Transformers For Vision

Lecture 5

Outline

- Background on Transformer models
- Transformers for image classification

- [Admin interlude]

- Perceiver models [guest talk from Drew Jaegle, DeepMind]

Transformers for Computer Vision

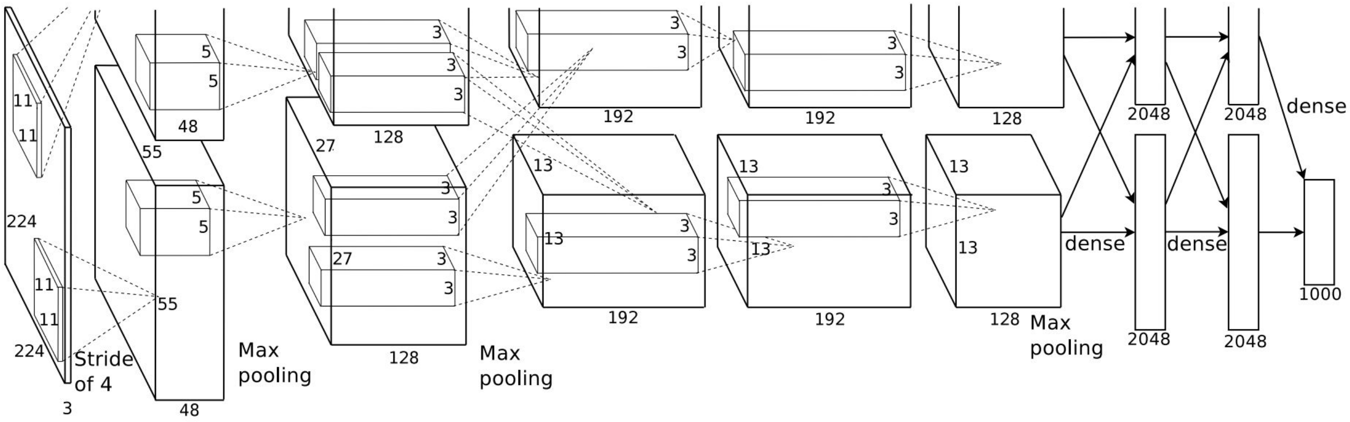
Alexey Dosovitskiy

EEML summer school
July 7th 2021, Budapest (virtually)

Google Research

AlexNet

- AlexNet (2012) - first big success of deep learning in vision*

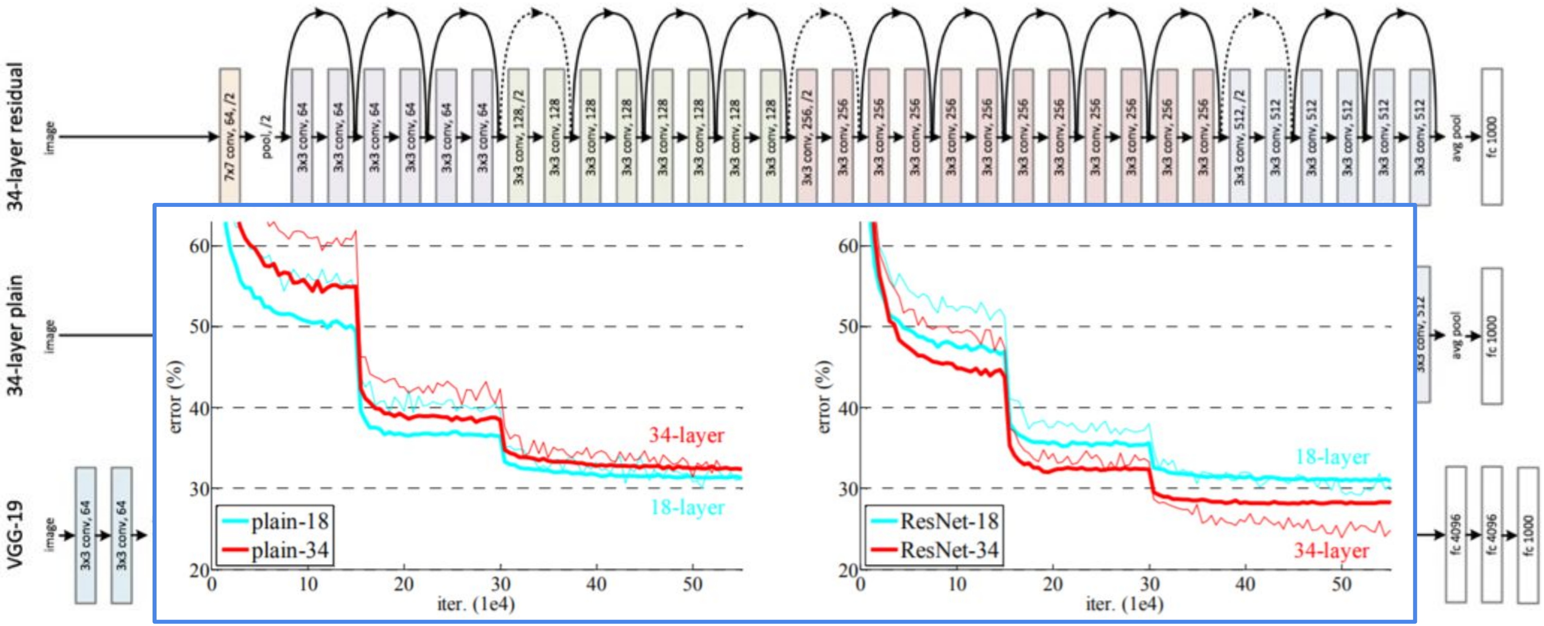


* ConvNets had previously shown good results on specialized dataset like handwritten digits (LeCun et al.) or traffic signs (Ciersan et al.), but not on large and diverse “natural” datasets

