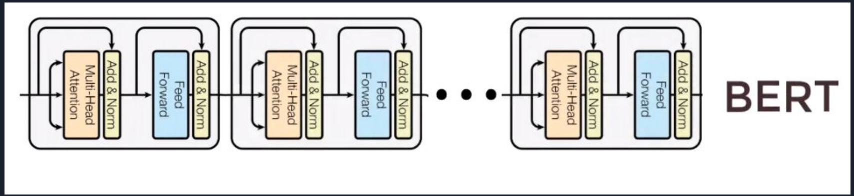


YT-CodeEmporium

BERT and GPT

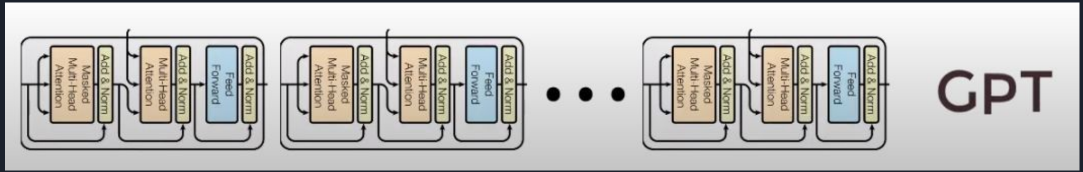
BERT (2018) by Google

A stack of Encoders



GPT (2018) by OpenAI

A stack of Decoders



YT-CodeEmporium

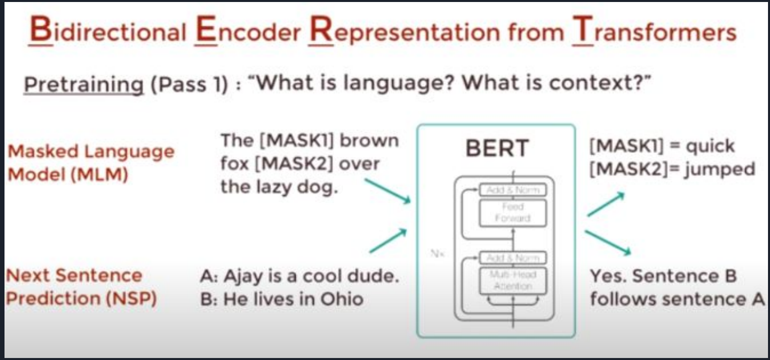
BERT (Bidirectional Encoder Representations from Transformers)

state-of-the-art performance on a number of NLU tasks

- GLUE (General Language Understanding Evaluation)
- SQuAD (Stanford Question Answering Dataset)
- SWAG (Situations With Adversarial Generations)

BERT pre-training procedure

1. Masked Language Model
2. Next Sentence Prediction



YT-CodeEmporium

GPT-1(Generative Pre-trained Transformer)

- GPT is a "transformer" model, which uses "attention" in place of previous recurrence- and convolution-based architectures.
- It showed how a generative model of language is able to acquire world knowledge and process long-range dependencies by **pre-training** on a diverse corpus with long stretches of contiguous text.

GPT-2 (1.5 Billion Parameters)

Dataset for pretraining : 40 GB of text

required tens of petaflop/s-days*

translates text, answers questions, summarizes passages, and generates text output on a level that, while sometimes indistinguishable from that of humans

