Transformers: Never Ending Story

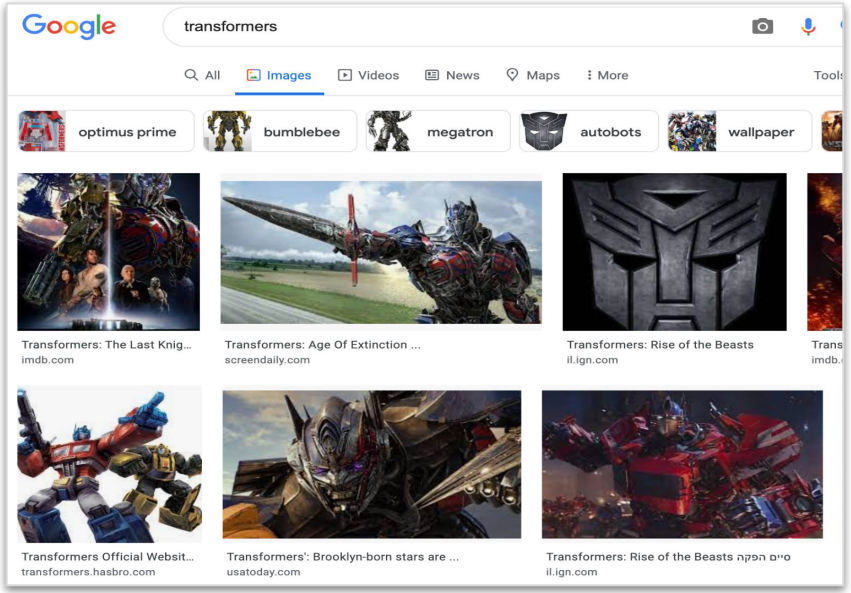
Mike Erlihson, PhD

Principal Data Scientist, Salt Security

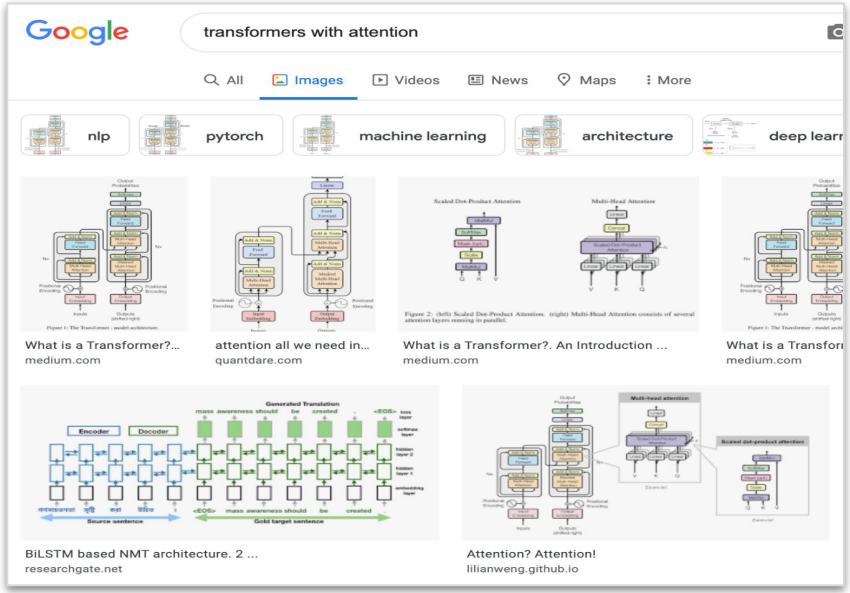
Presentation Credits: Hila Paz H.

Transforming 2017

Transformers before 2017



Transformers today



About Me

Mike Erlihson, PhD

- Math, DS and ML

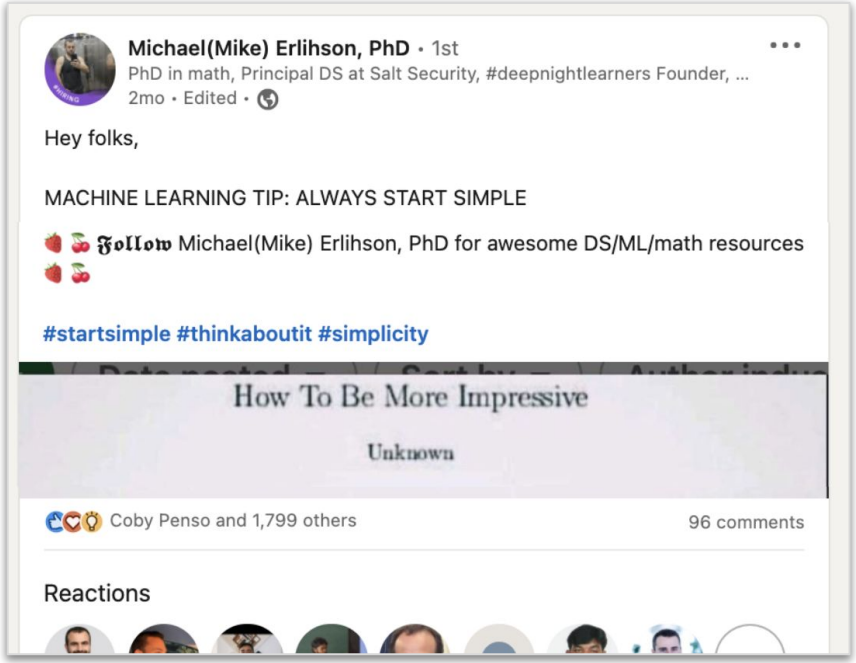
  enthusiast + populizer

- Leading the DS at Salt Security

  Attackers are more sophisticated
  Transformers in Cyber

I'm an Influencer

- #deepnightlearners
  Founder
- Machine and Deep Learning in Hebrew
  Coauthor
- Keep in touch:
  - michael-mike-erlihson-phd
  - maerlich

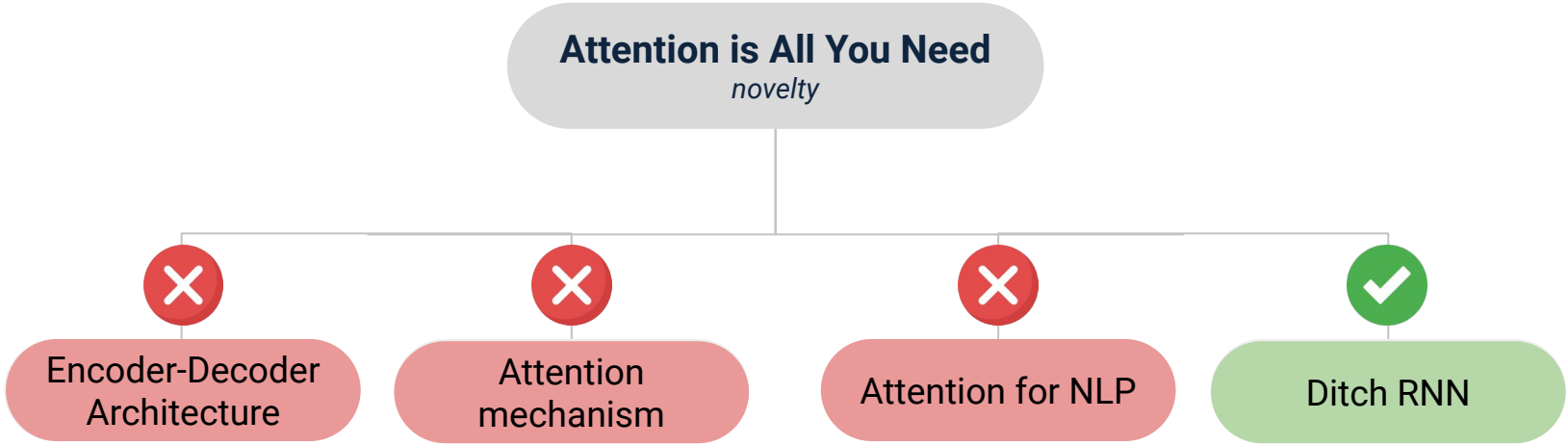


Transformers: Never Ending Story

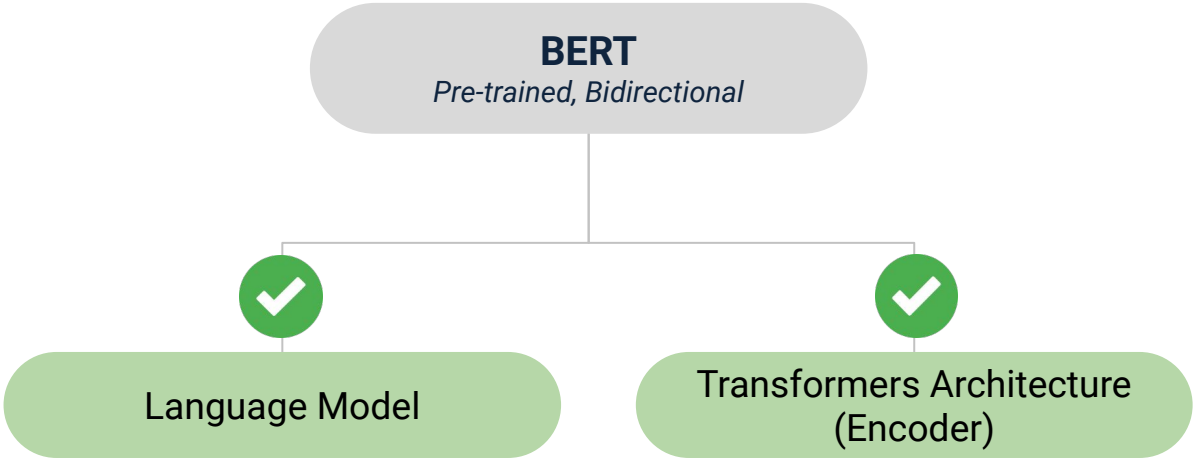
Agenda

- What's so special about 2017?
- Breaking the myth: what really changed
- What's still improving

2017: Meet Transformers

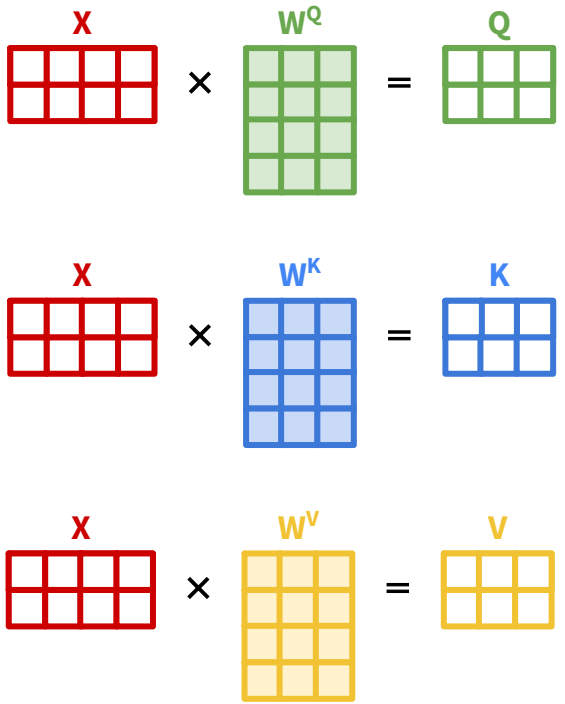


2017: Meet Transformers & BERT

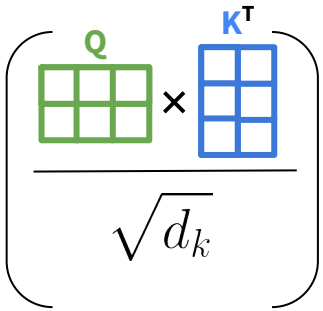


Known Ideas, Different Terminology

- Attention: “measuring the token context”
- Contextualized representations
- We had it before 2017

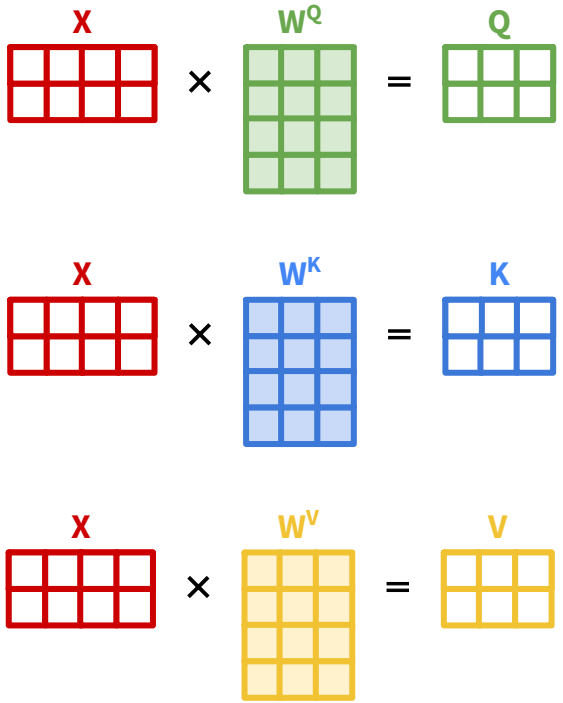


Attention

$Z = softmax$ 

$$\text{Attention Weights} = softmax(\frac{Q \times K^T}{\sqrt{d_k}})$$

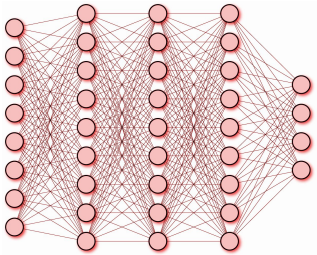
$$Attention\ Output = Attention\ Weights \times V \times X$$