[Krizhevsky et al., *ImageNet Classification with Deep Convolutional Neural Networks*, NIPS 2012]

ResNet

- ResNet (2015) - make deep models train well by adding residual connections



[Kaiming He et al., *Deep Residual Learning for Image Recognition*, CVPR 2016]()

Transformer

- Non-vision specific model
  - Typically applied to 1-D sequence data
- Transformer “encoder”
  - A stack of alternating self-attention and MLP blocks
  - Residuals and LayerNorm
- Transformer “decoder” (not shown)
  - A slightly more involved architecture useful when the output space is different from the input space (e.g. translation)

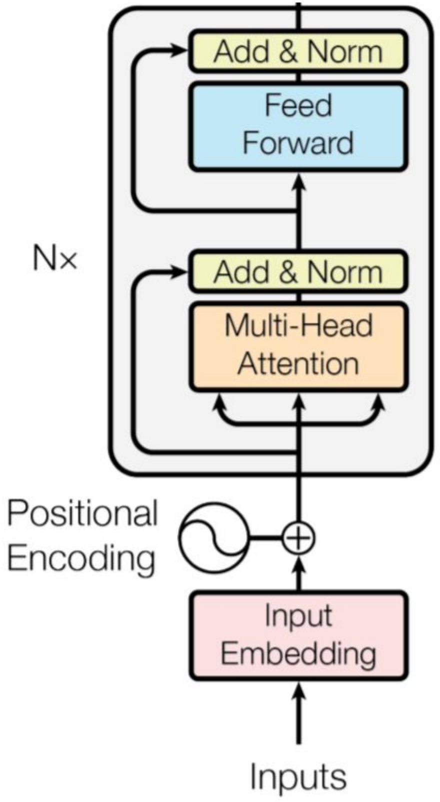
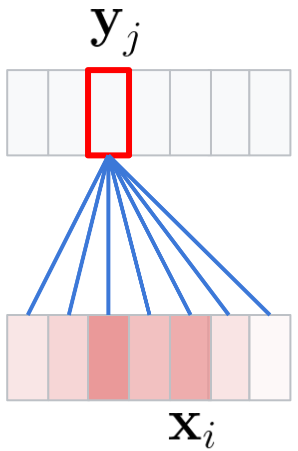


Figure from Vaswani et al.

Vaswani et al., *Attention Is All You Need*, NeurIPS 2017

Self-attention

- Each of the tokens (=vectors) attends to all tokens
  - Extra tricks: learned key, query, and value projections, inverse-sqrt scaling in the softmax, and multi-headed attention (omit for simplicity)
- It's a set operation (permutation-invariant)
  - ...and hence need "position embeddings" to "remember" the spatial structure
- It's a global operation
  - Aggregates information from all tokens



$$\boldsymbol{\alpha}_j = softmax\left(\frac{K\mathbf{x}_1 \cdot Q\mathbf{x}_j}{\sqrt{d_K}}, \ldots, \frac{K\mathbf{x}_n \cdot Q\mathbf{x}_j}{\sqrt{d_K}}\right)$$

$$\mathbf{y}_j = \sum_{i=1}^{n} \alpha_{ji} V\mathbf{x}_i$$

Simplified! Multi-headed attention not shown

Vaswani et al., *Attention Is All You Need*, NeurIPS 2017

Self-Attention with Queries, Keys, Values

Make three version of each input embedding x(i)

- **Query** vector $\mathbf{q}^{(i)} = \mathbf{W}_q \mathbf{x}^{(i)}$
- **Key** vector: $\mathbf{k}^{(i)} = \mathbf{W}_k \mathbf{x}^{(i)}$
- **Value** vector: $\mathbf{v}^{(i)} = \mathbf{W}_v \mathbf{x}^{(i)}$

The **attention weight of the *j*-th position** to compute the **new output for the *i*-th position** depends on the **query of i** and the **key of j (scaled):**

$$w_j^{(i)} = \frac{\exp(\mathbf{q}^{(i)}\mathbf{k}^{(j)})/\sqrt{k}}{\sum_j (\exp(\mathbf{q}^{(i)}\mathbf{k}^{(j)})/\sqrt{k})}$$

The **new output vector for the i-th position** depends on the **attention weights** and **value** vectors of all **input positions j**:

$$\mathbf{y}^{(i)} = \sum_{j=1..T} w_j^{(i)} \mathbf{v}^{(j)}$$

[Julia Hockenmaier, Lecture 9, UIUC]

