

Visual Transformers and Transformers-XL

Jack Lipson + Arin Manohar

Math 270: Survey of Deep Learning for Mathematicians

October 20, 2025

What's Lacking with CNN/RNN

CNNs incorporate inductive bias into architecture, then train for parameters.

This bias is useful if we are doing supervised learning and have small data sets.

But, this is restrictive with **larger sets**.

RNNs add in (long) short-term memory to learn (but training hidden nodes **takes time**).

Sequence modelling also precludes parallelization.

--- --- ---

Transformers give a **flexible** alternative, with **shorter training time** on **larger sets**.

Attention is All You Need

Nixes recurrence.

Counterintuitive b/c incorporates less inductive bias.

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
lukaszkaizer@google.com

Illia Polosukhin* †
illia.polosukhin@gmail.com

Dot-product Self-Attention

input tokens in embedding space

↓ (attention)

tokens in embedding space with richer semantic structure

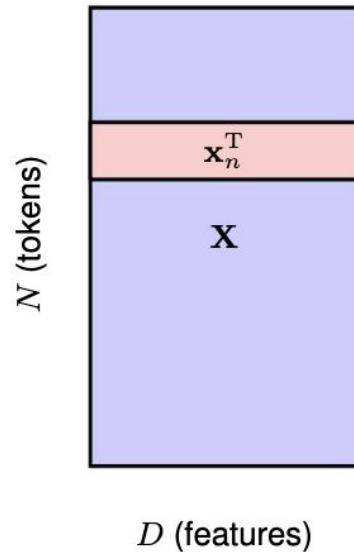
Easiest attempt is $y_n = a_n \text{ dotprod } x_m$.

(a_n in $[0,1]$ and $\sum a_{\{nm\}} = 1$)

"attention coefficients" should depend on input data.

$$a_{nm} = \frac{\exp(\mathbf{x}_n^T \mathbf{x}_m)}{\sum_{m'=1}^N \exp(\mathbf{x}_n^T \mathbf{x}_{m'})}$$

$$\mathbf{Y} = \text{Softmax} [\mathbf{X} \mathbf{X}^T] \mathbf{X}$$



Bishop, §12

Dot-product Improvements

1. Modify feature vectors ($X' = X U$ w/ U learnable $D \times D$).

$$Y = \text{Softmax} [X U U^T X^T] X U.$$

2. Above transformer matrix (w/o softmax) is symmetric...

Softmax still is not flexible \rightarrow we want to emphasize asymmetric attention weights (e.g. tool vs. chisel).

Solution via 3 trainable weight matrices query + key + value.

$$Q = X W^{(q)}$$

$$K = X W^{(k)}$$

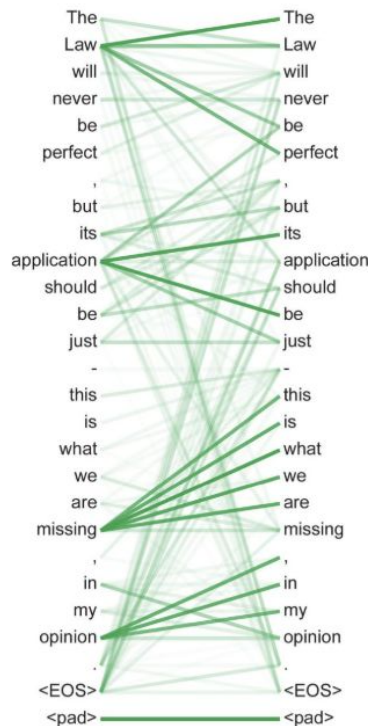
$$V = X W^{(v)}$$

(scaled to not kill subtleties in softmax)

$$Y = \text{Softmax} [Q K^T] V$$

Figure 12.2

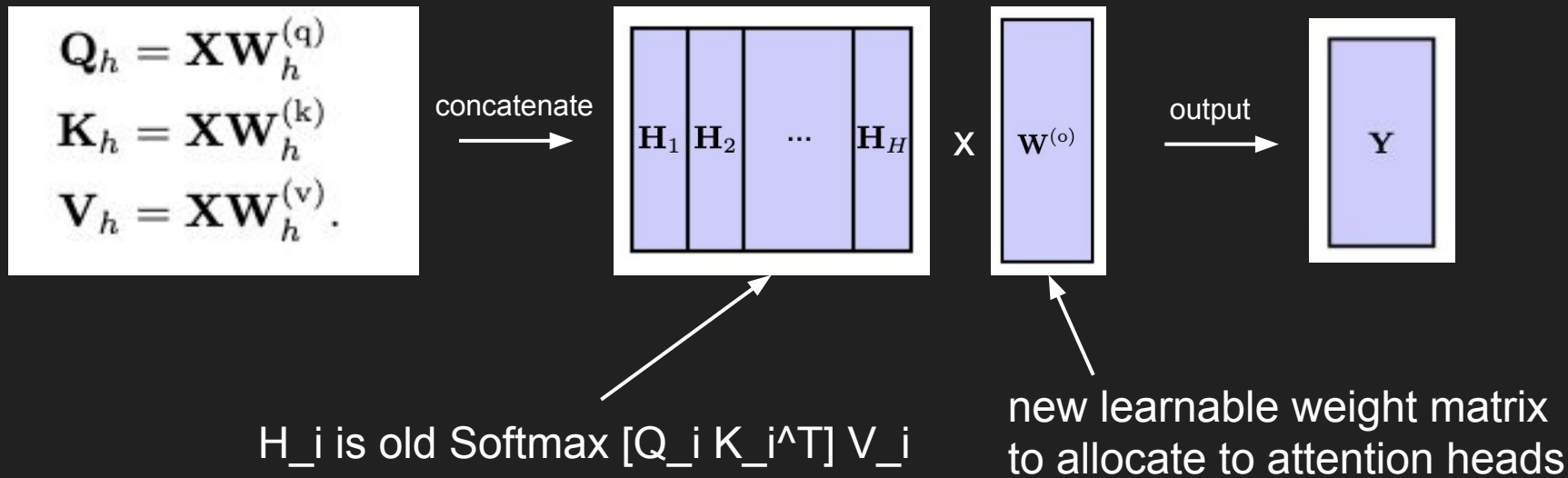
An example of learned attention weights. [From Vaswani et al. (2017) with permission.]