*One petaflop/s-day is approximately equal to 10^{20} neural net operations

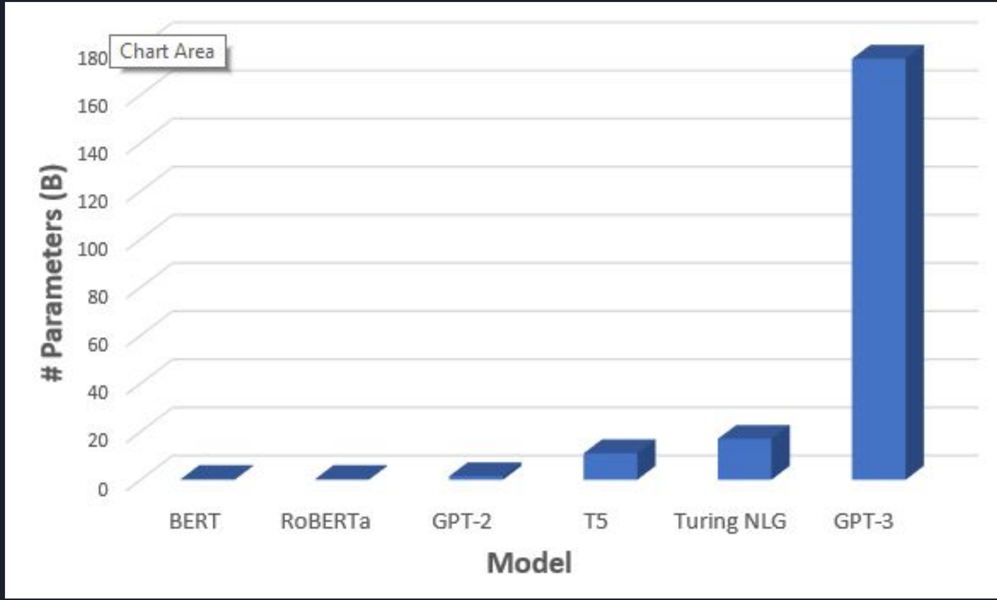
GPT-3 (175 Billion Parameters)

Dataset for pretraining : 570 GB of text

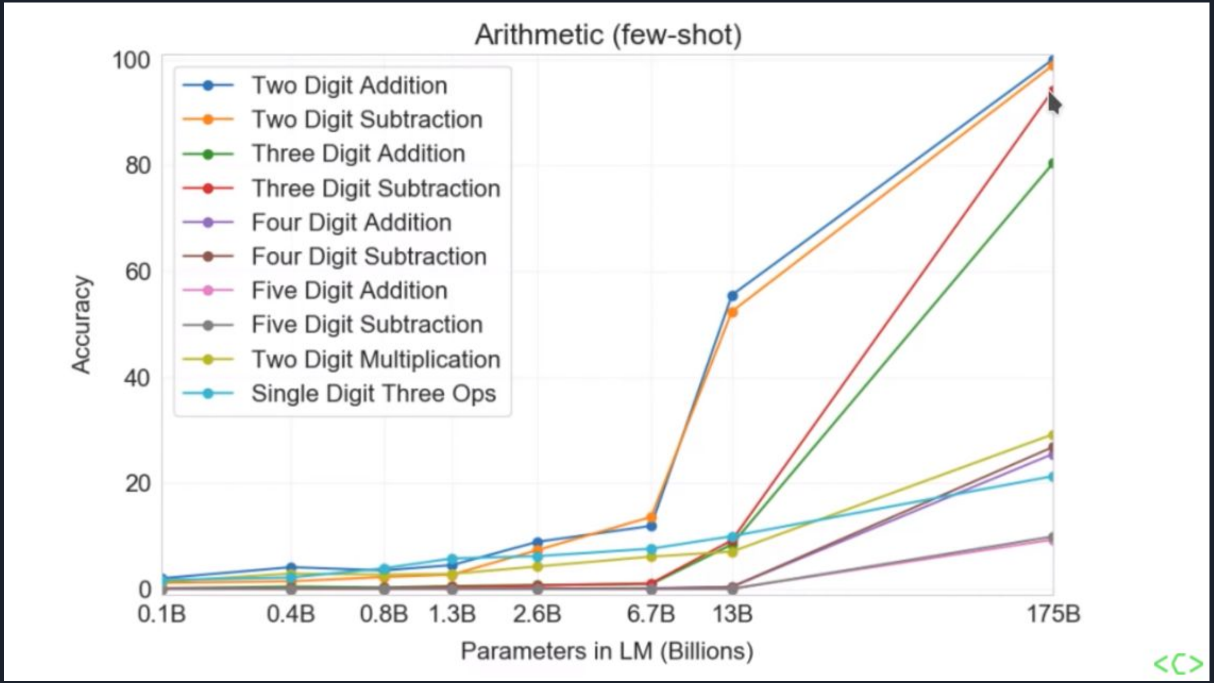
required several thousand petaflop/s-days*

*One petaflop/s-day is approximately equal to 10^{20} neural net operations

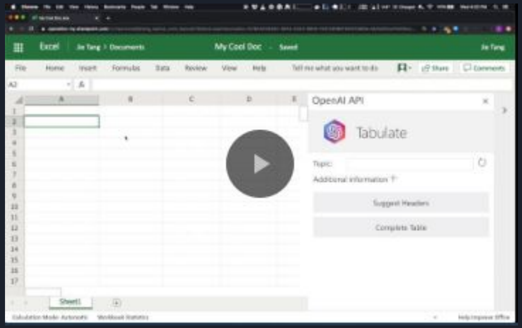
Comparison of sizes



Size Matters!