Transformer self-attention layer

Input: X (matrix of n embedding vectors, each dim m)

Parameters (learned): W_Q, W_K, W_V

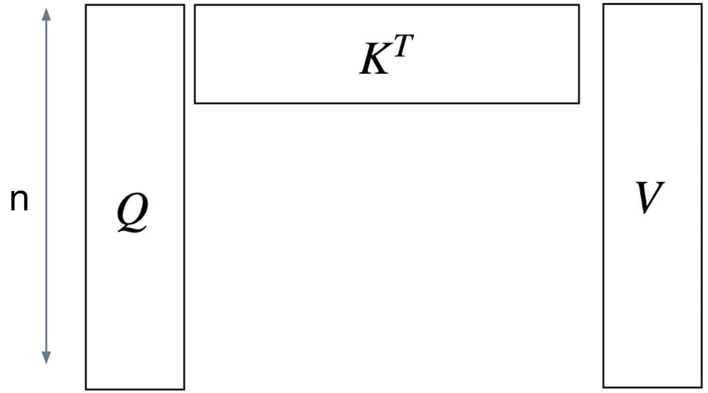
Compute:

$$Q = XW_Q$$

$$K = XW_K$$

$$V = XW_V$$

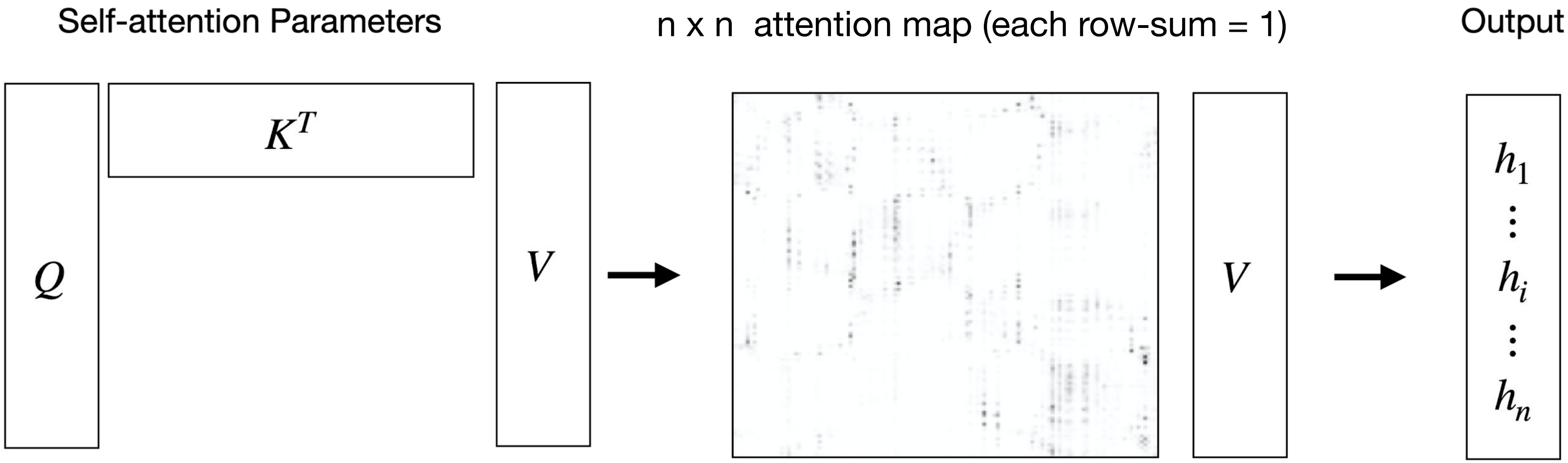
Self-attention Parameters



$$Q, K, V \in \mathbb{R}^{n \times m}$$

$$QK^T \in \mathbb{R}^{n \times n}$$

Transformer self-attention



Output matrix H = $\mathrm{softmax}(\frac{1}{\sqrt{d}} QK^T) \cdot V$

Self-attention explicitly models interactions between all pairs of input embeddings

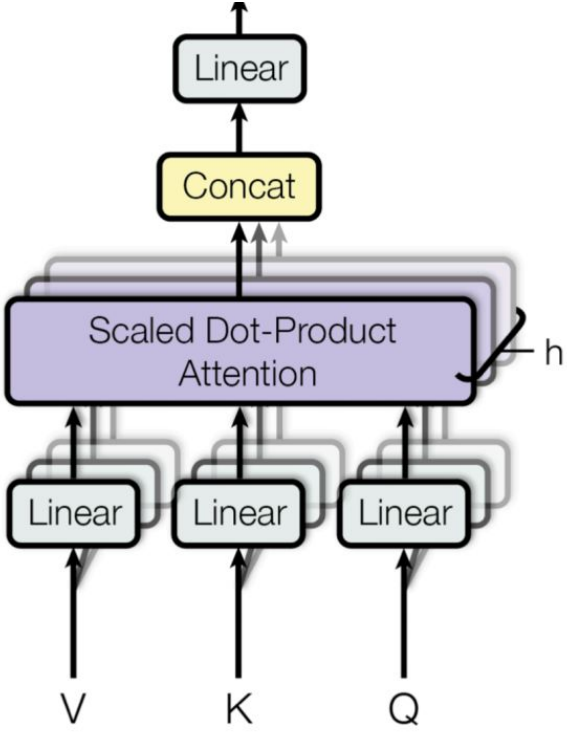
Multi-Head attention

— Learn *h* different linear projections of Q,K,V

— Compute attention separately on each of these *h* versions

— Concatenate and project the resultant vectors to a lower dimensionality.