Vaswani, et. al

Multihead Self-Attention

Want to attend to different patterns (e.g. vocab vs. tense in NLP)



Final Transformer Architecture

Even with softmax, multi-head structure is mostly linear

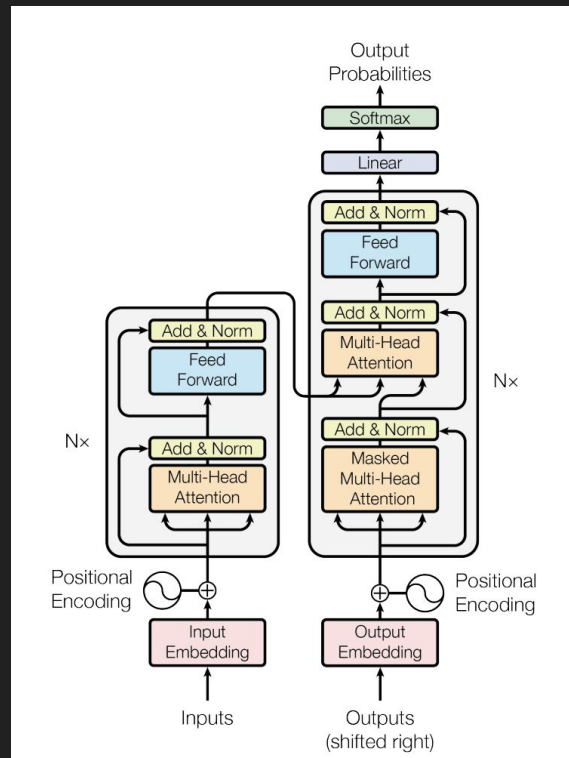


Post-compose multilayer perceptron (MLP, "feed-forwards") nonlinear neural networks (ReLU activation fns) for expressiveness



Normalize

(since our transformer is unfortunately permutation-equivariant wrt. the inputs, we add position encoding vectors r_n to each x_n (where the r_n are likely almost orthogonal since the embedding space's dimension is so large))



Complexity of Base Transformer

Layer Type	Complexity per Layer	Sequential Operations	Maximum Path Length
Self-Attention	$O(n^2 \cdot d)$	$O(1)$	$O(1)$
Recurrent	$O(n \cdot d^2)$	$O(n)$	$O(n)$
Convolutional	$O(k \cdot n \cdot d^2)$	$O(1)$	$O(\log_k(n))$
Self-Attention (restricted)	$O(r \cdot n \cdot d)$	$O(1)$	$O(n/r)$

n = # of input tokens, each of dimension (length) d

Success of Base Transformer ("bigger models are better")

Table 2: The Transformer achieves better BLEU scores than previous state-of-the-art models on the English-to-German and English-to-French newstest2014 tests at a fraction of the training cost.

Model	BLEU		Training Cost (FLOPs)	
	EN-DE	EN-FR	EN-DE	EN-FR
ByteNet [18]	23.75			
Deep-Att + PosUnk [39]		39.2		$1.0 \cdot 10^{20}$
GNMT + RL [38]	24.6	39.92	$2.3 \cdot 10^{19}$	$1.4 \cdot 10^{20}$
ConvS2S [9]	25.16	40.46	$9.6 \cdot 10^{18}$	$1.5 \cdot 10^{20}$
MoE [32]	26.03	40.56	$2.0 \cdot 10^{19}$	$1.2 \cdot 10^{20}$
Deep-Att + PosUnk Ensemble [39]		40.4		$8.0 \cdot 10^{20}$
GNMT + RL Ensemble [38]	26.30	41.16	$1.8 \cdot 10^{20}$	$1.1 \cdot 10^{21}$
ConvS2S Ensemble [9]	26.36	41.29	$7.7 \cdot 10^{19}$	$1.2 \cdot 10^{21}$
Transformer (base model)	27.3	38.1	$3.3 \cdot 10^{18}$	
Transformer (big)	28.4	41.8	$2.3 \cdot 10^{19}$	

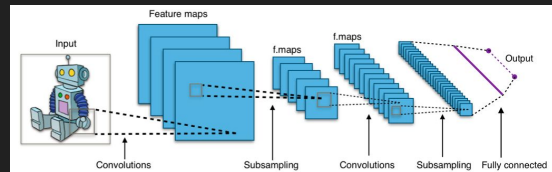
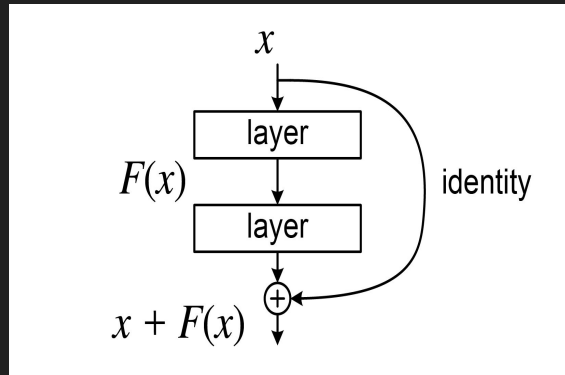


Vision Transformer (Motivation)

Pre-Transformer, state of the art for image processing: **ResNet: CNNs + Skip connection**

Advantages:

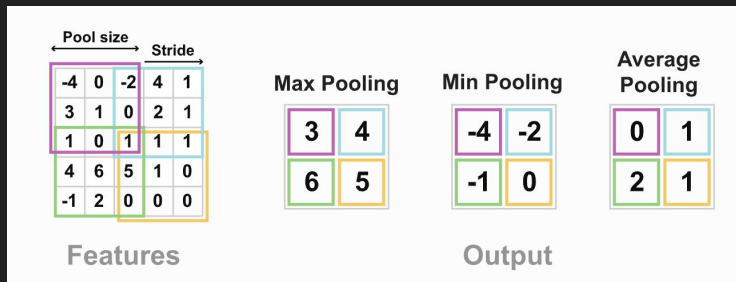
- Less weights -> More data efficient
- CNNs were tailored for images
 - “Inductive biases”:
 - Equivariance - $f(Tx) = Tf(x)$
 - Locality - pixels in neighborhood more important than pixels far away
- With residual connections:
 - Deep CNNs with hierarchical feature learning + addresses vanishing/exploding gradient problem