GPT-3 Demos


Excel
My Cool Doc
OpenAI API
Tabulate


Wikipedia
wikipedia.org
WIKIPEDIA
English
6 095 000+ articles
Español
1 604 000+ artículos
日本語
1 210 000+ 記事
Deutsch
2 441 000+ Artikel
Русский
Français

Transformers Drawbacks

- Very large models.
  - Memory and compute intensive to train
- Relatively young class of models
  - so we know less about them
- Might be worse for hierarchical data (Tran et al, ACL 2018)

Challenges

1. Transformer complexity
2. Longer sequences

Shrinking Transformer

Transformers are becoming both more accurate and larger (t5 has 11 billion parameters)

But there are ways to make them smaller without hurt performance:

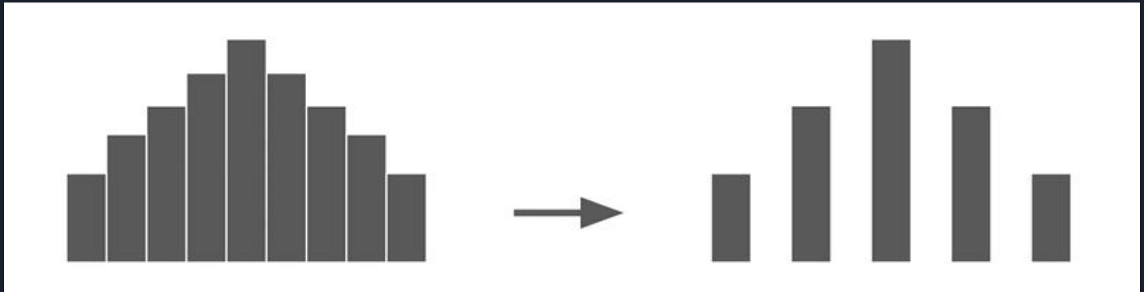
1. Quantization
2. Distillation
3. Pruning
4. More specialized models

Quantization

Reduced number of bit needed to store the trained parameters in model

Convert 32 bit floating point to 8 bit integer

Problem: usually hardware dependant



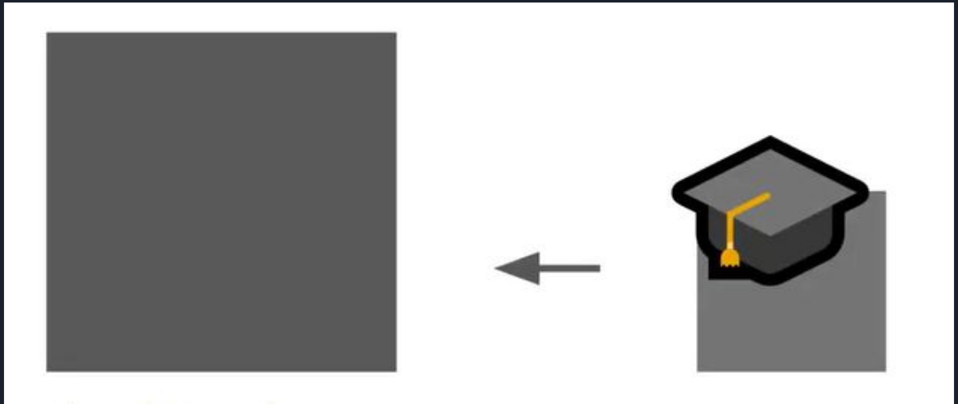
YT-Rasa

Distillation

A new model is trained to predict the weights of one or more layers of the larger model