— Each attention head can use low dimensionality



$$\mathrm{MultiHead}(Q, K, V) = \mathrm{Concat}(\mathrm{head_1}, ..., \mathrm{head_h})W^O$$
$$\text{where } \mathrm{head_i} = \mathrm{Attention}(QW_i^Q, KW_i^K, VW_i^V)$$

[Julia Hockenmaier, Lecture 9, UIUC]

Positional Encoding (1-D)

How to capture sequence order?

Add positional embeddings to input embeddings

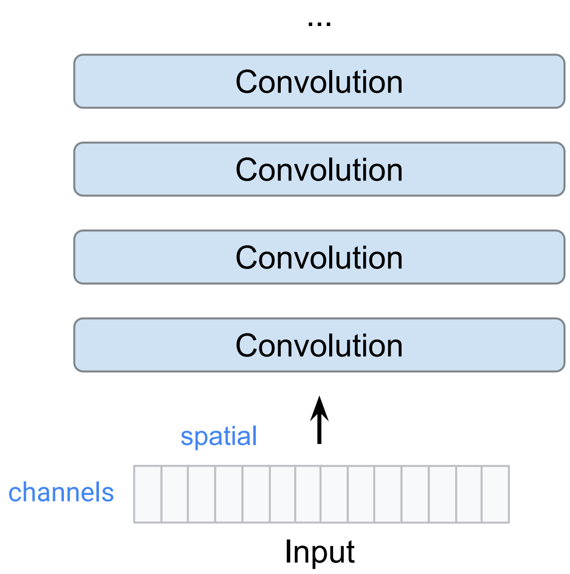
- Same dimension
- Can be learned or fixed

Fixed encoding: sin / cos of different frequencies:

$$PE_{(pos,2i)} = sin(pos/10000^{2i/d_{\text{model}}})$$
$$PE_{(pos,2i+1)} = cos(pos/10000^{2i/d_{\text{model}}})$$

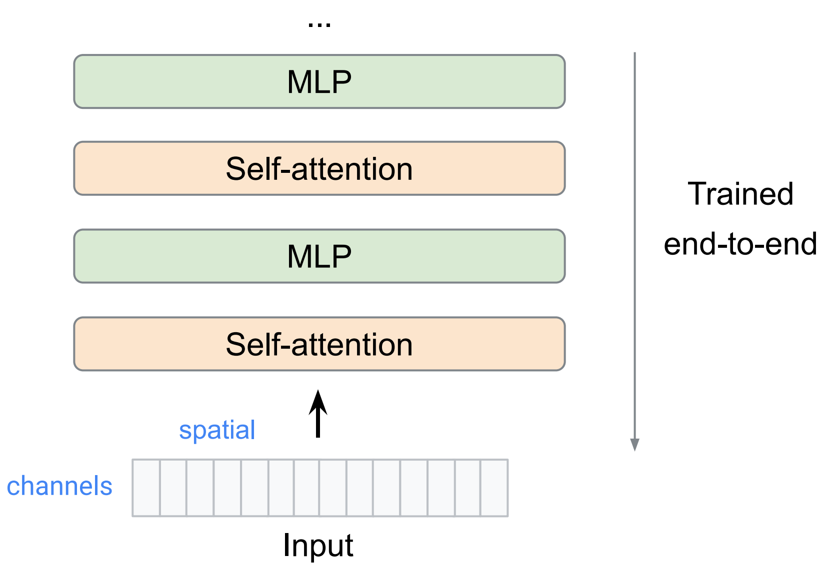
ConvNet vs Transformer

ConvNet



Convolutions (with kernels > 1x1) mix both the channels and the spatial locations

Transformer (encoder)



MLPs (=1x1 convs) only mix the channels, per location

Self-attention mixes the spatial locations (and channels a bit)

*ResNets have grouped of 1x1 convolutions that are nearly identical to transformer's MLPs

BERT model in NLP

- Transformers pre-trained self-supervised perform great on many NLP tasks
    - Masked language modeling (MLM)
    - Next sentence prediction (NSP)