Vision Transformer (Motivation)

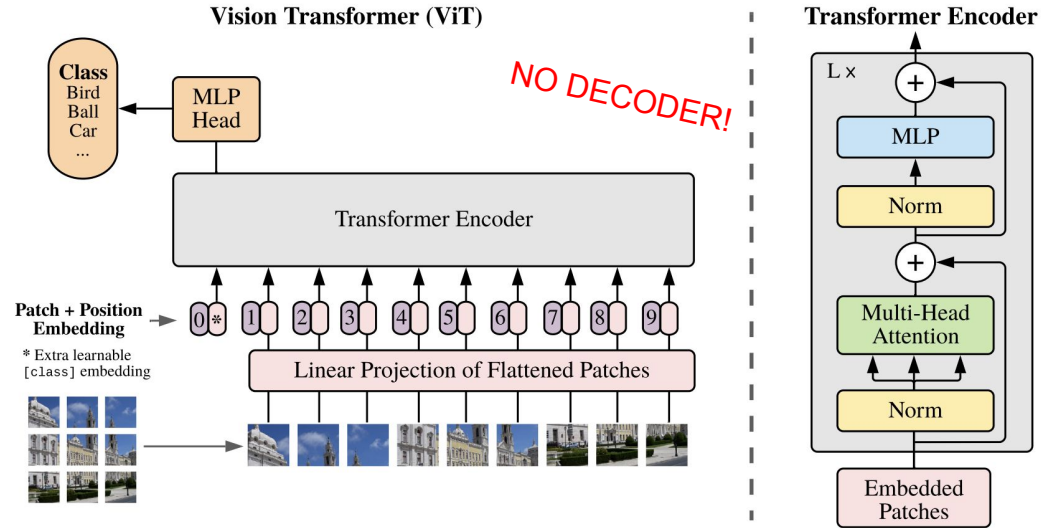
But...

- Hard to Scale
 - Pooling, Striding needs careful tuning
 - ResNets only help up to a point
- Inductive bias is a bottleneck for big supervised data



It would be great if we could have a scalable and more generalizable architecture for big data image tasks...

Vision Transformer (Model)



Vision Transformer (Model)

Big Idea:

- Image Preprocessing -> Standard Transformer Encoder

Preprocessing:

- Break image into small square patches of size P . Patches = Tokens

Input to Transformer: For original image of (H, W, C) , patch size P

- Flattened Patch embeddings: $N_patches = HW/P^2$, Patch dim = P^2C
- x_class token: A learnable embedding that draws attention from all patches in order to make a classification at output layer L
- Position Embeddings: Encodes spatial position of each patch relative to other patches

Vision Transformer (Model)

$$\mathbf{z}_0 = [\mathbf{x}_{\text{class}}; \mathbf{x}_p^1 \mathbf{E}; \mathbf{x}_p^2 \mathbf{E}; \dots; \mathbf{x}_p^N \mathbf{E}] + \mathbf{E}_{\text{pos}}, \quad \mathbf{E} \in \mathbb{R}^{(P^2 \cdot C) \times D}, \mathbf{E}_{\text{pos}} \in \mathbb{R}^{(N+1) \times D} \quad (1)$$

$$\mathbf{z}'_\ell = \text{MSA}(\text{LN}(\mathbf{z}_{\ell-1})) + \mathbf{z}_{\ell-1}, \quad \ell = 1 \dots L \quad (2)$$

$$\mathbf{z}_\ell = \text{MLP}(\text{LN}(\mathbf{z}'_\ell)) + \mathbf{z}'_\ell, \quad \ell = 1 \dots L \quad (3)$$

$$\mathbf{y} = \text{LN}(\mathbf{z}_L^0) \quad (4)$$

- \mathbf{z}_i = output to transformer encoder
 - \mathbf{z}_i^0 = the classification token
- \mathbf{E} = learnable embedding projection matrix
- MSA = Multi-Headed Self Attention (Each Head attends to a feature in parallel)
- MLP = Multilayer Perceptron (FC + Nonlinear activation)
- LN = Layer Norm (for bad gradients and stability)

Vision Transformer (Model)

Summary of Learnable params

- X_{class} – token
- E – embedding projection matrix
- E_{pos} – embedding matrix for position
- Transformer encoder weights (L layers)
 - W_k, W_Q, W_v, W_0
 - MLP
 - Layer Norms
- MLP Head

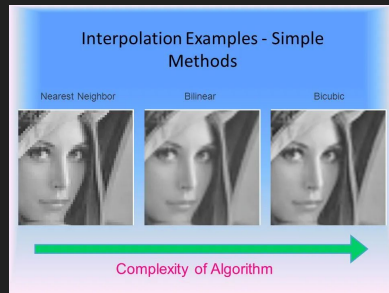
Vision Transformer (Training)

Supervised Pretrain:

- General training for a dataset
- Low resolution images

Finetuning:

- Specialized training for smaller specialized tasks
- Replace original classification MLP head with custom MLP head for new problem
- Higher resolution data with same patch size
- Interpolate 2D position encodings
 - You resize the old positional encodings from $14 \times 14 \rightarrow 24 \times 24$ using 2D interpolation (like resizing an image).