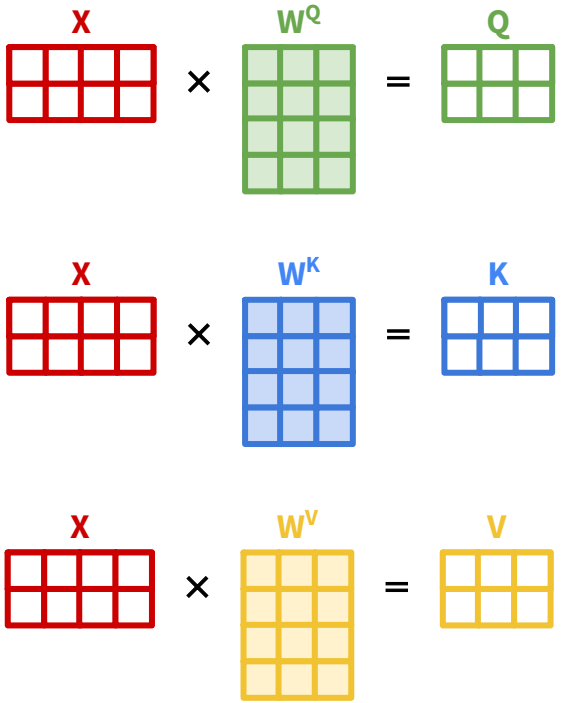


Attention with a Punch of Math

It's just a fully connected layer with input- dependent weights

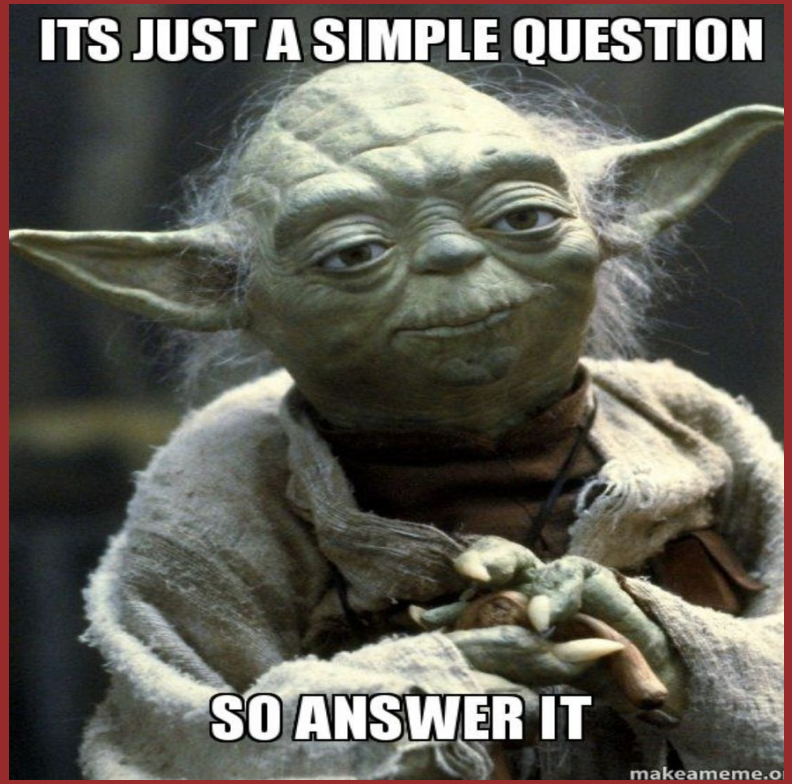
=> Transformers are a generalization of a fully connected network!





Transformers Revolution

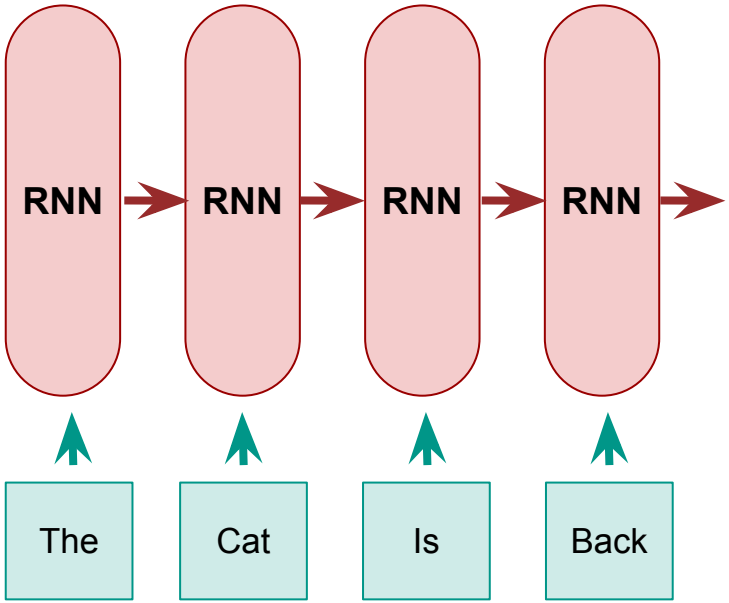
How to feed the network?



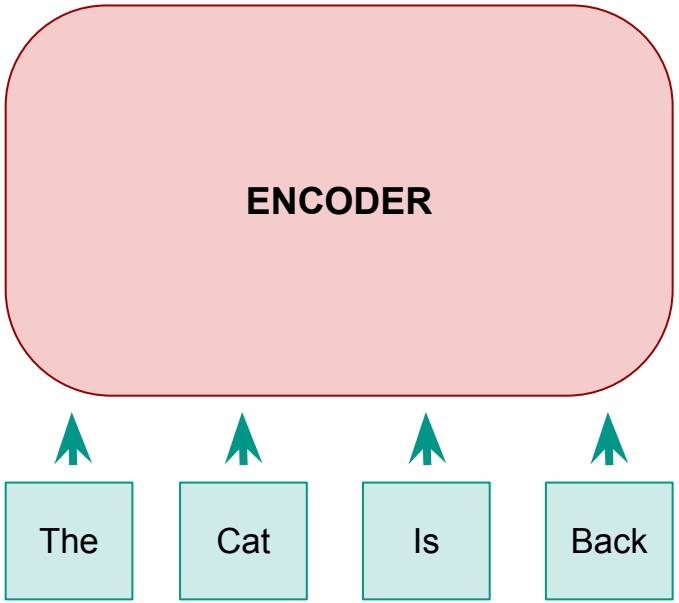
Feed the entire input sequence.

Transformers: Feed the sequence

RNN Base Encoder



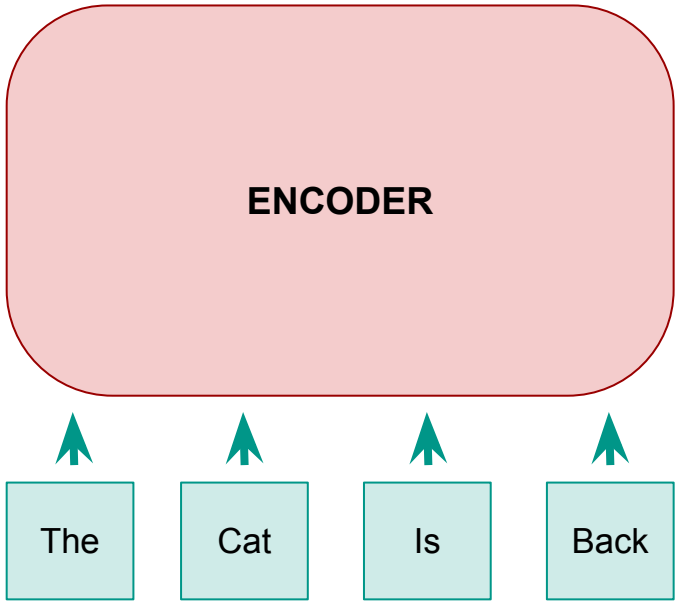
Transformer's Encoder



What is Different?

No dependencies
among hidden states.

Transformer's Encoder



The History Started Before 2017

(-∞ , 2017)

- Encoder-Decoder Architecture
- Attention mechanism
- Contextualized embeddings

2017

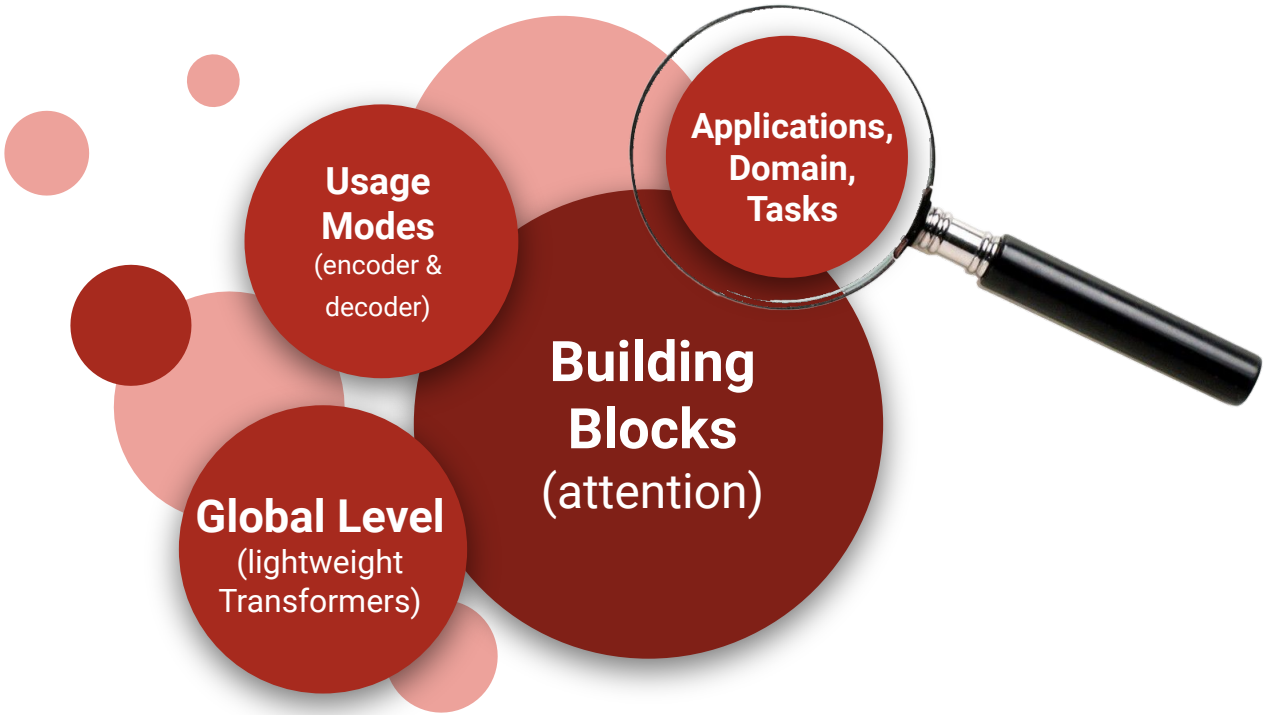
- Totally NO-RNN Architecture
- Multi-Head Attention (many transformer blocks stacked)

Transformers Revolution

Mission Accomplished?

We’re just getting started!

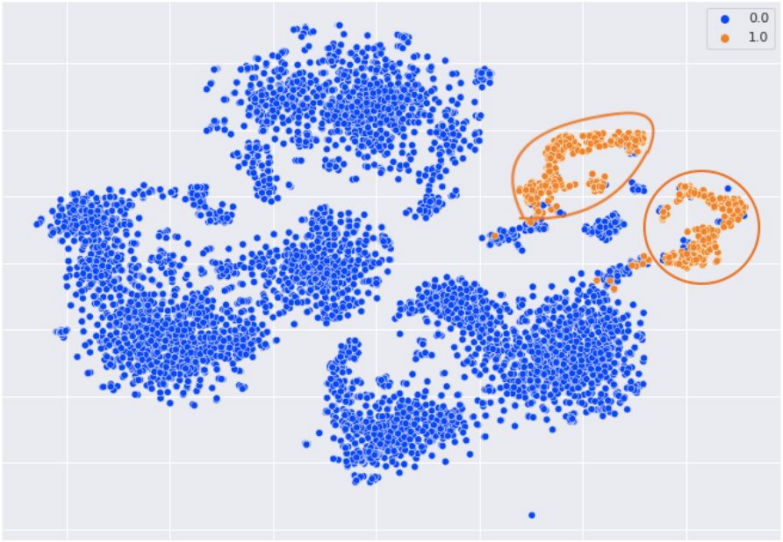
Transformers Improvement Directions



Transformers Applications in 2017

NLP

- Machine translation
- Language modeling
- Sentiment Analysis
- Named entity recognition
- Summarization
- Topic Modeling



Transformers Applications After 2017

Audio Processing

- Speech recognition
- Speech synthesis
- Speech enhancement
- Music generation

Transformers Applications After 2017

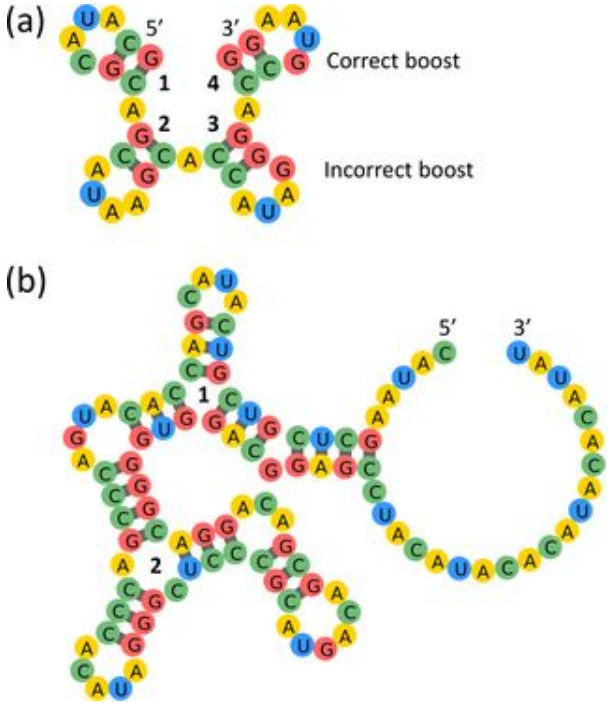
Computer Vision

- Image classification
- Object detection
- Image generation
- Segmentation
- Video processing

Transformers Applications After 2017

Life Science: Biology, Chemistry and Genetics

- Protein folding prediction
- DNA & RNA sequences
- Drug discovery



Transformers Applications After 2017

Multimodal

- Visual question answering
- Visual commonsense reasoning
- Caption generation
- Speech-to-text translation
- Text-to-image generation

  "Colourful cubist painting of a parrot in a cage"

But I'm working at a cyber security company!