Up to 100x smaller and 15x faster

Problem: need more setup

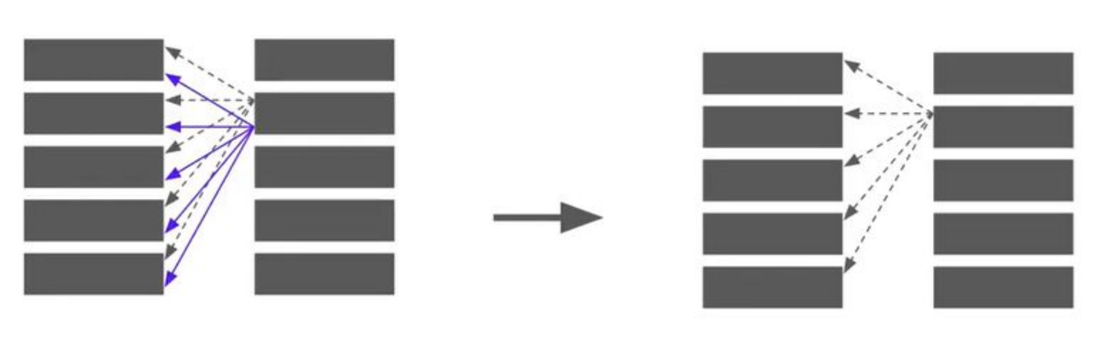


YT-Rasa

Pruning

Remove attention heads based on how useful they are for a specific task

Up to 80% the heads of trained transformer heads can removed without significantly reducing accuracy

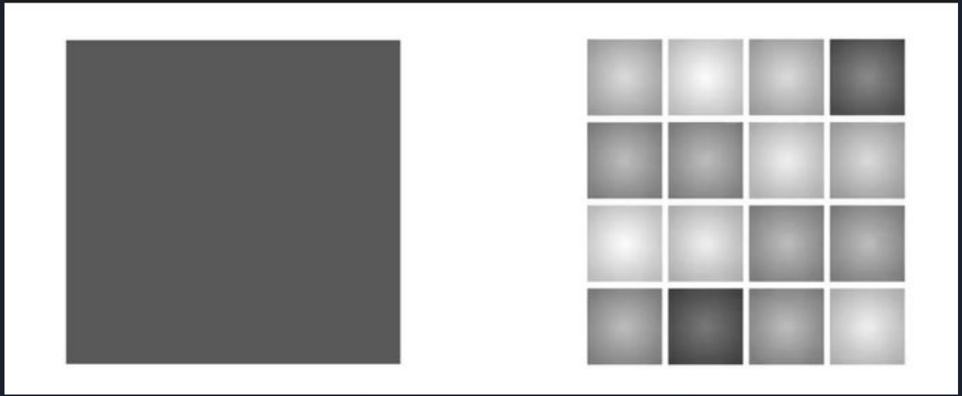


YT-Rasa

More specialized models

Train a special smaller model

Really large nlp models (like bert & gpt) tend to be open domain



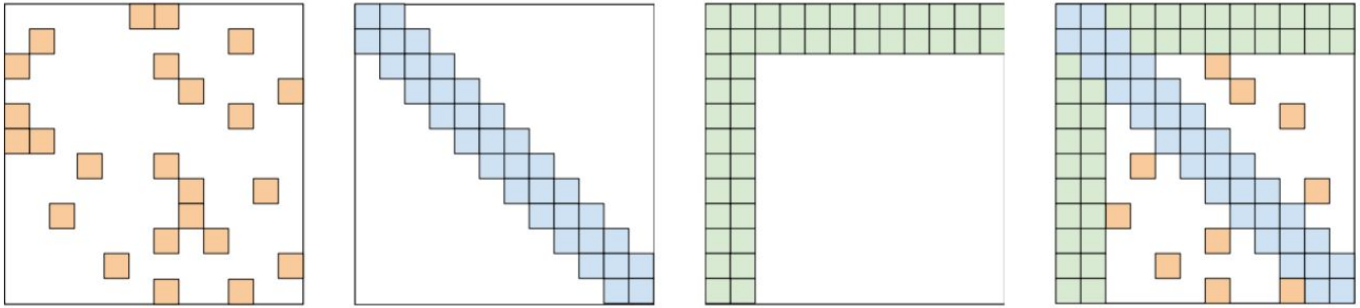
YT-Rasa

Why don’t these methods hurt performance?