Vision Transformer (Experiments)

Model	Layers	Hidden size D	MLP size	Heads	Params
ViT-Base	12	768	3072	12	86M
ViT-Large	24	1024	4096	16	307M
ViT-Huge	32	1280	5120	16	632M

Pretrain Various ViT models above on varying size datasets and compare to ResNet:

- ImageNet (1k classes, 1.3M images)
- ImageNet-21k (21k classes, 14M images)
- JFT (18k classes, 303M HiRes images)

Vision Transformer (Experiment 1)

Goal: Classify images, compare accuracy between ViT and previous SOTA models

Vision Transformer (Experiment 1)

Results:

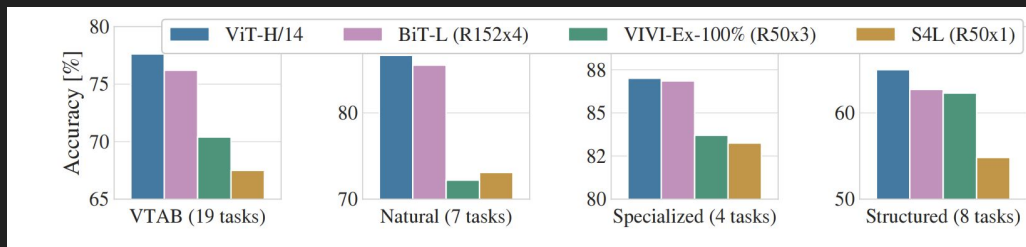
Huge/P=14

Large/P=16

- Huge ViT achieves better accuracy than SOTA on training and takes less compute

	Ours-JFT (ViT-H/14)	Ours-JFT (ViT-L/16)	Ours-I21k (ViT-L/16)	BiT-L (ResNet152x4)	Noisy Student (EfficientNet-L2)
ImageNet	88.55 ± 0.04	87.76 ± 0.03	85.30 ± 0.02	87.54 ± 0.02	88.4/88.5*
ImageNet ReaL	90.72 ± 0.05	90.54 ± 0.03	88.62 ± 0.05	90.54	90.55
CIFAR-10	99.50 ± 0.06	99.42 ± 0.03	99.15 ± 0.03	99.37 ± 0.06	—
CIFAR-100	94.55 ± 0.04	93.90 ± 0.05	93.25 ± 0.05	93.51 ± 0.08	—
Oxford-IIIT Pets	97.56 ± 0.03	97.32 ± 0.11	94.67 ± 0.15	96.62 ± 0.23	—
Oxford Flowers-102	99.68 ± 0.02	99.74 ± 0.00	99.61 ± 0.02	99.63 ± 0.03	—
VTAB (19 tasks)	77.63 ± 0.23	76.28 ± 0.46	72.72 ± 0.21	76.29 ± 1.70	—
TPUv3-core-days	2.5k	0.68k	0.23k	9.9k	12.3k

- Huge ViT does better on assorted visual tasks in VTAB



VTAB includes image classification of natural images (flowers, cats), niche images (satellites and aerial photographs), structured tasks (count how many dots)

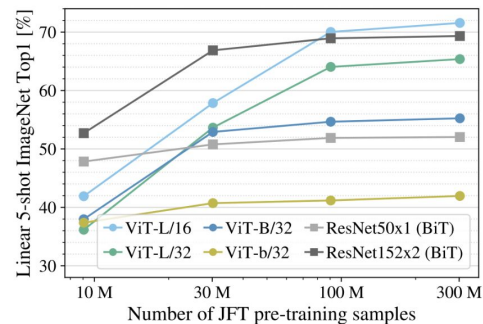
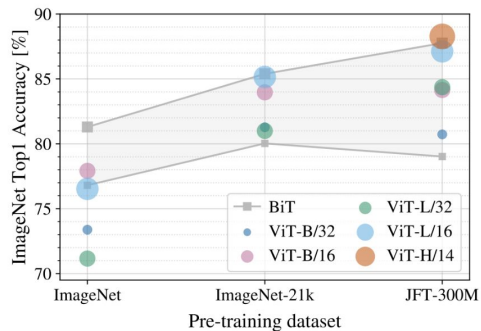
Vision Transformer (Experiment 2)

Goal: Compare performance of ViT with varying model size and sample size

Vision Transformer (Experiment 2)

Results:

- ViT H/14 outperforms ResNet on JFT
- ResNet saturates earlier, ViT eventually outperforms



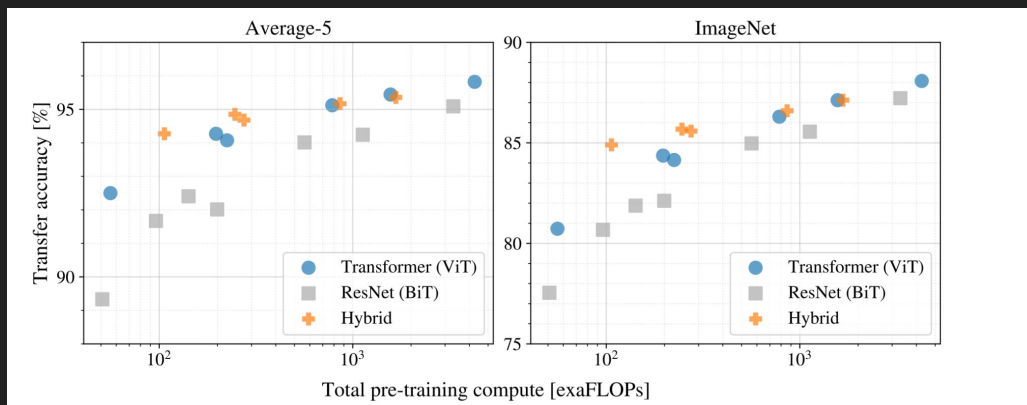
Vision Transformer (Experiment 3)

Goal: Compare Accuracy vs Compute between ViT, ResNet, Hybrid

Vision Transformer (Experiment 3)

Results:

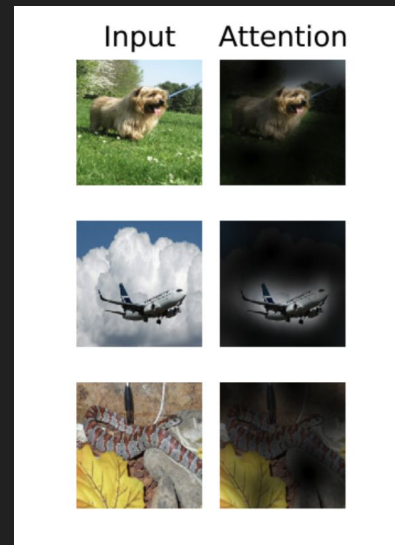
- Hybrid performs best in small compute budget
- At large compute budget, pure transformer wins



Vision Transformer (Inspection)