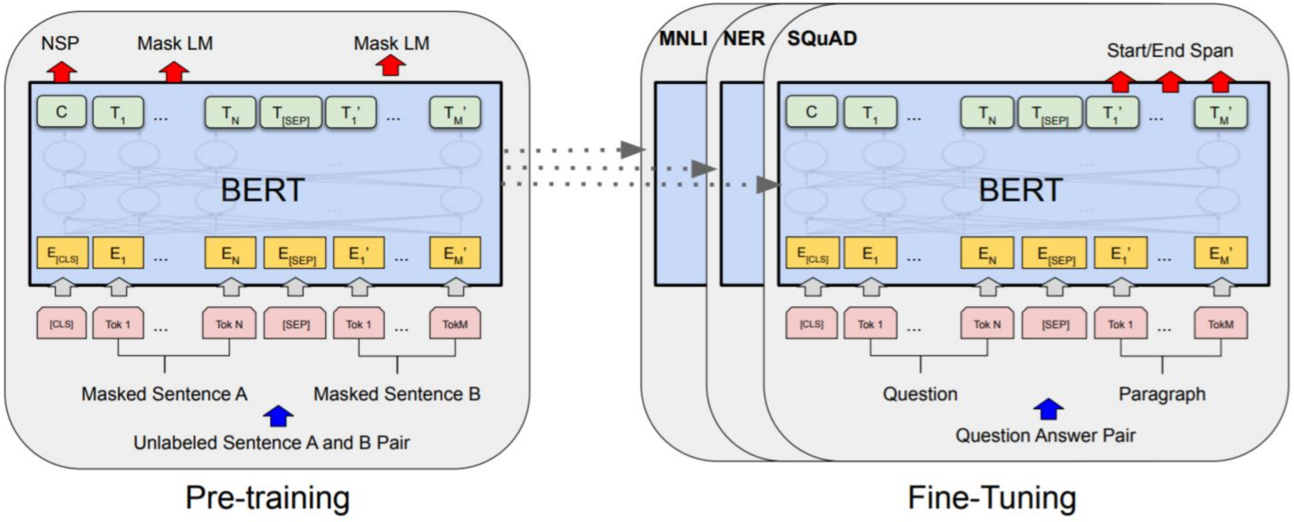
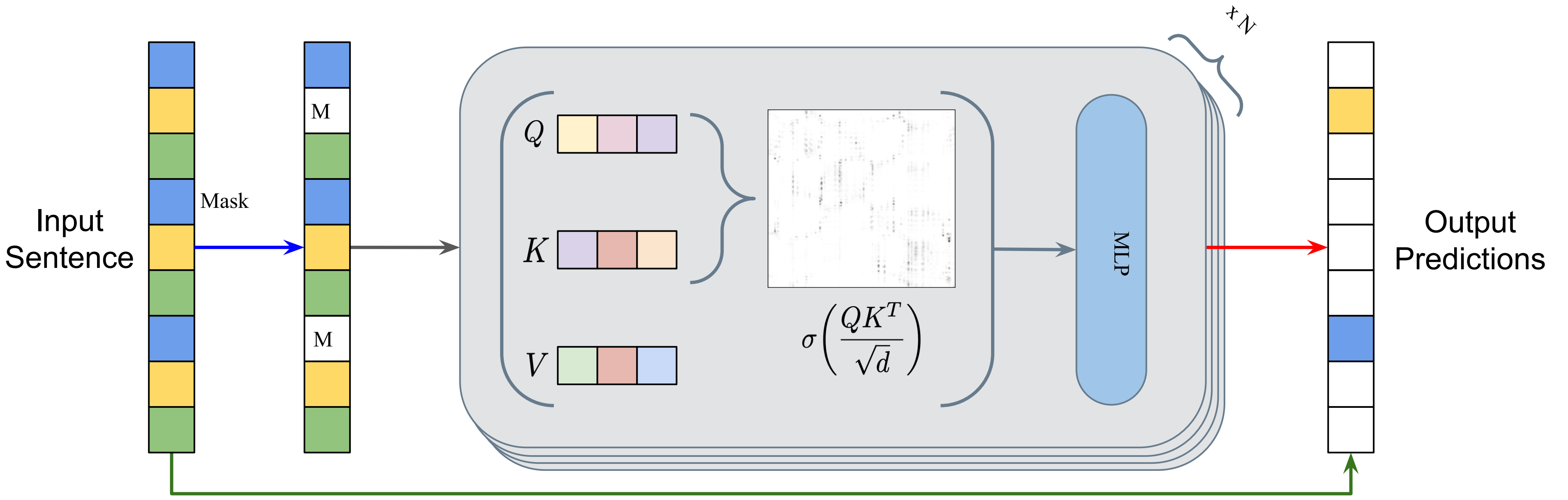


Figure from Devlin et al.

[Devlin et al., *BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding*, arXiv 2018]

Masked language modeling with Transformers (in NLP)



Training: Predict Masked Tokens

$$\mathcal{L}_{\text{MLM}}(X;\theta) = \mathop{\mathbb{E}}_{x\sim X}\,\mathop{\mathbb{E}}_{\text{mask}} \sum_{i\in\text{mask}} \log p(x_i | x_{j\notin\text{mask}};\theta)$$

(mask 15% at a time)

T5, GPT-3

- T5 (Text-to-Text Transfer Transformer)
  - Formulate many NLP tasks as text-to-text
  - Pre-train a large transformer BERT-style and show that it transfers really well

- GPT-3 (Generative Pre-Training)
  - Same basic approach, but generative pre-training and even larger model
  - Zero-shot transfer to many tasks: no need for fine-tuning!