1. Really large transformers are bigger than they need to be for some tasks
2. There is a lot of redundancy in these models

Bigbird

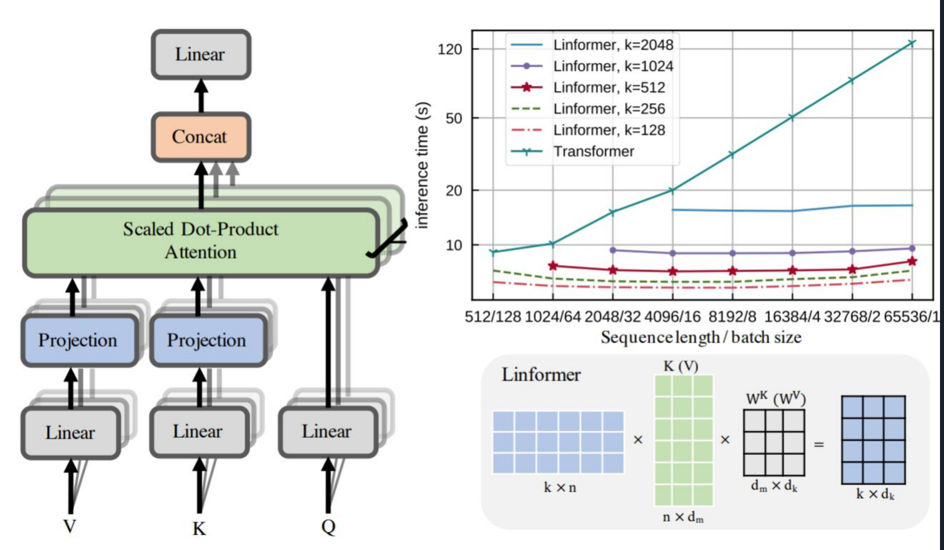
last revised 8 Jan 2021



(a) Random attention (b) Window attention (c) Global Attention (d) BIGBIRD

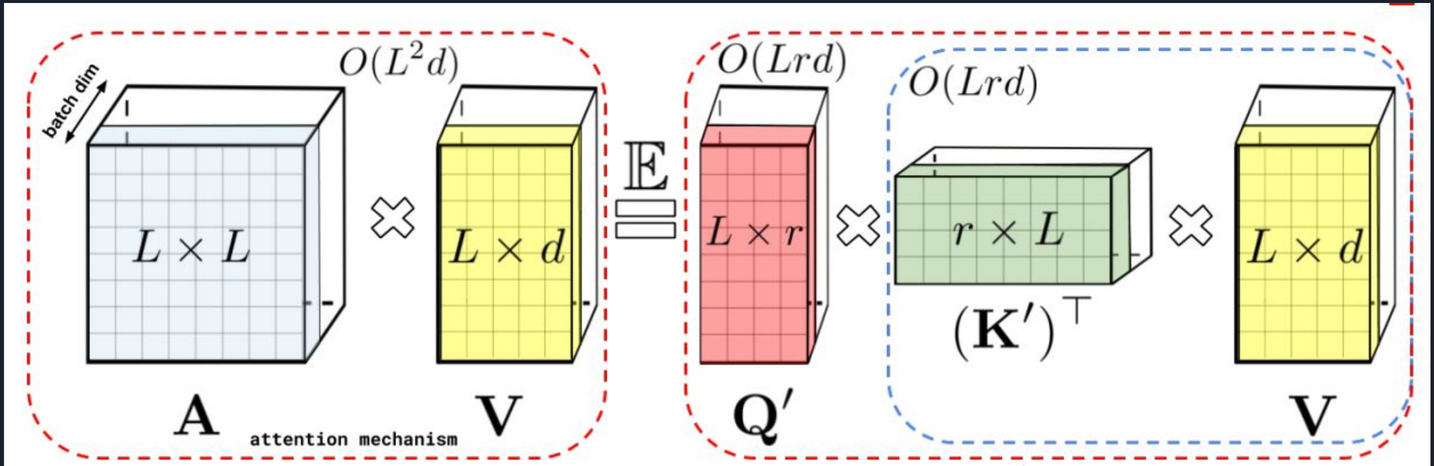
Linformer

last revised 14 Jun 2020



Performer

last revised 16 Feb 2021



References

Recurrent neural network - Wikipedia

Long short-term memory - Wikipedia

Gated recurrent unit - Wikipedia

Transformer and language models - YouTube (playlist)

BigBird - arxiv

Linformer - arxiv

Performer - arxiv

Thanks

hasanmahani08@gmail.com

rezadar1378@gmail.com