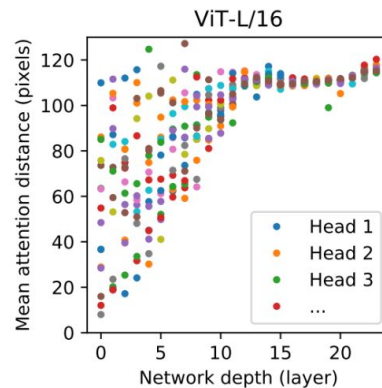
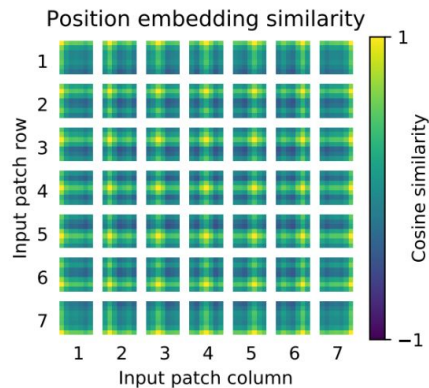
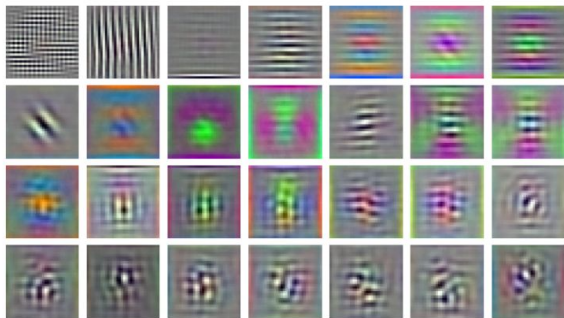
Heads (in MSA) attend to components of image which are highly relevant for classification.

- Attention is the same shape as input x , it “lights up” in patches of x with high attention



Vision Transformer (Inspection)

RGB embedding filters
(first 28 principal components)



Vision Transformer (Inspection)

1. Principal Components of Embedding Projection matrix
 - a. Each tile represents PC of E. These are reshaped (P, P) tiles with R, G, B overlaid
2. Embedding of Position per patch location
 - a. Tile = corresponds to one specific patch
 - b. Colors inside tile = cosine similarity to all other patch position embeddings
3. How far in pixel space does head look
 - a. In early layers, high locality, behaves like CNN
 - b. In deeper layers, global structure, looks farther

Vision Transformer (Conclusion)

- Transformers in Vision have shown many successes over ResNets
 - More computationally efficient
 - Comparable if not better accuracy
- Identifies features for classification purely through attention rather than convolutional layers
- Works better on larger supervised datasets
- Demonstrates the multimodality of transformer architecture, not just NLP but image classification as well

Transformer-XL (motivation)

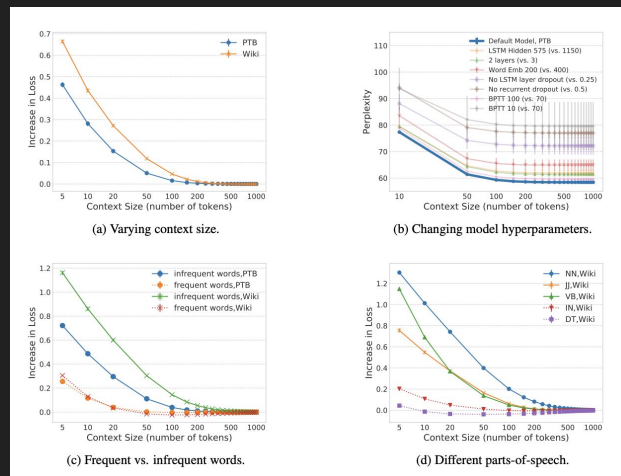
Recall self-attention complexity per layer is of order $O(n^2 * d)$.

As **context window length n** increases **linearly**, attention scales **quadratically**.


Seemingly, RNNs are better-suited because complexity is $O(n * d^2)$.

But, their **effective context** stabilizes:

Attention may give long-term dependency.



[Khandelwal, et. al]

Transformer-XL (first look, Al-Rfou)  same Vaswani, et. al block
A non-recurrent model (deep 64-layer transformer) outperforms RNNs (2018)

Character-Level Language Modeling with Deeper Self-Attention

Rami Al-Rfou*

Dokook Choe*

Noah Constant*

Mandy Guo*

Llion Jones*

Google AI Language

Character-level language modeling of natural language is very challenging:
(i) no vocab, (ii) dependencies, (iii) computation

RNNs use 200 token batches with hidden state from previous batch revealed.
(But we know this extra info is not effectively used by last slide.)

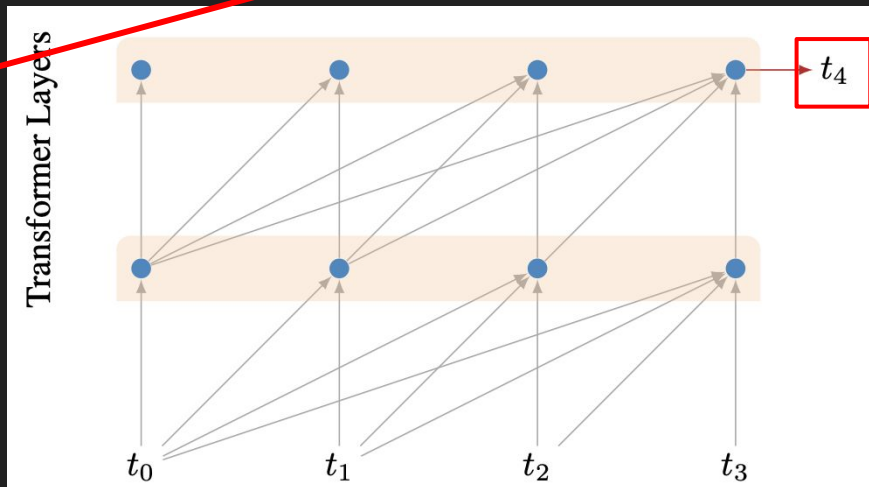
Al-Rfou, et. al's model used mini-batches at random starts w/ NO info passed on.