Large-scale self-supervised pre-training “solved”* NLP

*at least made some really impressive progress

Raffel et al., *Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer*, JMLR 2020

Brown et al., *Language Models are Few-Shot Learners*, NeurIPS 2020

Transformers for image classification

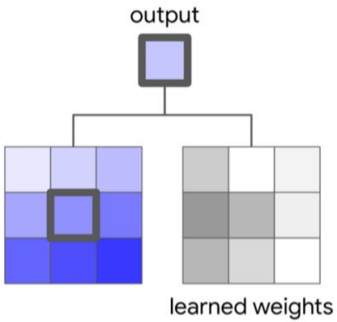
Transformers for vision?

- “LSTM → Transformer” ~ “ConvNet → ??? ”
- Issue with self-attention for vision: computation is quadratic in the input sequence length, quickly gets very expensive (with > few thousand tokens)
  - For ImageNet: 224x224 pixels → ~50,000 sequence length
  - Even worse for higher resolution and video

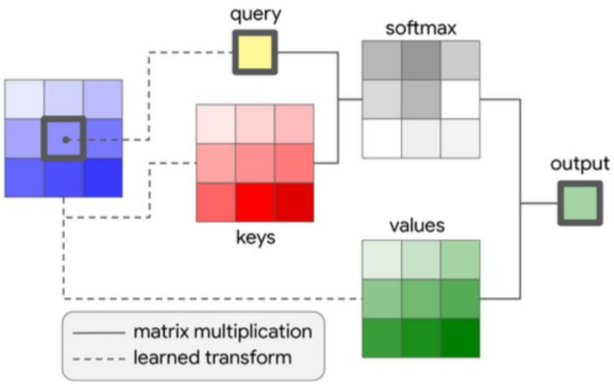
How can we deal with this quadratic complexity?

Local Self-Attention

Convolution



Local self-attention



Idea: Make self-attention local, use it instead of convs in a ResNet

[Hu et al., Local Relation Networks for Image Recognition, ICCV 2019]

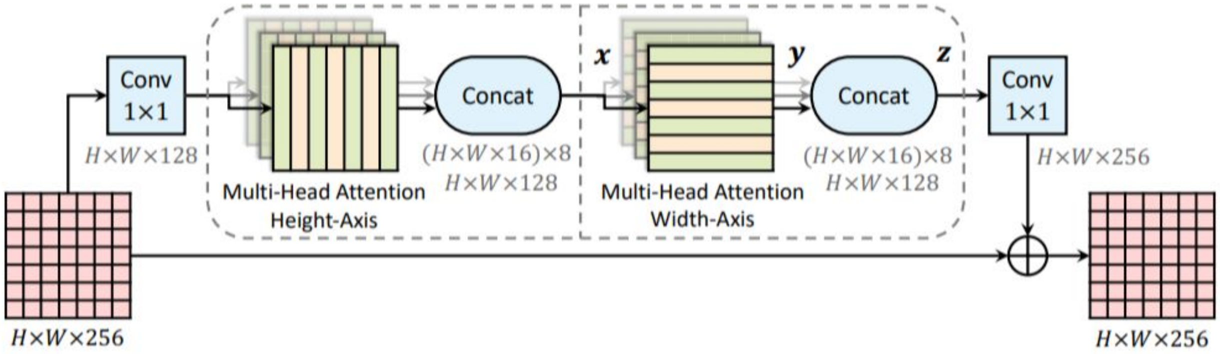
[Ramachandran et al., Stand-Alone Self-Attention in Vision Models, NeurIPS 2019]

[Zhao et al., Exploring Self-attention for Image Recognition, CVPR 2020]

Figures from Ramachandran et al.

Axial Self-Attention

Axial attention block



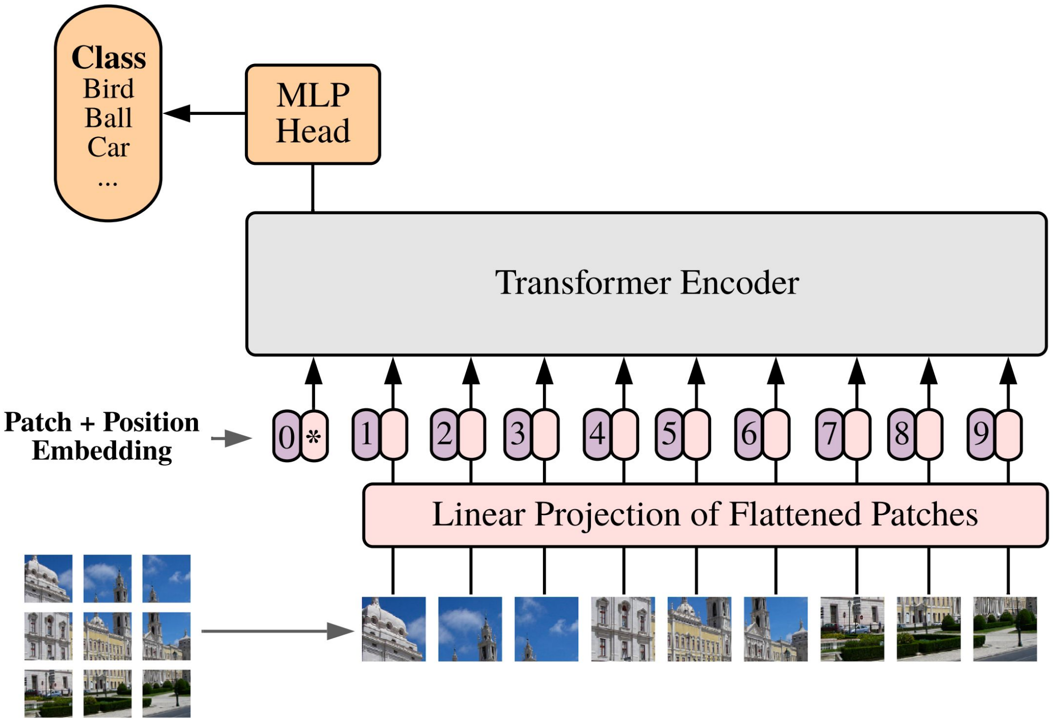
Idea: Make self-attention 1D (a.k.a. axial), use it instead of convs