Transformers and Cybersecurity

- Treat JSON (key:value) as tokens
- Benefit from Transformers to track expected behavior
- Detect outliers

```
{
    “date”: “2022-02-20 00:00:01”,
    “activity”:{
        ”copy”: “all-resources”
        “Id”: “I am a top-notch hacker”
    },
    “userAgent”: “Very Suspicious-UA”
}
```

Salt Security

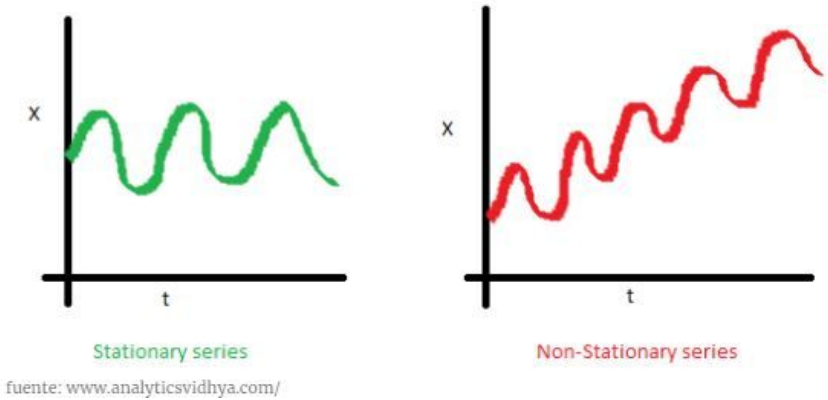
And Transformers

*** Promoted content alert!*

Time Series

Non-stationary multivariate time-series analysis and outlier detection

Statistical Anomaly Detection Techniques in Non-Stationary Environment

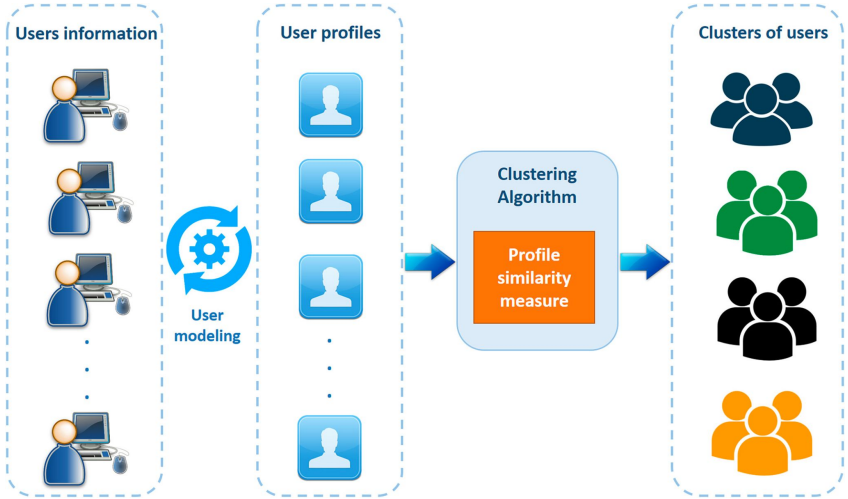


fuente: www.analyticsvidhya.com/

User Clustering

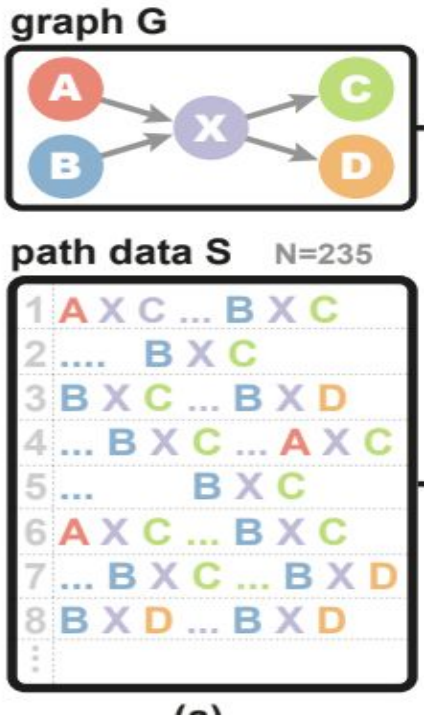
Using online and streaming clustering techniques (due to non-stationarity)

Issue: Appropriate distance metric choice is very untrivial



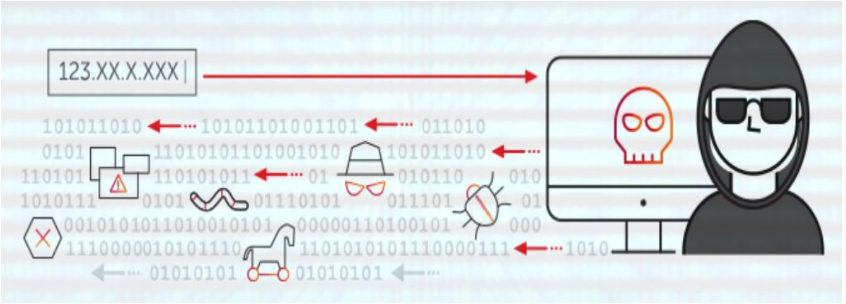
Anomaly Detection

Detection of anomalous path (routes) on user activity graphs

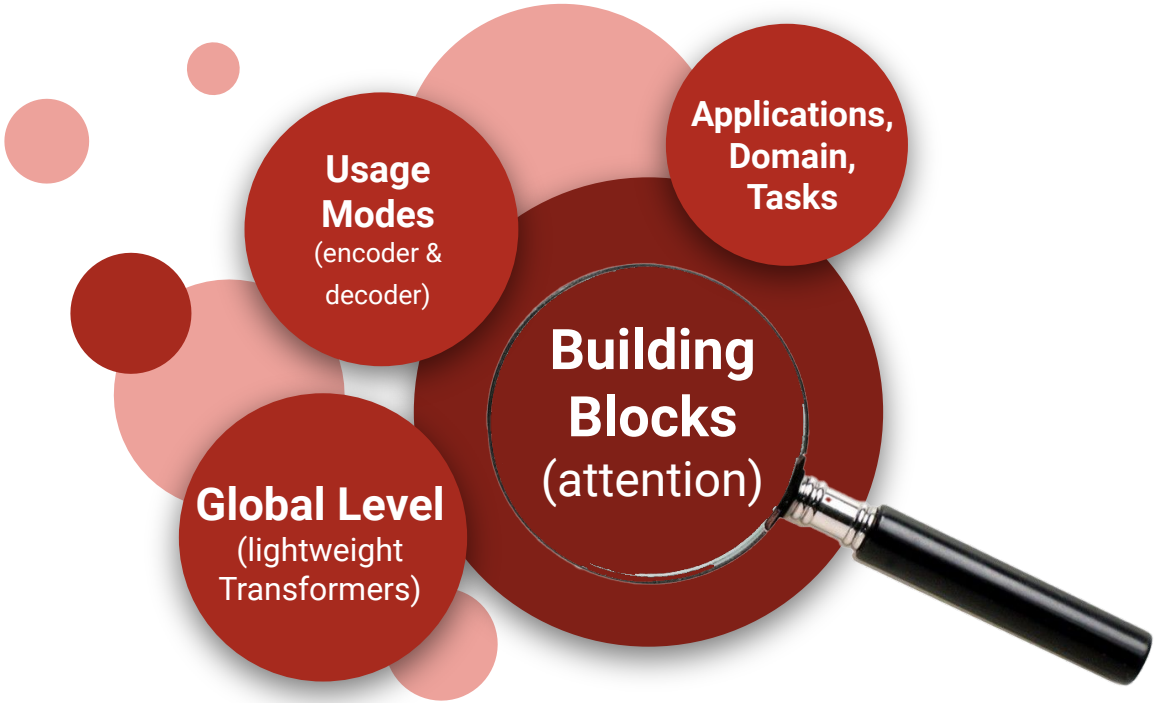


Pattern Analysis

API Requests Content Analysis in Constantly Changing Environment



Transformers Improvement Directions



What's wrong with 2017 attention?

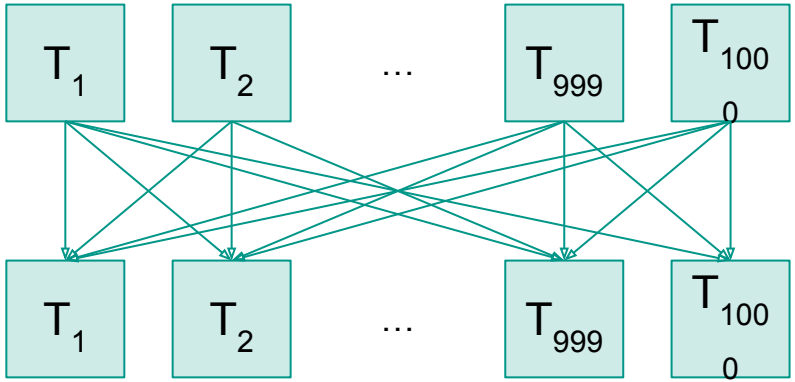
Quadratic Complexity (Input Length)

For each attention head:

- Compute all dot products (nxn)
- Compute all attention scores: softmax (nxn)

For 1K inputs and h heads

- h □ (1M + 1M)



Quadratic Complexity (Input Length)

For each attention head:

- Compute all dot products (nxn)
  Compute all attention scores: softmax (nxn)

For a 256x256 Image

- 64K tokens
- 4B attention coefficients

For a 4K Image?

- …

No Inductive Bias

Even if we had all computation power in the world

- Many features
- Few Train Samples

= **Overfit** (false attentions)

Which Attention Should We Choose?

Not All Attentions Are Equal

Attention wise:

- 1 & 2 are important
- 1 & 3 are nearly as important
- 1 & 4 seems less important

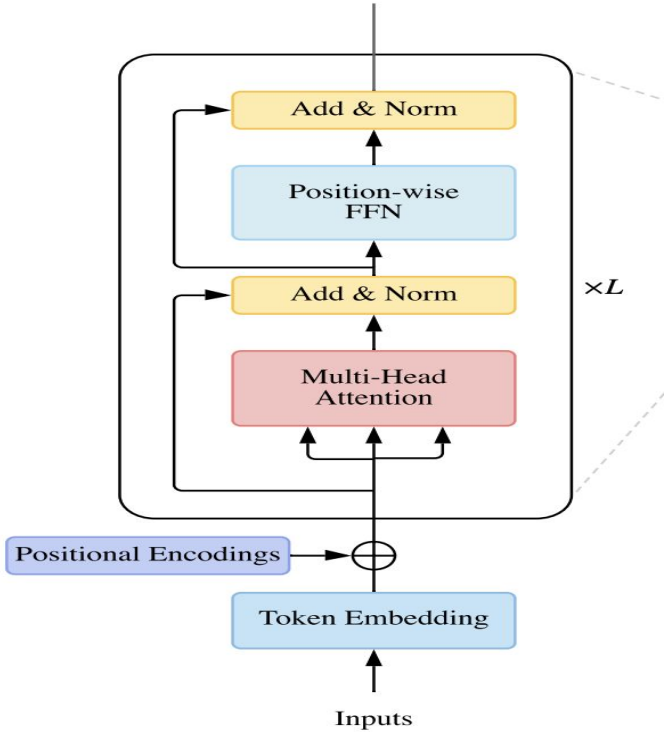
Should Position Matter?

Token position encoding **is** handled efficiently using Fourier-style representation

But we can do better!