Transformer-XL (first look, AI-Rfou)

$$\Pr(t_{0:L}) = P(t_0) \prod_{i=1}^L \Pr(t_i | \underline{t_{0:i-1}}),$$

transformer processes

Attention layers are masked via causal attention (i.e. no leftward arrows).



transformer predicts

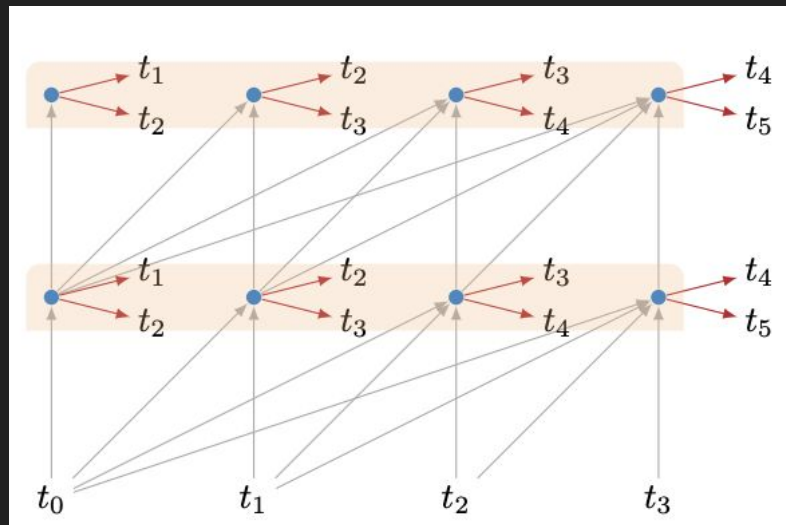
Transformer-XL (first look, AI-Rfou)

A few ~~experiments~~ improvements:

1. Auxiliary losses (w/ decay) resolved slow conv./accuracy (due to high (>10 layers) depth).

2. Added predictions (+ loss fns) at each t_i in final and intermediate layers (*red arrows*).
(This resembles what an RNN does.)
(*middle layers decay w/ lower layers faster*)

3. Vaswani, et. al's sinusoidal pos. embedding in layer 1 is replaced w/ a learned-per-layer pos. embedding since order information could be lost in propagation.



Transformer-XL (first look, success but limitation)

bits req. to
predict next
token

→ [

Model	Parameters ($\times 10^6$)		bpb
	train	inference	
FS-LSTM-4 (Mujika, Meier, and Steger 2017)	47	-	1.25
mLSTM (Krause et al. 2016)	46	-	1.24
cmix v13 (Knol 2017)	-	-	1.23
T12 (ours)	44	41	1.11
T64 (ours)	235	219	1.06
mLSTM + dynamic eval (Krause et al. 2017)	46	-	1.08

Table 3: Comparison of various models on `enwik8` test.

Al-Rfou, et. al's pos. embedding = 512-dim embedding for L positions * N layers

$L * N * 512$ added parameters!

"context fragmentation"

Only possible since model does not accept longer contexts than $L \rightarrow$ need XL.

^{"extra-long"} Transformer-XL (Dai, et. al)

"[...] although the self-attention mechanism is less affected by the vanishing gradient problem compared to RNNs, the [AI-Rfou] model is not able to fully exploit this optimization advantage." - [Dai, et. al]

AI-Rfou, et. al's model is limited because:

- 1) largest dependency is batch segment length from training,
- 2) chunking wrt. semantic boundaries (".", ":", etc.) still leaves out longer context and is surprisingly *less* efficient than careless selection.
- 3) it adds one position at a time, so new segment is gen. from scratch each iter.
(computationally expensive)

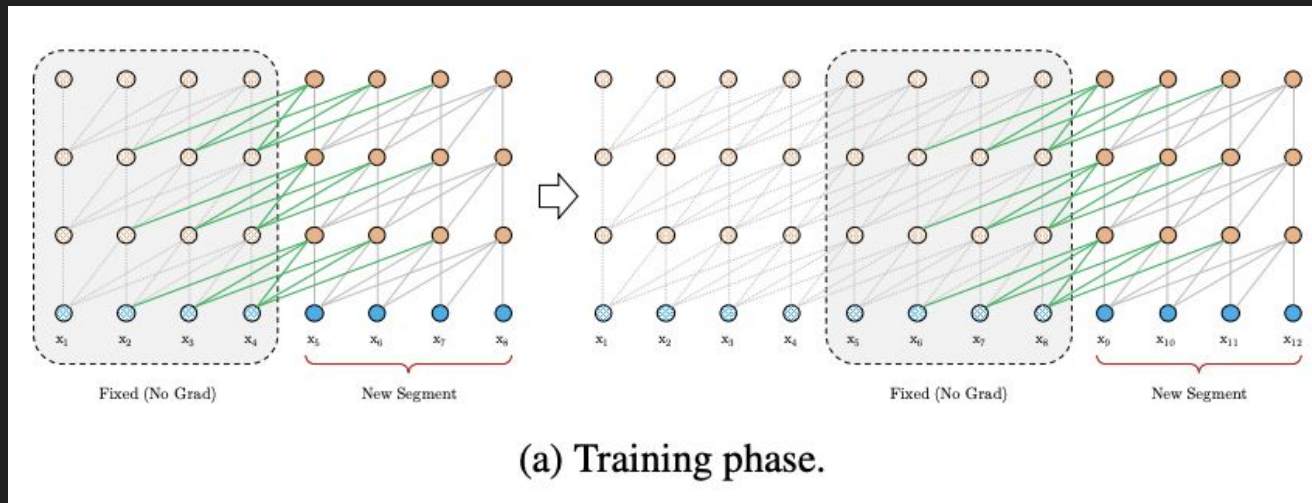
Transformer-XL addresses these by adding **recurrence + relative pos. embeddings**.

Transformer-XL (Recurrence)

Recurrence: computed hidden state sequence cached for next segment

Difference w/ RNN-LMs: recurrent dependency shifts one layer down

Can carry as much memory as GPU memory allows, e.g. tracking last M states gives:
memory := $m_{\{\text{time step } \tau\}^k} \in \mathbb{R}^{M \times d}$ (k is layer #)



Transformer-XL (Positional Embedding)

Secret New Problem: how will we embed the positions of the old hidden states?