Figure from Wang et al.

[Wang et al., *Axial-DeepLab: Stand-Alone Axial-Attention for Panoptic Segmentation*, ECCV 2020]

Vision Transformer (ViT)

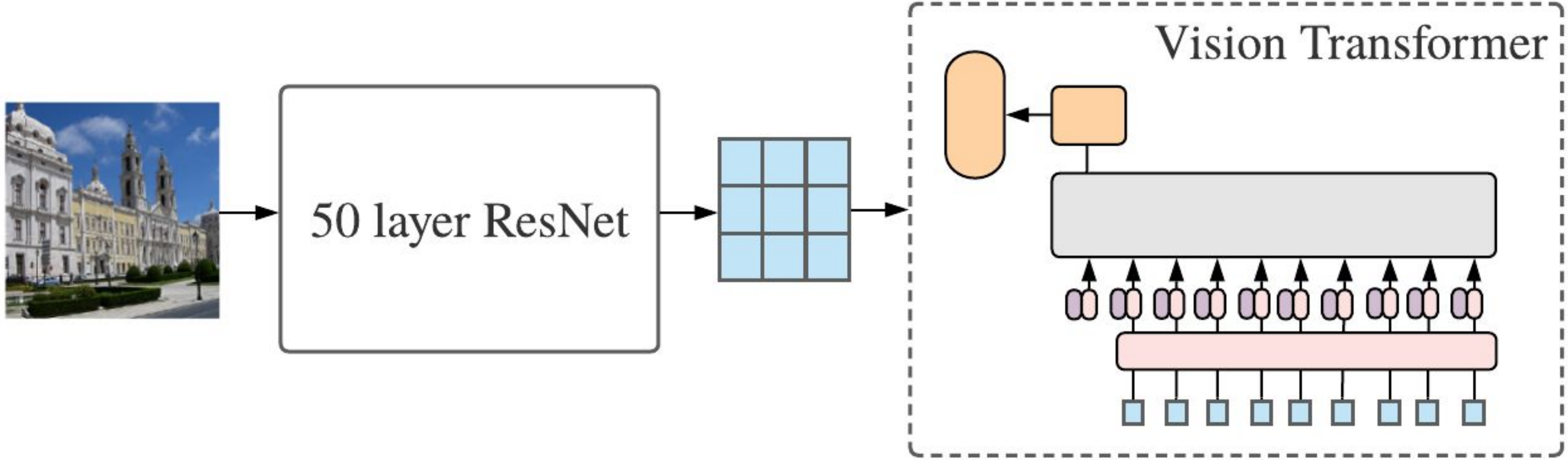
Idea: Take a transformer and apply it directly to image patches



[Cordonnier et al., *On the Relationship between Self-Attention and Convolutional Layers*, ICLR 2020]

[Dosovitskiy et al., *An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale*, ICLR 2021]

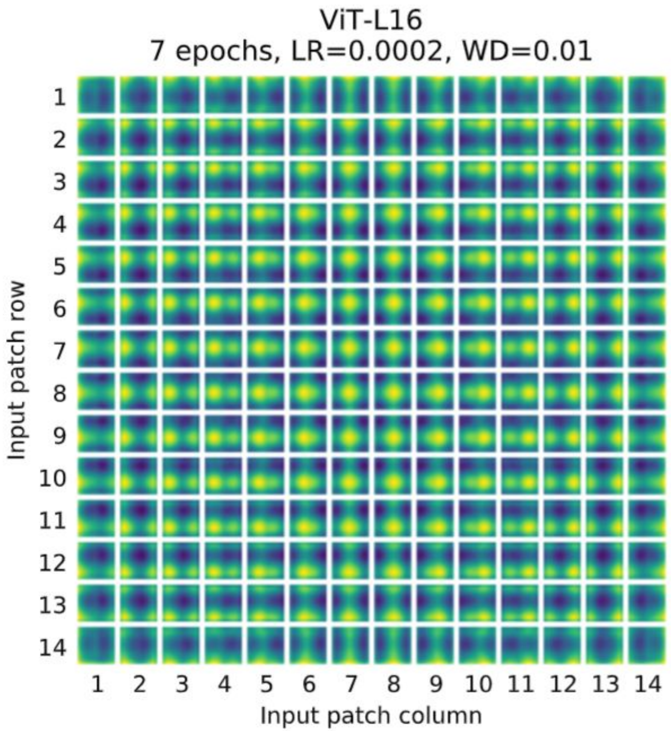
ResNet-ViT Hybrid



Bichen Wu et al. *Visual Transformers: Token-based Image Representation and Processing for Computer Vision*, arXiv 2020