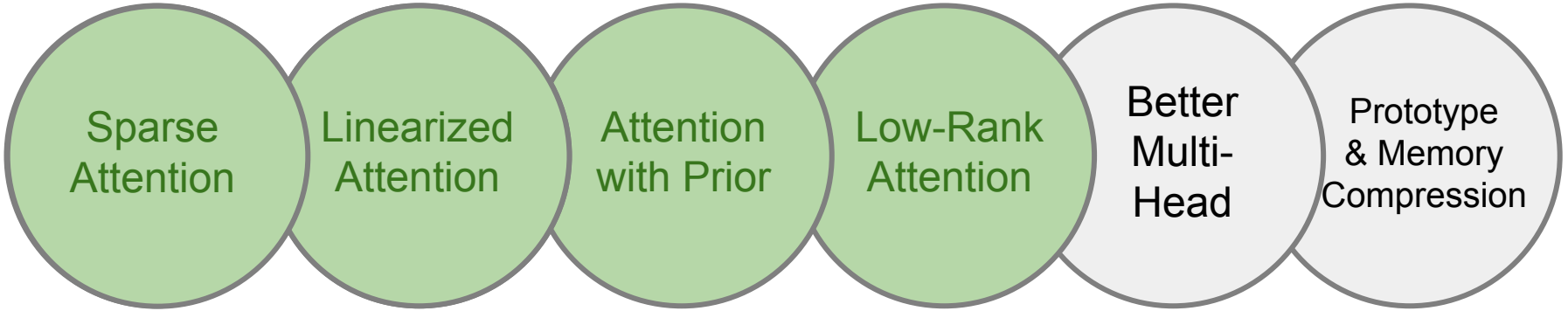
Transformers

Let’s Transform even further!



The original building blocks

How to Improve the Attention Mechanism



Sparse Attention

Not all inputs should be connected

How to choose which are important?

Sparse Attention

Compute Attention Between Part of Tokens Pairs Only

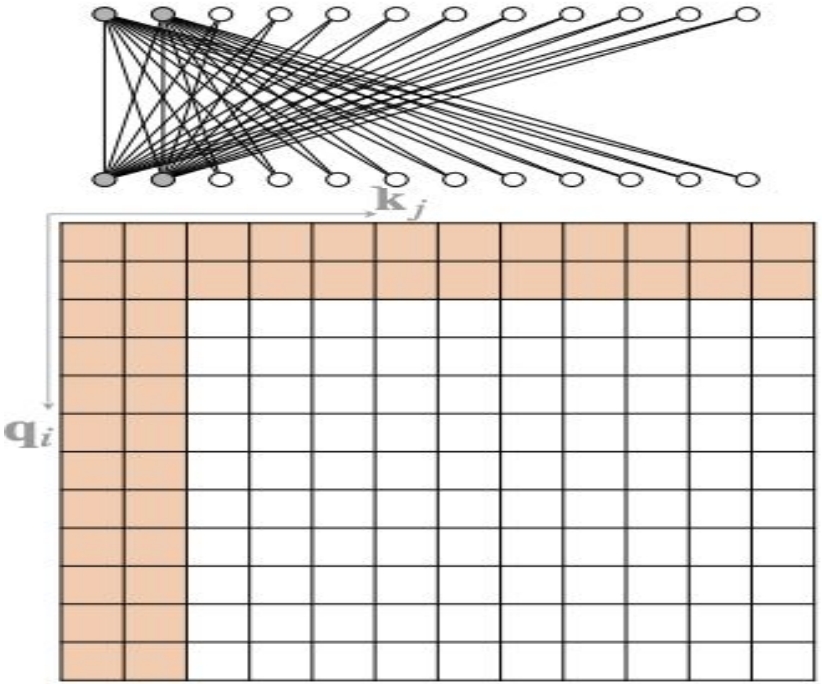
Sparsity (inductive) bias into attention mechanism, leading to reduced complexity.

Sparse Attention Variants

Atomic Pattern 1

- Add first k “global” tokens

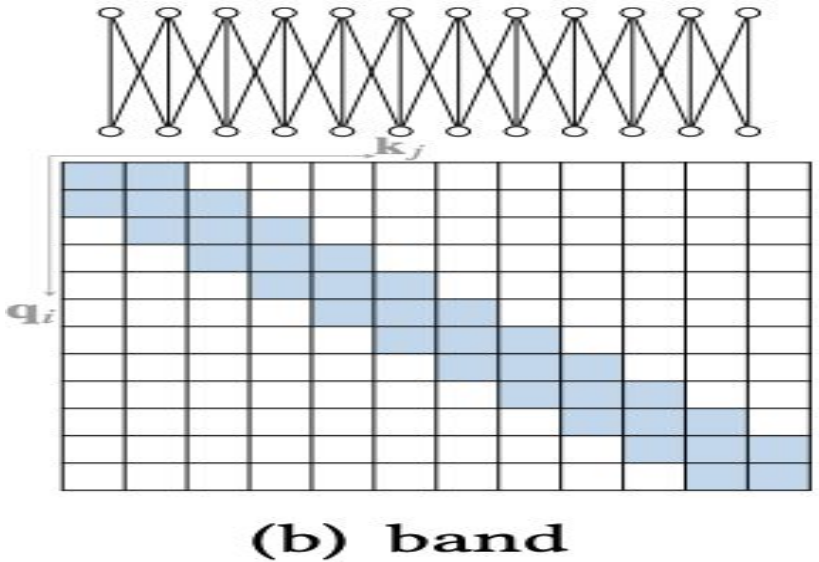
- Compute global attention (‘hubs’ among tokens)



Sparse Attention Variants

Atomic Pattern 2:
Sliding window/local attention (band attention)

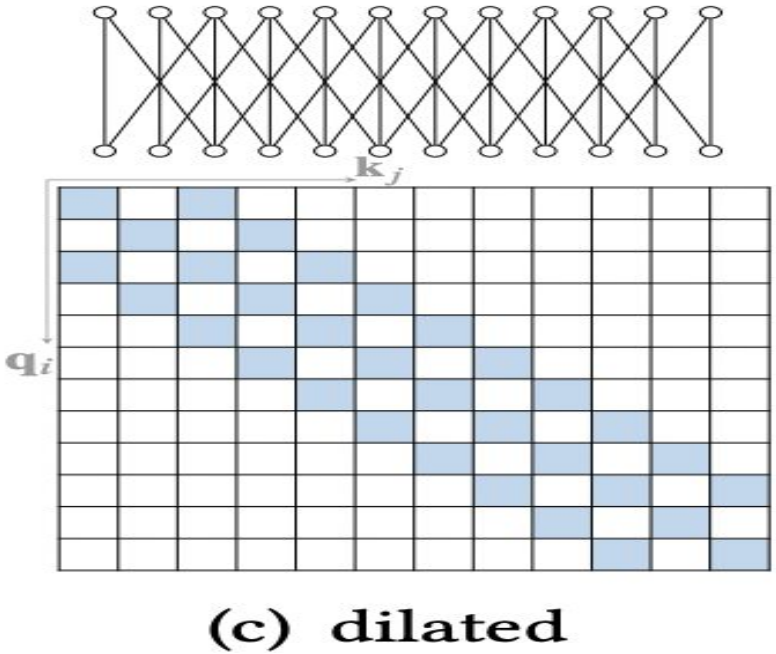
- Query attends neighbor elements
- Suitable for data types with inherent locality dependencies
- BackDoor Inductive Bias



Sparse Attention Variants

Atomic Pattern 3:
Dilated (Strided) Attention

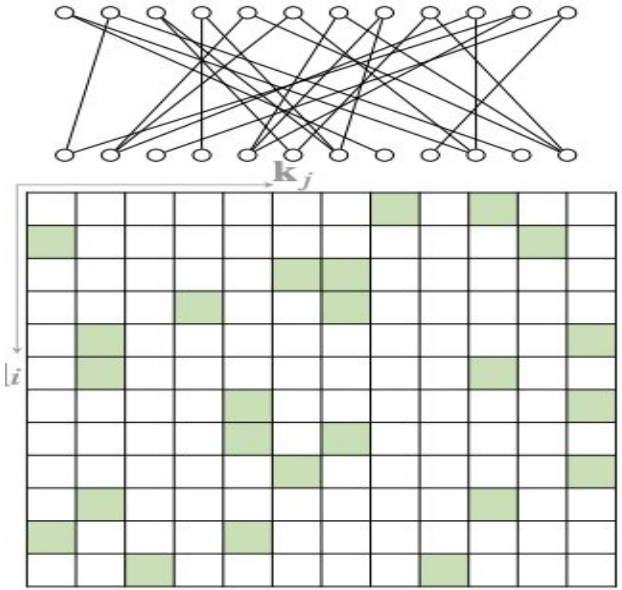
- Increase band attention receptive field without increasing computation complexity



Sparse Attention Variants

Atomic Pattern 4:
Random Attention

- Few tokens randomly sampled for every query to increase “expressiveness” of non-local interactions

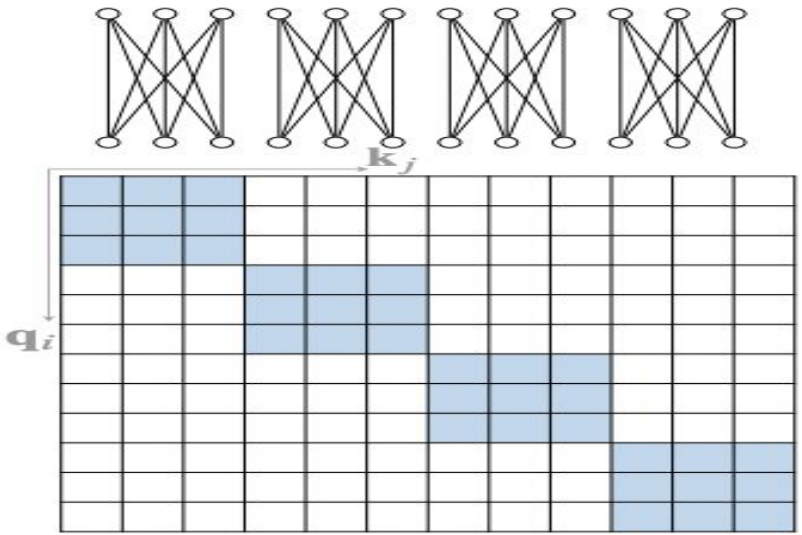


(d) random

Sparse Attention Variants

Atomic Pattern 5:
Block local attention

- Split input sequence into several non-overlapping Q blocks
- Each one has its local memory block. Q-s attend to K-s in the corresponding cluster



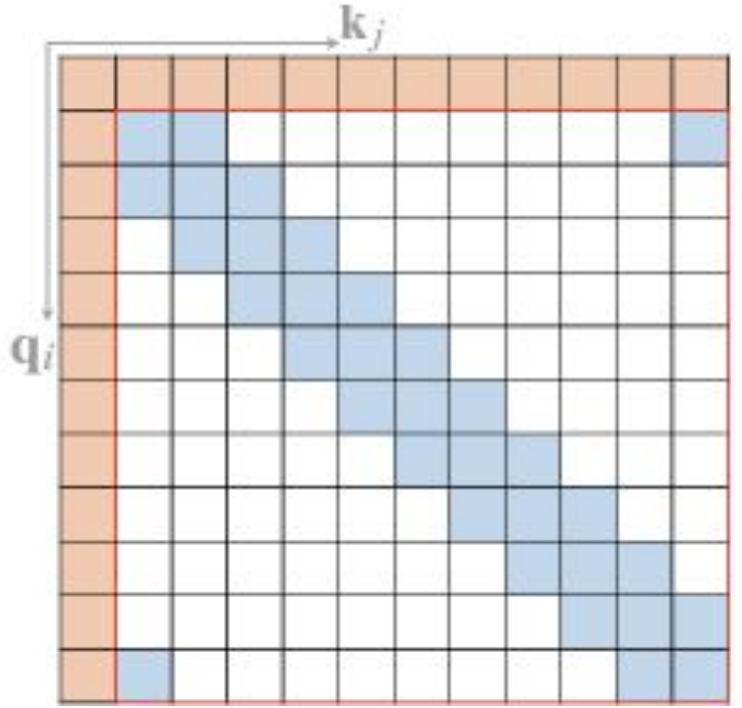
(e) block local

Choose Only One?

Sparse Attention Variants

We can combine a few atomic patterns together!

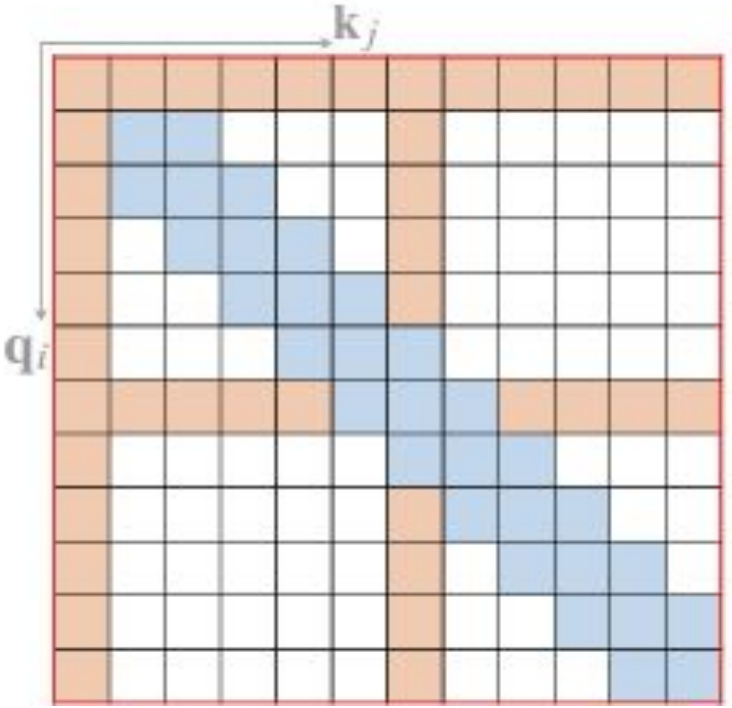
- Star Transformer



Sparse Attention Variants

We can combine a few atomic patterns together!