Wrong Ans: if we simply add the same pos. vectors to +1-shifted states, the model cannot distinguish states corresponding to τ and $\tau + 1 \rightarrow$ **LOSS**

Right Intuition: want to encode relative pos. info (absolute can be found recursively)

In detail:

1. **Say:** x_i embedded as E_i , added to pos. embedding U_i .
2. **Recall:** $\text{Attention}(x_i, x_j) = (E_i + U_i) W_K W_Q^T (E_j + U_j)^T$

3. **Expanding:**
$$E_i W_K W_Q^T E_j^T + E_i W_K W_Q^T U_j^T + U_i W_K W_Q^T E_j^T + U_i W_K W_Q^T U_j^T.$$

4. **Replacing:**
$$E_i W_K W_Q^T E_j^T + E_i W_K u^T + R_{j-i} W_R W_Q^T E_j^T + R_{j-i} W_R v^T.$$

u^T, v^T, R are learnable!

Transformer-XL (Positional Embedding)

$$E_i W_K W_Q^\top E_j^\top + E_i W_K u^\top + R_{j-i} W_R W_Q^\top E_j^\top + R_{j-i} W_R v^\top$$

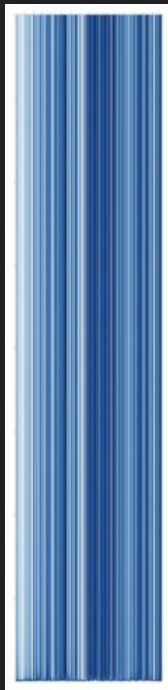
(content) + (global bias for token) + (pos. bias for token) + (global rel. bias)

avg dist. over past
640 tokens

similar

focus on bottom...

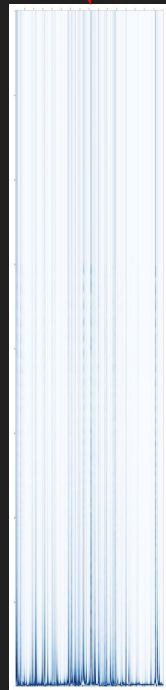
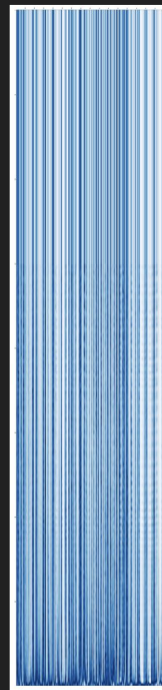
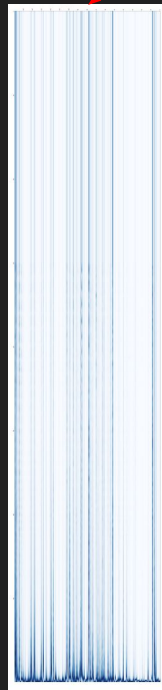
Model has
16 10-head
transformers
+ memory of
length 640



cols = attention heads
(left is #1)

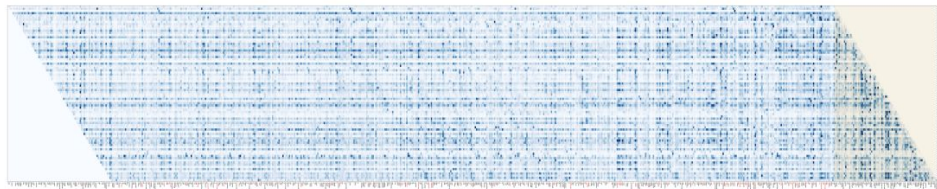
rows = rel. location
(top is #1)

darker = higher value

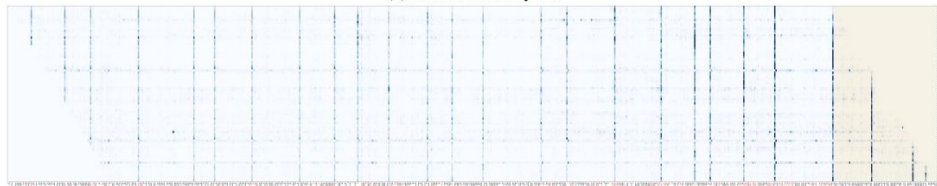


Transformer-XL (Positional Embedding)

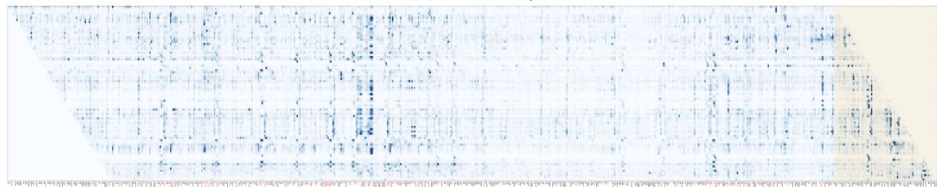
Model has
16 10-head
transformers
+ memory of
length 640



(a) Head 8 from layer 1.



(b) Head 78 from layer 8.



(c) Head 158 from layer 16.

Figure 6: Visualization of the three heads with a wide attention range. Each row corresponds to a target location/token and each column corresponds to a context location/token. Tokens in the memory that have top 20% attention values are highlighted in red.

first layer
almost uniform,
screening for
higher layers

middle layer
focused; rows
share locations

final layer; rows
(targets) have diff.
context locations,
but some loc.
specialized

Transformer-XL (Results)

Model	#Param	bpc
Ha et al. (2016) - LN HyperNetworks	27M	1.34
Chung et al. (2016) - LN HM-LSTM	35M	1.32
Zilly et al. (2016) - RHN	46M	1.27
Mujika et al. (2017) - FS-LSTM-4	47M	1.25
Krause et al. (2016) - Large mLSTM	46M	1.24
Knol (2017) - cmix v13	-	1.23
Al-Rfou et al. (2018) - 12L Transformer	44M	1.11
Ours - 12L Transformer-XL	41M	1.06
Al-Rfou et al. (2018) - 64L Transformer	235M	1.06
Ours - 18L Transformer-XL	88M	1.03
Ours - 24L Transformer-XL	277M	0.99

Table 2: Comparison with state-of-the-art results on enwik8.