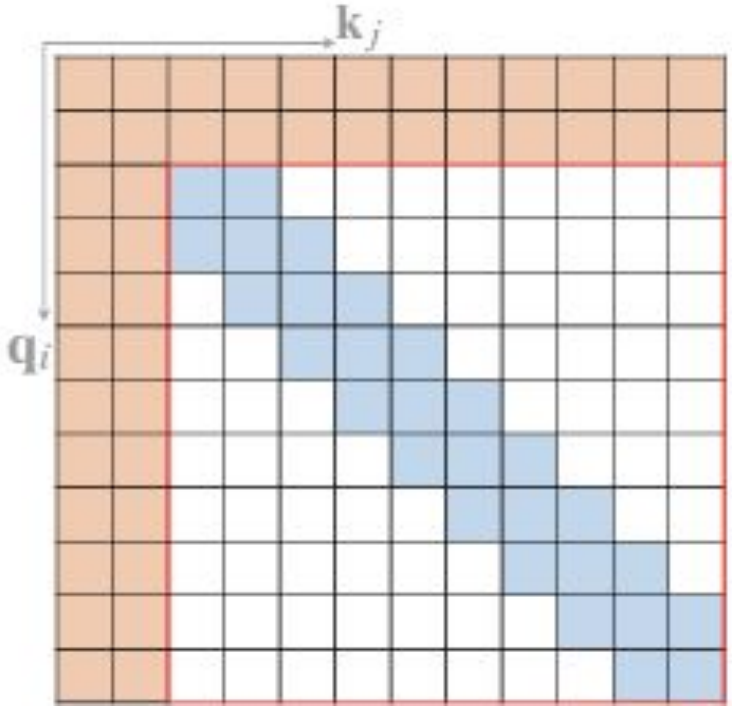
- Longformer



Sparse Attention Variants

We can combine a few atomic patterns together!

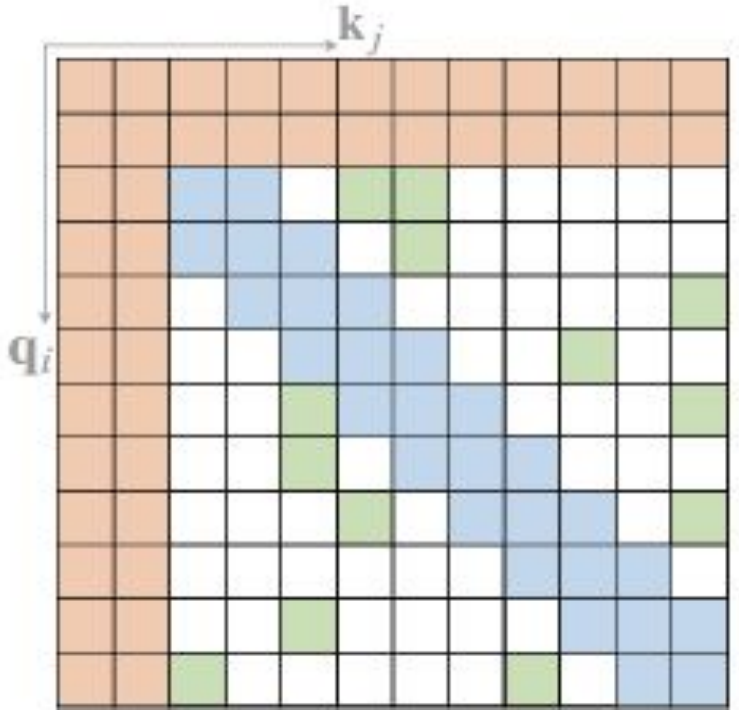
- ETC



Sparse Attention Variants

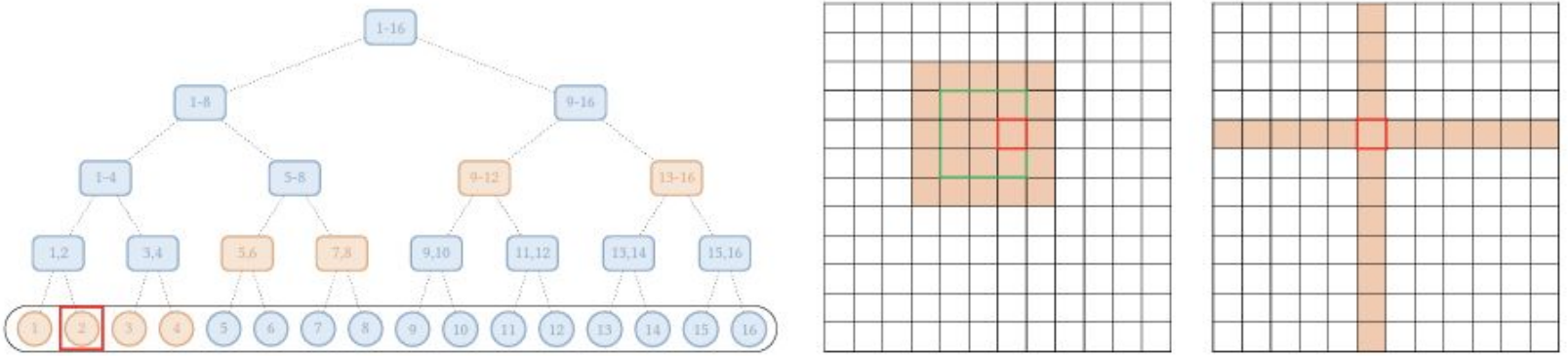
We can combine a few atomic patterns together!

- Bigbird



Sparse Attention Variants: Extended Attention

Mostly for images

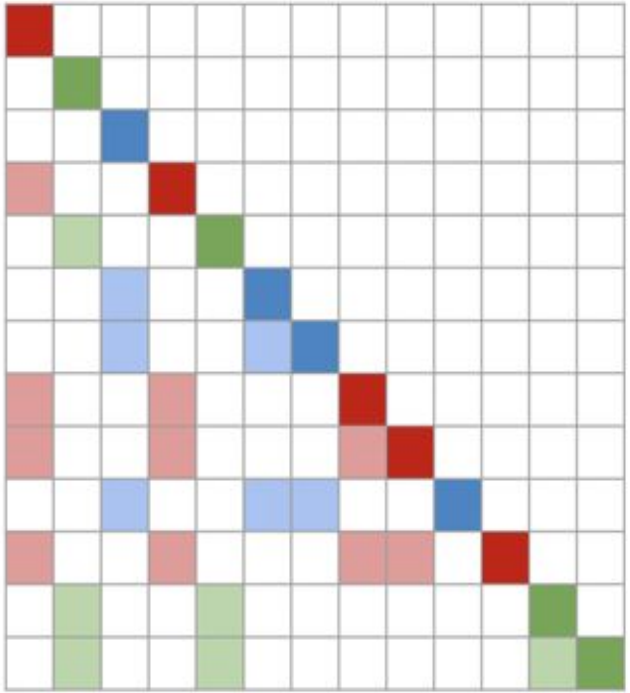


(a) BPT (b) block local (2D) (c) axial (2D)

- The red box indicates the query position
- Orange nodes/squares means corresponding tokens attended to by the query

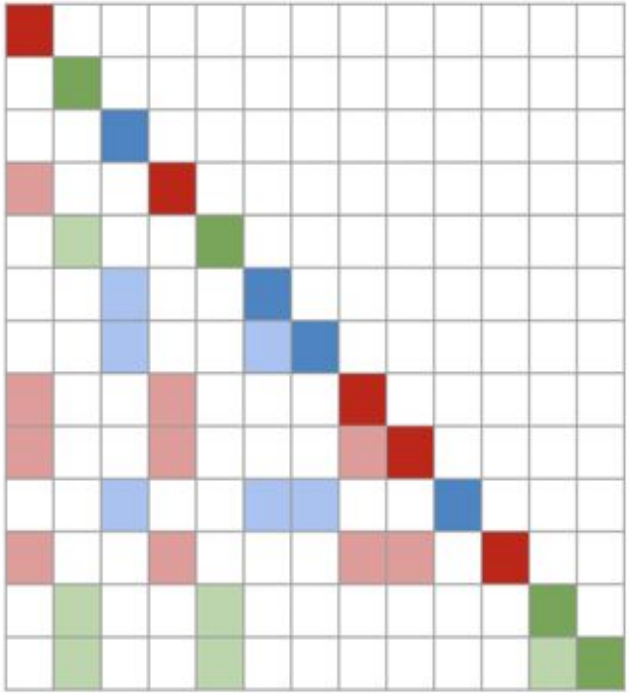
Sparse Attention Variants: Content-based

- Select keys likely having high similarity scores with a given query
- But do we still need to compute all attention scores for it
- NO!!

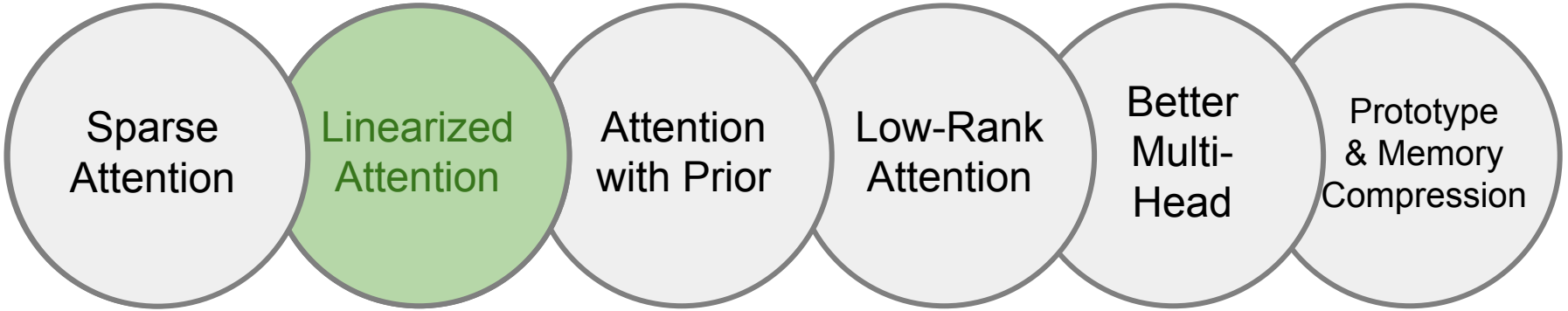


Sparse Attention Variants: Content-based

- Cluster queries & keys on the same set of centroid vectors
- Attention computed for Qs and Ks in the same K-means clusters
- Cluster centroids are updated with the exponentially moving average of vectors assigned to it



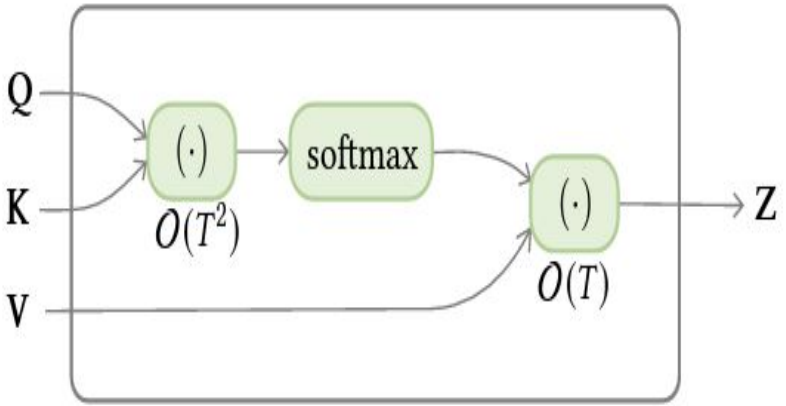
How to Improve the Attention Mechanism



Linearized Attention

Linearly approximates softmax in attention weights computations - computed in reversed order with linear complexity

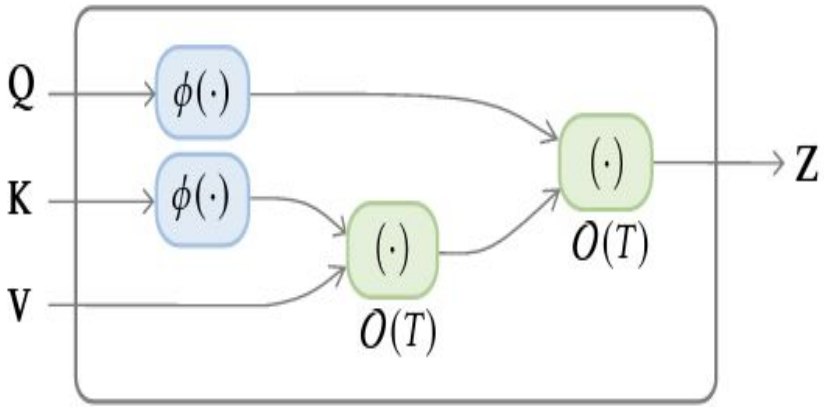
- Standard Self-Attention



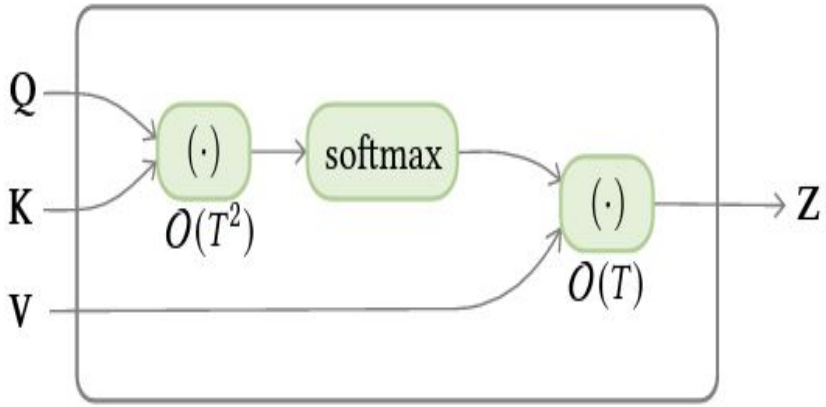
Linearized Attention

Linearly approximates softmax in attention weights computations - computed in reversed order with linear complexity

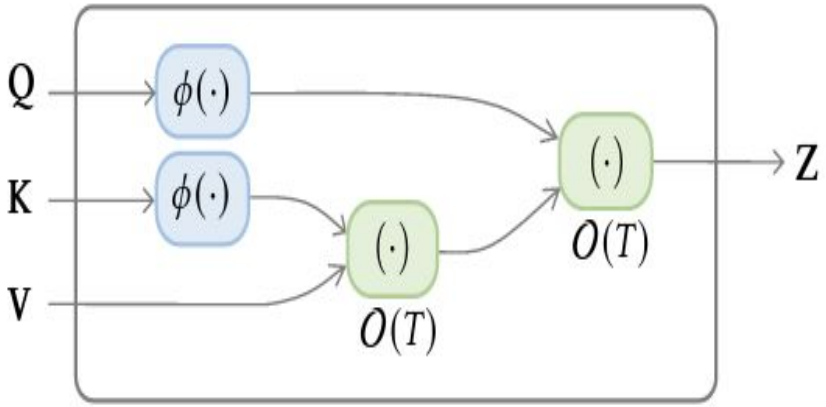
- Linearized Self-Attention



Linearized Attention



(a) standard self-attention



(b) linearized self-attention

Linearized Attention

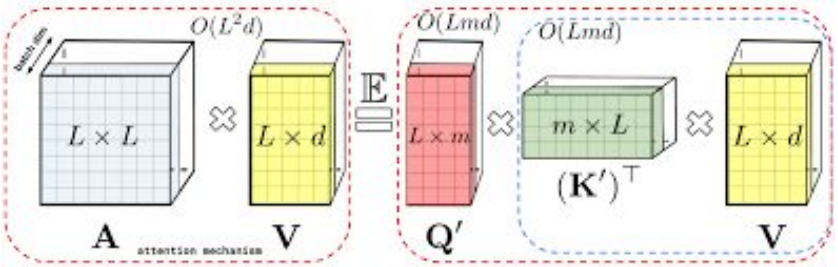
Decompose softmax(QK^T) into ($Q'K'^T$)

- Compute K'^TV
- Multiply by Q'
- Linear complexity in T

- Self-attention is now:

$$Z_i = \sum_j \frac{sim(q_j, k_j)}{\sum_{j'} sim(q_j, k_{j'})} \cdot v_j$$

- Get rid of quadratic complexity



Linearized Attention

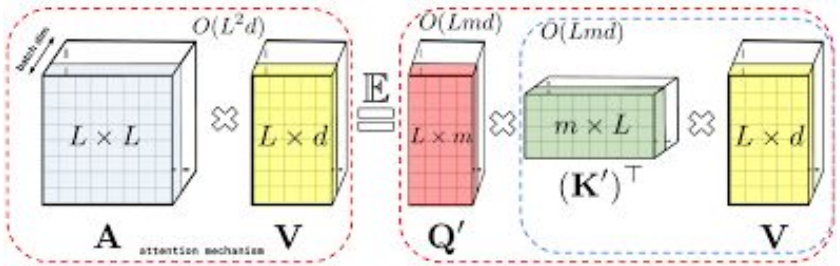
- Choose kernelized form

$$sim(x, y) = k(x, y) = \phi(x)\phi(y)^T$$

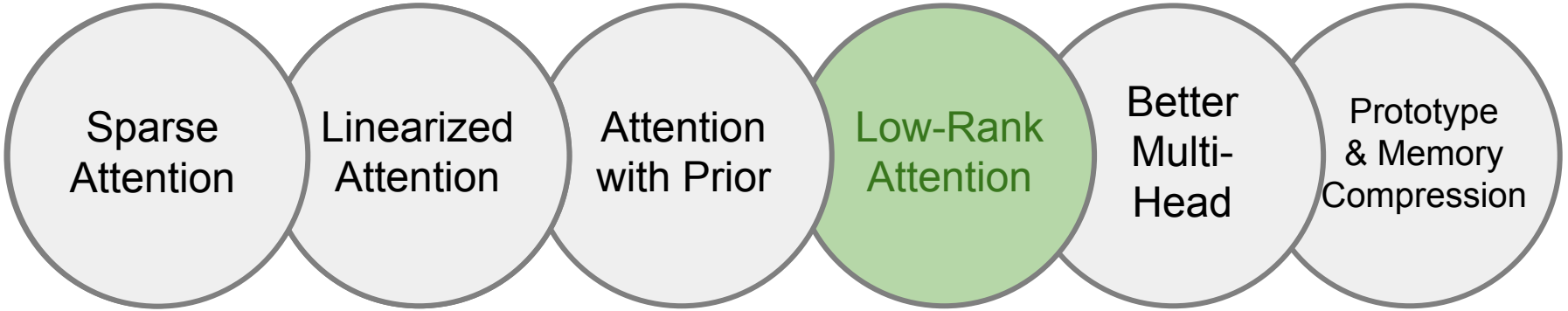
- Use “inverse kernel trick”

$$Z_i = \sum_j \frac{\phi(x)\phi(y)^T}{\sum_{j'} \phi(x)\phi(y)^T} \cdot v_j$$

$$Z_i = \frac{\phi(q_i) \sum_j \phi(k_j)^T \otimes v_j}{\phi(q_i) \sum_{j'} \phi(k_{j'})^T}$$



How to Improve the Attention Mechanism



Low-rank Self-Attention

- Self-attention matrix A (after softmax) is often low-rank
- We can decompose it!

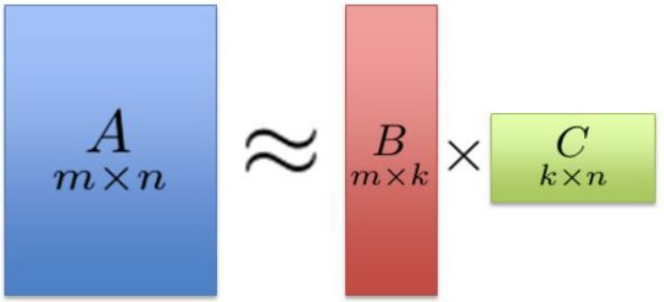
$$\begin{bmatrix} 1 & 2 & 3 \\ 1 & 2 & 3 \\ 1 & 2 & 3 \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} \begin{bmatrix} 1 & 2 & 3 \end{bmatrix}$$

Low-rank Self-Attention

- Model A with low-rank parameterization or replace it by a low-rank approximation

$$\underset{m\times n}{A} \approx \underset{m\times k}{B}\ \underset{k\times n}{C}$$

where $k \ll \min\{m, n\}$



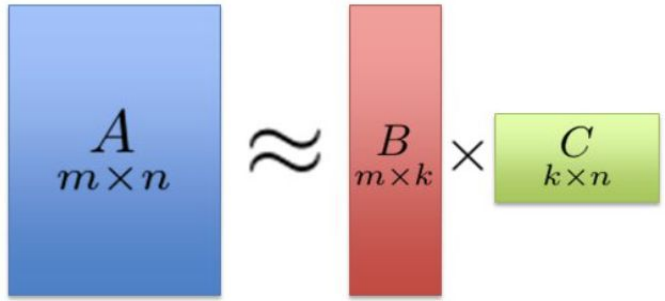
Low-Rank Parameterization/Approximation

- If rank(A) < sequence length T for small T ⇒ embedding dimension D > T may lead to overfitting (and not just overparameterization)

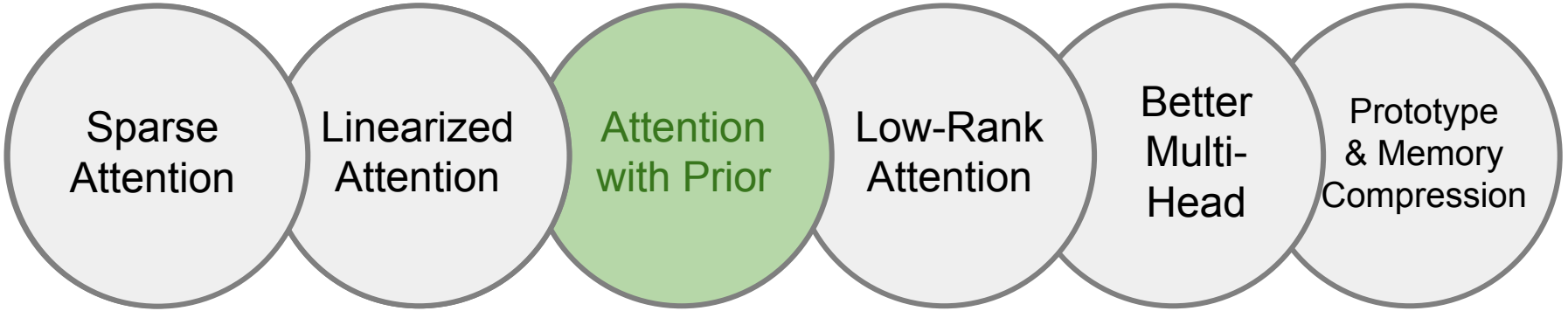
- Limit the dimension of D to explicitly model self-attention low-rank property as an inductive bias

$$\underset{m\times n}{A} \approx \underset{m\times k}{B}\ \underset{k\times n}{C}$$

where $k \ll \min\{m, n\}$



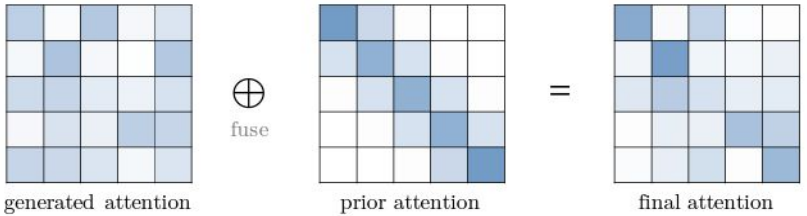
How to Improve the Attention Mechanism



Attention with Prior

Attention = weighted sum of input tokens representations; weights “describe relationships between tokens”.

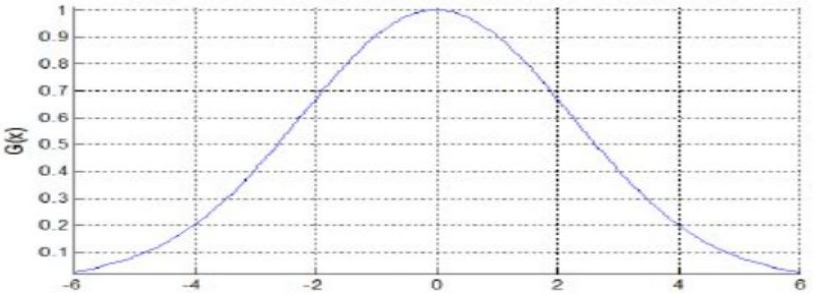
Attention weights distribution can also come from other sources/assumptions/priors



Attention with Prior: Locality-Anchored

- Data with **strong locality properties** is **“explicitly encoded** as” a prior attention
  - Multiply generated attention scores by 1D Gaussian density filter coeffs

- **Attention + Bias G:** high Gij indicates a high prior probability of “stronger connection” between i-th and j-th inputs



Distance between i-th and j-th token

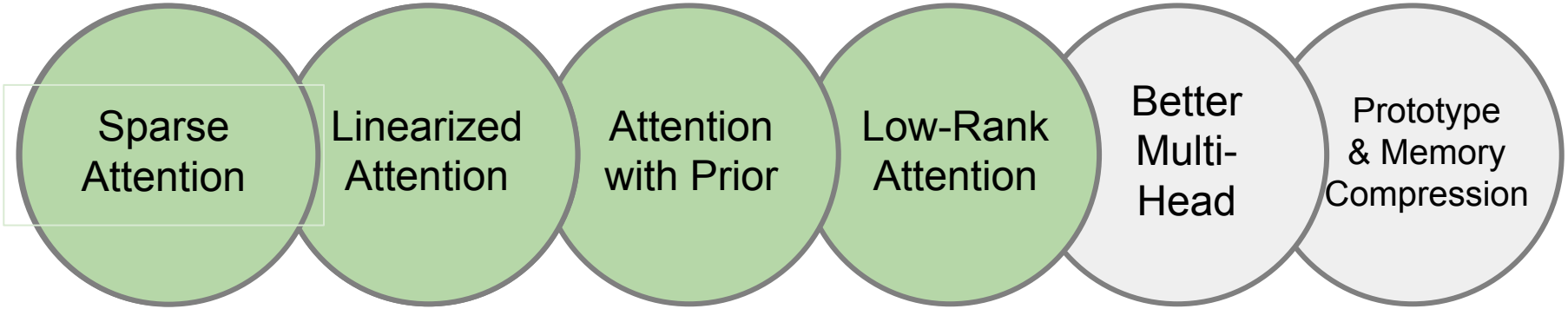
Attention with Prior: Multi-task Adapted Priors