Remark	Recurrence	Encoding	Loss	PPL init	PPL best	Attn Len
Transformer-XL (128M)	✓	Ours	Full	27.02	26.77	500
-	✓	Shaw et al. (2018)	Full	27.94	27.94	256
-	✓	Ours	Half	28.69	28.33	460
-	✗	Ours	Full	29.59	29.02	260
-	✗	Ours	Half	30.10	30.10	120
-	✗	Shaw et al. (2018)	Full	29.75	29.75	120
-	✗	Shaw et al. (2018)	Half	30.50	30.50	120
-	✗	Vaswani et al. (2017)	Half	30.97	30.97	120
Transformer (128M) [†]	✗	Al-Rfou et al. (2018)	Half	31.16	31.16	120
Transformer-XL (151M)	✓	Ours	Full	23.43	23.09	640
					23.35	300

Attn Len How much Al-Rfou et al. (2018) is slower

3,800	1,874x
2,800	1,409x
1,800	773x
800	363x

Table 9: Slowdown in terms of running time during evaluation. Evaluation is based on per-token time on one GPU.

Backprop Len	Recurrence	Encoding	Loss	pplx best	pplx init	Attn Len
128	✓	Ours	Full	26.77	27.02	500
128	✓	Ours	Partial	28.33	28.69	460
176	✗	Ours	Full	27.98	28.43	400
172	✗	Ours	Partial	28.83	28.83	120

Table 10: Ablation study on WikiText-103 with the same GPU memory constraints.

Table 10 compares Transformer-XL with baseline under the same memory budget. Transformer-XL still outperforms the baseline even with a shorter backprop length.

Transformer-XL (Results)

Context:

Kershaw started the 2010 season by posting a 3.07 ERA in April, but did so by way against the Milwaukee Brewers at Dodger Stadium, throwing just 57 pitches in 11 1/3 pitcher. He was booed loudly upon being pulled from the game. Kershaw said after feeling to let your teammates down, let everybody down. It stings, it hurts. I've got a two-hitter and out-dueling the then undefeated Ubaldo Jiménez. He credited his corner was suspended for five games after hitting Aaron Rowand of the Giants with a pitch warning following Giants ace Tim Lincecum hitting Matt Kemp earlier in the game against San Francisco and finished the season with a record of 13 – 10 and a 2.91 ERA. == == 2011 season: 1st Cy Young Award == == ==

After finishing the 2010 season strong, the Dodgers named Kershaw as the Opening game shutout of his career, striking out 10 while winning a two-hitter against the F

+

Reference:

, a 3.15 ERA and 593 career strikeouts in 568.2 innings. According to the Elias Sports Bureau, Kershaw was the first 23-year-old pitcher to have that many victories, an ERA that low and an average of more than one strikeout per inning since ERA became an official statistic in 1910. Kershaw was selected to the National League team for the 2011 Major League Baseball All-Star Game, his first All-Star selection. In the month of July, Kershaw was 4 – 1 with a 2.02 ERA and NL-leading 45 strikeouts, earning him the National League Pitcher of the Month Award. On August 23, he struck out Matt Holliday of the St. Louis Cardinals for his 200th strikeout of the season and became the 10th Dodger pitcher to record back-to-back 200 strikeout seasons and the first since Chan-Ho Park did it in the 2001 season. Kershaw finished the 2011 season by leading

XL Generation:

, tying him with Luis Castillo for the third best start total in the league and trailing only Johnny Bench and Eddie Mathews (37). The Dodgers signed Kershaw to a one-year deal on July 14, 2011, with a player option for 2012, in a deal worth just under \$ 300,000. The Dodgers entered the 2011 season as the major league leaders, and Kershaw had the best record in the National League in saves (39), games started (78) and walks (120); he also had the lowest win / loss ratio in the league (1.87), and he had the worst on-base percentage (.349). He had the best ERA (2.00), most wins (16), or most shutouts (29) in his career (25), while leading the league in losses per nine innings pitched (4.01). Following the 2011 season, the Dodgers traded Kershaw to the Kansas City Royals on July 29 for pitcher Javier Torres.

== == Kansas City Royals == ==

== == 2012 season == ==

During spring training, Kershaw played very well. He was selected to spring

⋮

was expected to miss the season as a result of a shoulder injury.

== == 2013 season == ==

On May 17, 2013, Kershaw sustained another back injury and di
August and October 2013. He appeared in 22 starts, all starts f

- Firstly, Kershaw never went to Royals in real life. Despite that, Transformer-XL stays on the fully imagined topic and keeps hallucinating the experience of Kershaw in Royals across the generated text.
- Secondly, notice that XL correctly tracks the chronological order from 2011 to 2012 and to the finally 2013 season in the section titles.
- In addition, notice that Transformer-XL accurately uses the phrase “another back injury” in the 2013 season paragraph, since it has talked about one earlier injury in the 2012 season. This shows again Transformer-XL’s ability of capturing long-term dependency.

COLAB

NEXT WEEK! :)

References

1. Vaswani, Ashish, et al. "Attention is all you need." *Advances in neural information processing systems*, 30 (2017). (<https://arxiv.org/pdf/1706.03762>)
2. Dosovitskiy et. al, *An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale*. International Conference on Learning Representations. (<https://arxiv.org/pdf/2010.11929>)
3. Dai et. al, *Transformer-XL: Attentive Language Models Beyond a Fixed-Length Context*. ACL 2019. (<https://arxiv.org/pdf/1901.02860>)
4. Khandelwal, Urvashi, et al. "Sharp nearby, fuzzy far away: How neural language models use context." *arXiv preprint arXiv:1805.04623* (2018). (<https://arxiv.org/pdf/1805.04623>)
5. Al-Rfou, Rami, et al. "Character-level language modeling with deeper self-attention." *Proceedings of the AAAI conference on artificial intelligence*. Vol. 33. No. 01. 2019. (<https://arxiv.org/pdf/1808.04444>)
6. Feng, T. *Lecture Notes*, Math 270
7. Bishop, C. M., & Bishop, H. (2024). *Deep Learning - Foundations and Concepts*. Springer.