- **Adapters** - task-dependent, trainable modules attached in pretrained network for cross-task efficient parameter sharing

$$M(\mathbf{z}_i) = \bigoplus_{j=1}^{m} A'_j(\mathbf{z}_i), \quad A'_j(\mathbf{z}_i) = A_j \gamma_i(\mathbf{z}_i) + \beta_i(\mathbf{z}_i),$$

- **Trainable attention prior** $M(\text{z\_}i)$ that depends on task encoding z_i

How to Improve the Attention Mechanism



Building Blocks Improvements: FC & Normalizatioln

✅ Activation Function Modifications (GeLU, ELU etc)

✅ Increased Capacity (key-memory based, mixture of experts, fixed attention)

✅ Dropping FC layer (for decoder)

✅ Normalization Layer Placement in the Transformer

✅ Different Variants of Norm Layer (group norm, L2-norm, normalization free)

Building Blocks Improvements: Positional Encoding

✅ Absolute Positional Encoding (encode actual token location)

✅ Relative Positional Encoding (encode token locations relative to each other)

✅ Encode token positions together with Token Embedding

Global-Level Improvements

✅ Lightweight transformers (less parameters, simpler architecture)

✅ Cross-Block Connectivity Between Different Transformer Layers

✅ Adaptive Computation Time (work hard only if for difficult data)

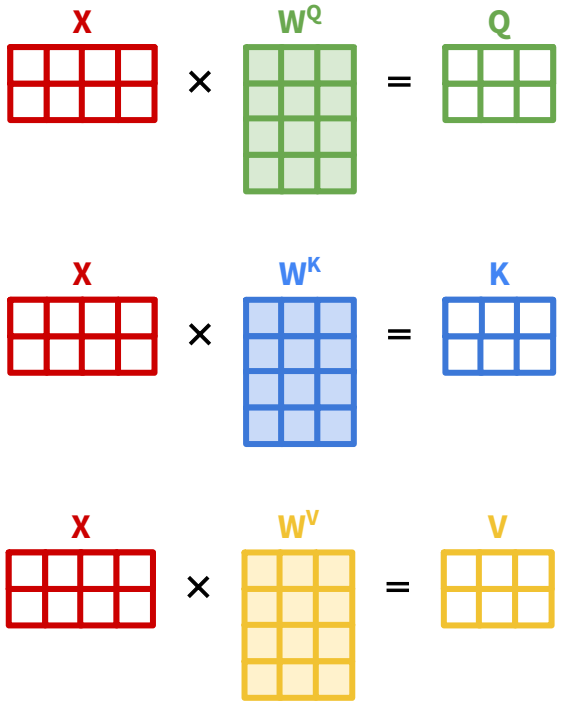
✅ Returning Recurrency to the Life

Do We Need Attention At All?

Do We Need Attention At All?

Transformer's Revolution Reasons: an attention mechanism (seasoned with FF and res-connections sauce)

Can we **ACHIEVE** similar performance **WITHOUT** attention mechanism?



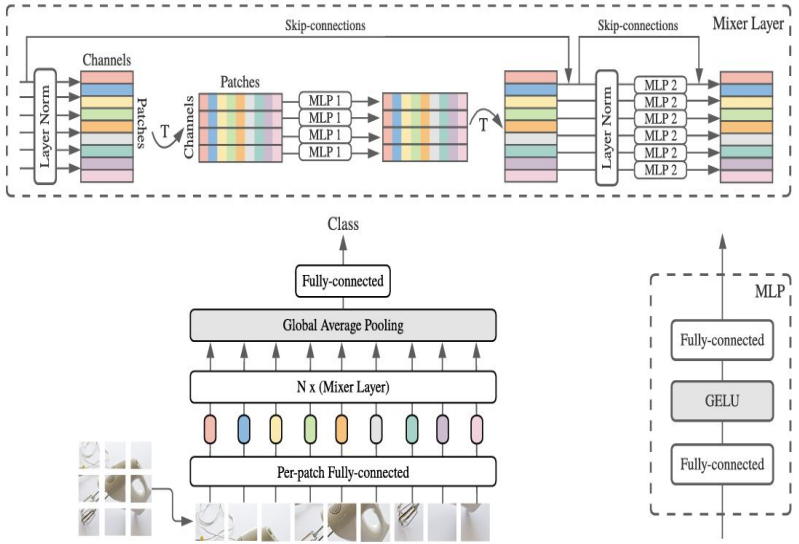
Attention Replacements

How to flow info in a network?

MLP Mixer (Pay Attention to MLPs), 2021

Stack a Bunch of FC Layers

Token-mixing MLP & channel-mixing MLP, each consisting of 2 FC layers & a GELU



MLP Mixer (Pay Attention to MLPs), 2021

Stack a Bunch of FC Layers

Token-mixing MLP & channel-mixing MLP, each consisting of 2 FC layers & a GELU

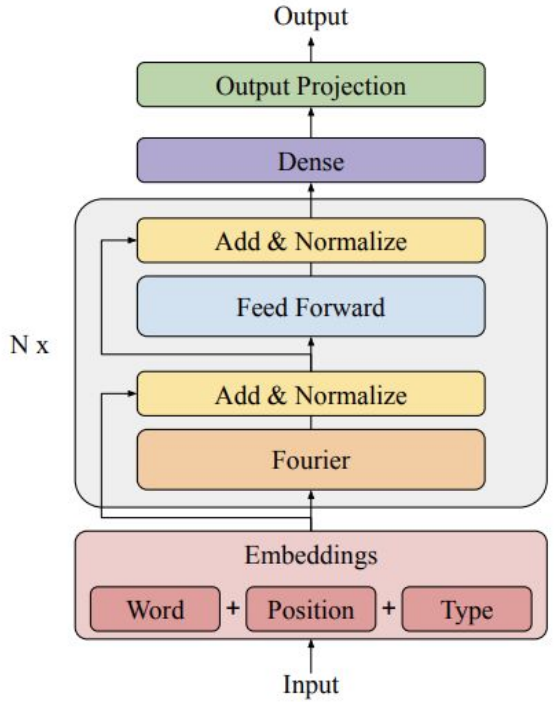
	ImNet top-1	ReaL top-1	Avg 5 top-1	VTAB-1k 19 tasks	Throughput img/sec/core	TPUv3 core-days
Pre-trained on ImageNet-21k (public)						
HaloNet [51]	85.8	—	—	—	120	0.10k
Mixer-L/16	84.15	87.86	93.91	74.95	105	0.41k
ViT-L/16 [14]	85.30	88.62	94.39	72.72	32	0.18k
BiT-R152x4 [22]	85.39	—	94.04	70.64	26	0.94k
Pre-trained on JFT-300M (proprietary)						
NFNet-F4+ [7]	89.2	—	—	—	46	1.86k
Mixer-H/14	87.94	90.18	95.71	75.33	40	1.01k
BiT-R152x4 [22]	87.54	90.54	95.33	76.29	26	9.90k
ViT-H/14 [14]	88.55	90.72	95.97	77.63	15	2.30k
Pre-trained on unlabelled or weakly labelled data (proprietary)						
MPL [34]	90.0	91.12	—	—	—	20.48k
ALIGN [21]	88.64	—	—	79.99	15	14.82k

FNet, 2021

2 discrete Fourier transforms over 2 different dimensions

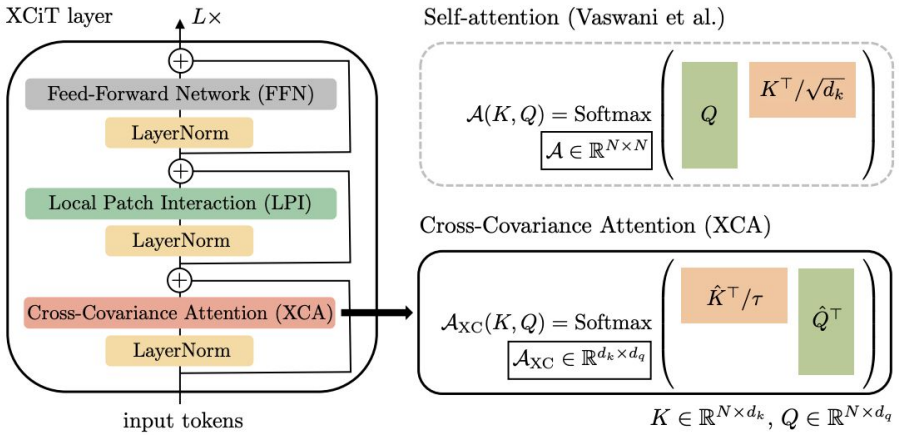
Mix over sequence dim & over the embedding dim

$$y = \Re\left(\mathcal{F}_{\text{seq}}\left(\mathcal{F}_{\text{h}}(x)\right)\right)$$



XCiT: Cross-Covariance Image Transformers, 2021

Attention across feature channels rather than tokens

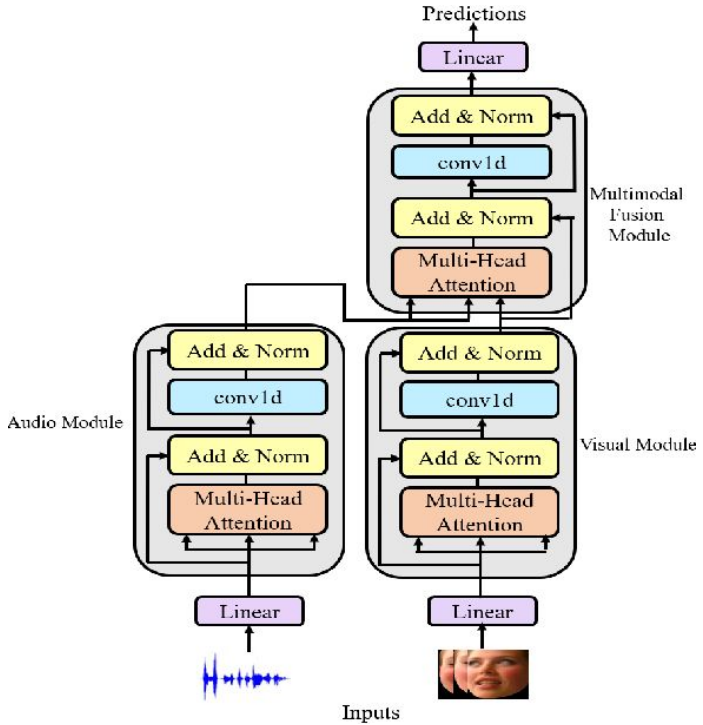


Future Research Directions

Unified Framework for Multimodal Data

Integrating multimodal data is proven to be useful for task performance improvement

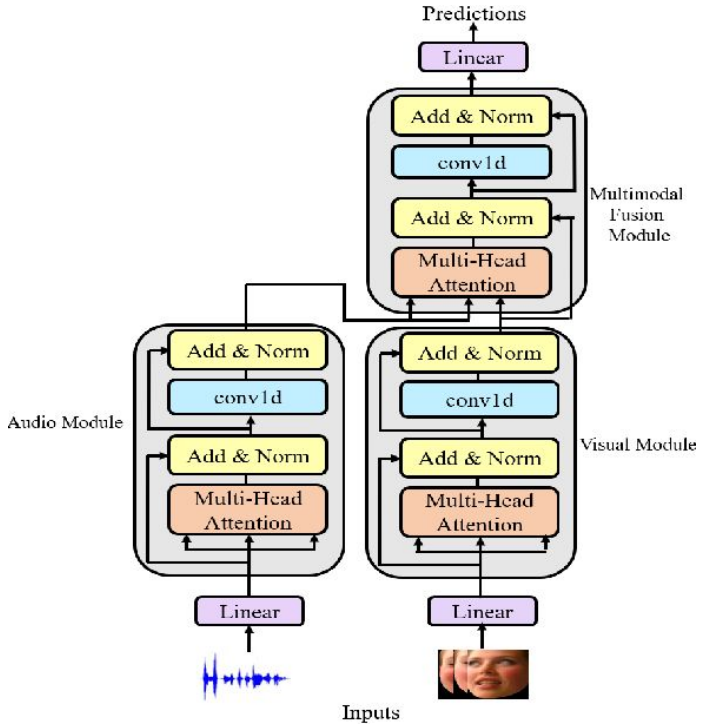
General AI needs to capture the semantic relations across different modalities



Unified Framework for Multimodal Data

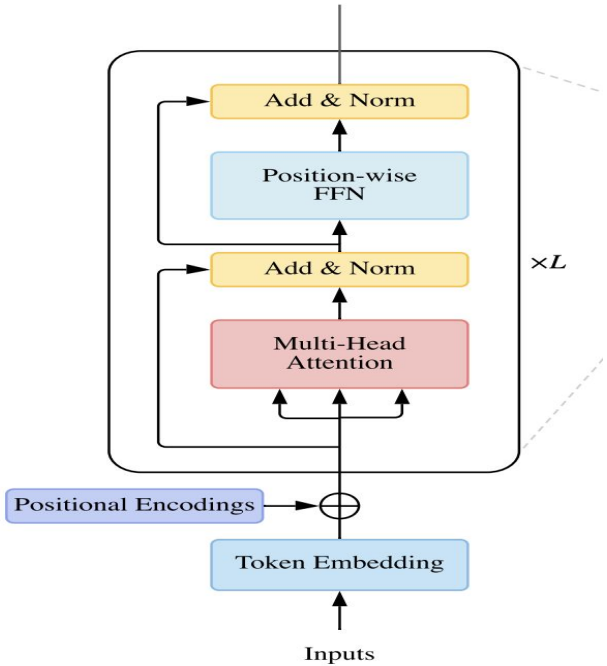
Transformers excel in text, image, video & audio domains

What about building unified framework fully leveraging interconnections of multimodal data?



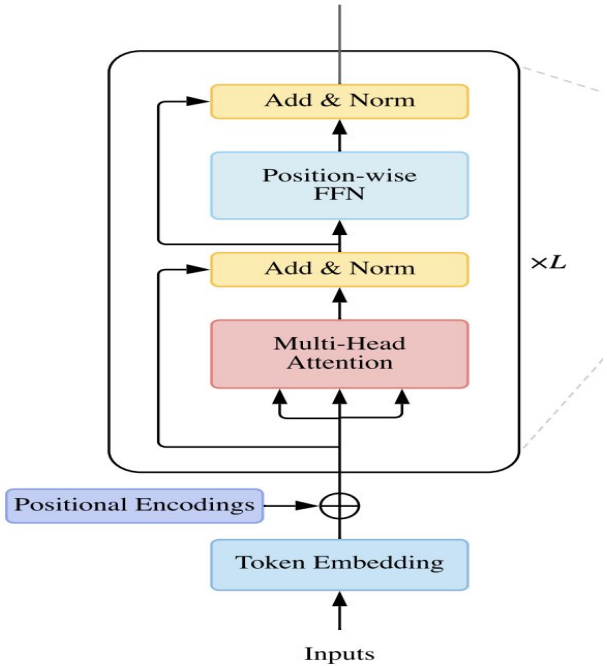
Theoretical Analysis - Transformers Capabilities

Transformers “empirically” shown to have a larger capacity (expressiveness) than CNNs/RNNs & hence to effectively “learn” huge amount on data



Theoretical Analysis - Transformers Capabilities

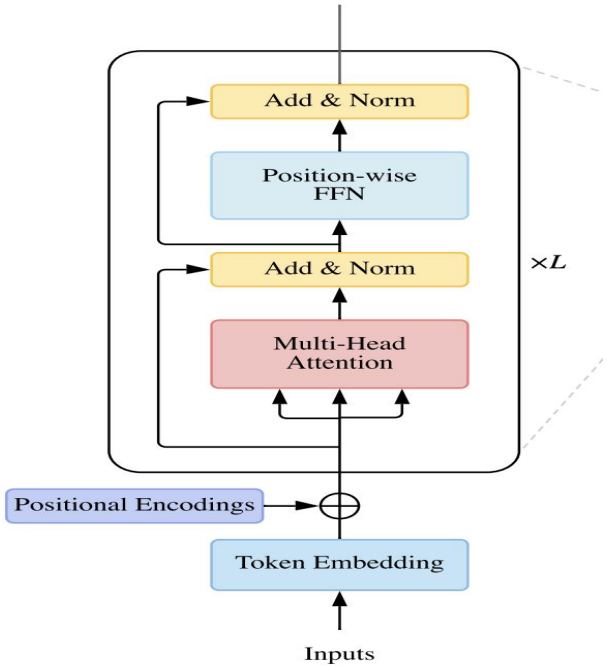
When Transformer trained on sufficiently large and diverse data, its performance is better than CNNs/RNNs.



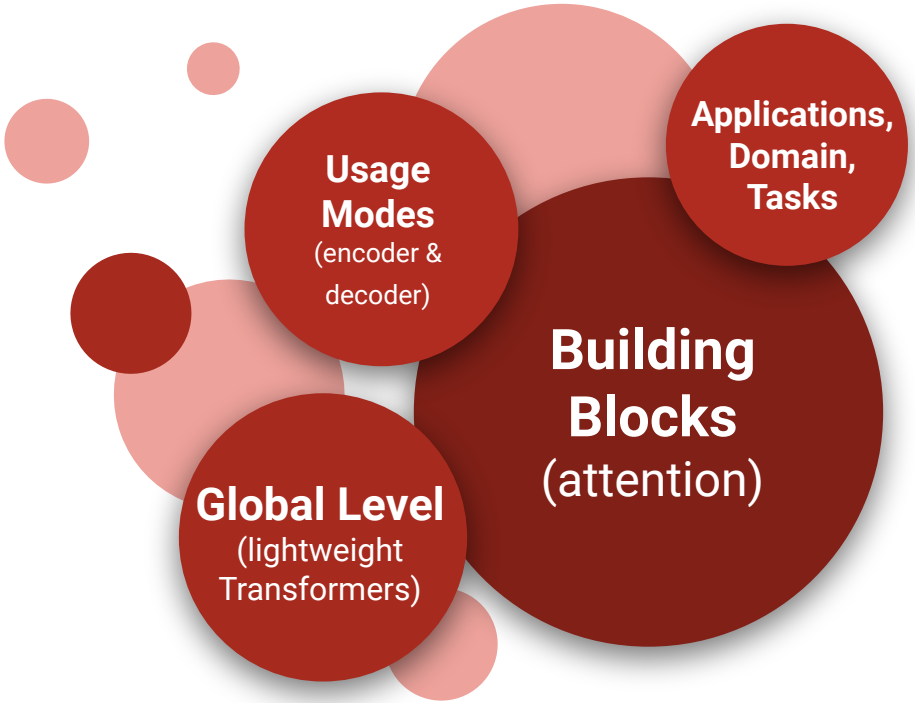
Theoretical Analysis - Transformers Capabilities

Transformer makes NO assumptions on the data structure and thus is more flexible?

But we still **DO NOT** have a Full Theoretical Explanation for the Transformer Power



More to Improve, But it's Out of Scope for Today.



So What did We See Today?

2017 grabbed much attention, but in fact much of the concepts were here already

The new concepts introduced in 2017 managed to move the industry forward

There are several directions to improve at

Transformers influence other domains such as cybersecurity, genetics and chemistry

Attention can improve

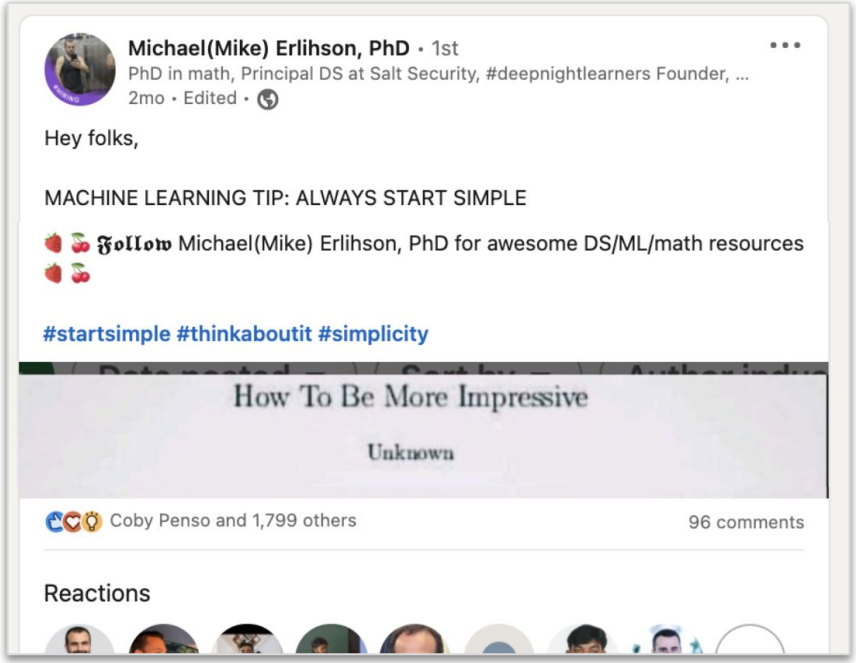
We can build transformers with no attention at all

Thanks!

Keep in touch:
Mike Erlihson, PhD

michael-mike-erlihson-phd

maerlich



Transformers: Never Ending Story

Mike Erlihson, PhD
Principal Data Scientist, Salt Security

Presentation Credits: Hila Paz H.

Agenda for Mike & Hila - Delete Later

Intro (~4 minutes)

1) Why is 2017 so important in the transformers' history

Part 1 - Transformers at 2017 (10 minutes)

1) What's the same (attention = generalization of fcn)
2) What's changed
    a) No RNN / feed the entire sequence
    b) Multi-head attention (not so important, maybe one slide)
3) What's not there yet (at 2017) - one slide, opening for part two
    a) Full attention, Applications are majorly present at NLP domain (teaser for cybersecurity and other domains)

Part 2 - Problems and Changes (15 minutes)

4) Applications (e.g. In Cyber) - 4

5) Full attention - no more
    a) Sparse attention (5 minutes)
    b) Linearized attention (performer - a classical research + QR code) (3 minutes)
    c) Replacing attention by other mechanisms (3 minutes)

Part 3 - Speed Overview (5 minutes)

6) Future research directions
    a) Multi-mode (video, speech, image captioning..)

Multi-Modal Transformers

Tackle task with multi-model data:

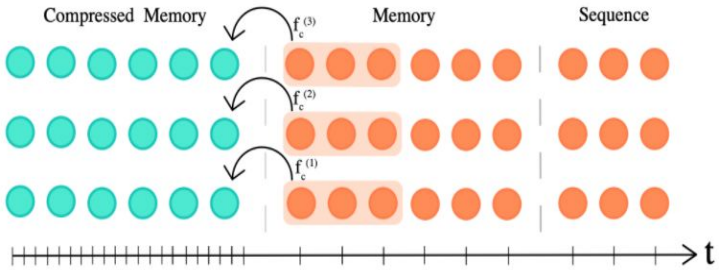
(e.g. visual question answering, OCR)

Multi-modal Transformer for Video Retrieval, 2020

Multimodal Transformer for Unaligned Multimodal Language Sequences, 2019

Prototype & Memory Compression

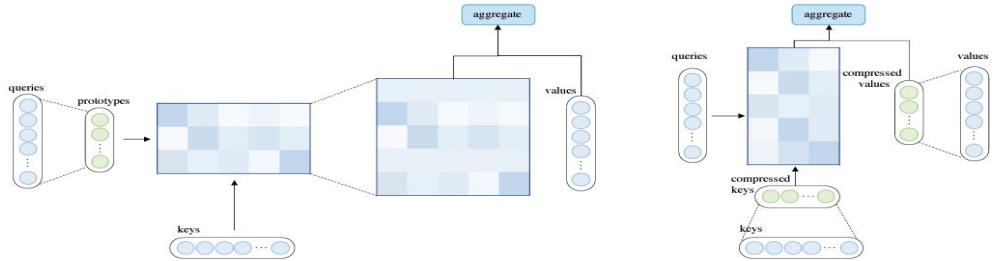
Reduces number of queries (Q) or/and key-value (KV) memory pairs to reduce attention matrix dimension



Query Prototyping & Memory Compression

Query Prototyping: compute attention vectors with several prototypes of Q-s

Clustered Attention: group Q-s into clusters & compute attention vector for cluster centroids. All cluster's Q-s share the attention vector of their centroid.



(a) Query prototyping (b) Memory compression

Fig. 8. Query prototyping and memory compression.

Multi-Head Attention: Recap

- Each head captures different contextual info by mixing tokens in a unique manner
- Expands the model's ability to focus on different positions
- Translating "The **animal** didn't cross the street because it was too **tired**", we want to know which word **"it"** refers to

Multi-Head Attention: But What About Implementation

- Gives the attention layer multiple “representation subspaces”
- Multiple sets of (Q,K,V) matrices
- Each (Q,K,V) projects token embeddings (vectors from lower encoders) into a different representation subspace

$$\text{MultiHead}(\mathbf{Q}, \mathbf{K}, \mathbf{V}) = [\text{head}_1, \ldots, \text{head}_h]\mathbf{W}_0$$

$$\text{where head}_i = \text{Attention}\left(\mathbf{Q}\mathbf{W}_i^Q, \mathbf{K}\mathbf{W}_i^K, \mathbf{V}\mathbf{W}_i^V\right)$$

Multi-Head Attention: Main Issue

No **guarantee** that different heads indeed capture **distinct features**

Head Behavior Modeling: Research Directions

Mechanisms guiding different attention heads behavior

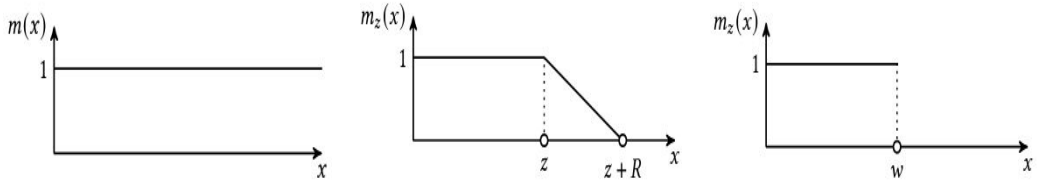
Restricted Attention: some heads focus attention distribution mainly in a **local context** while some others attend to **broader contexts**.

Interaction across multiple attention heads (**mix** instead **composition)**

Head Behavior Modeling: Restrict Attention Span

Locality: Induces explicit local constraints - best suited for data with locality prior

Efficiency: Scales to very long sequences w/o requiring additional memory footprint & computational time.



(a) mask function for vanilla attention (b) mask function for adaptive span (c) mask function for fixed span

Attention Replacement 3

XCiT: Cross-Covariance Image Transformers, 2021 - Attention is Replaced by:

- "Transposed" self-attention operating across feature channels rather than tokens
- Interactions are based on the cross-covariance matrix between keys and queries
- Has linear complexity in #tokens; efficient processing of high-resolution images