Dosovitskiy et al., *An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale*, ICLR 2021

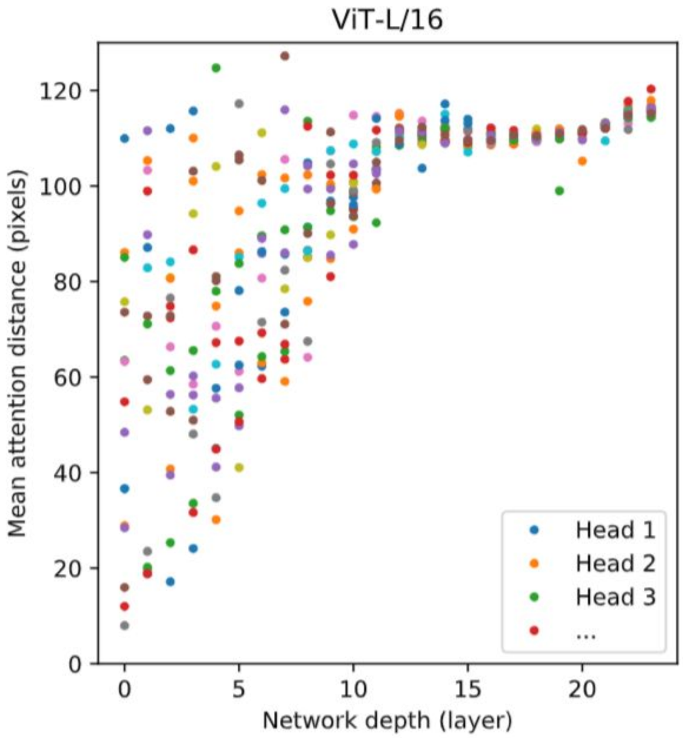
Analysis: Learned Position Embeddings



Conclusion: Learns intuitive local structures, but also deviates from locality in interesting ways

[Dosovitskiy et al., *An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale*, ICLR 2021]

Analysis: “Receptive Field Size”



Conclusion: Initial layers are partially local, deeper layers are global

[Dosovitskiy et al., *An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale*, ICLR 2021]

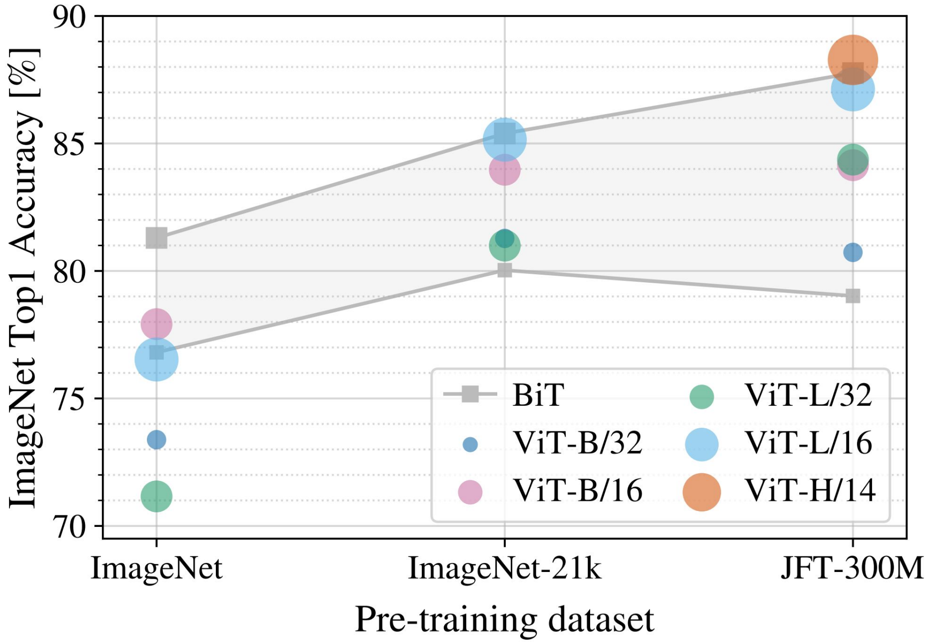
Scaling with Data

Key
ViT = Vision Transformer
BiT = Big Transfer (~ResNet)

ViT overfits on ImageNet, but shines on larger datasets

* with heavy regularization ViT has been shown to also work on ImageNet (Touvron et al.)

** training ViT on ImageNet with the sharpness-aware minimizer (SAM) also works very well (Chen et al.)

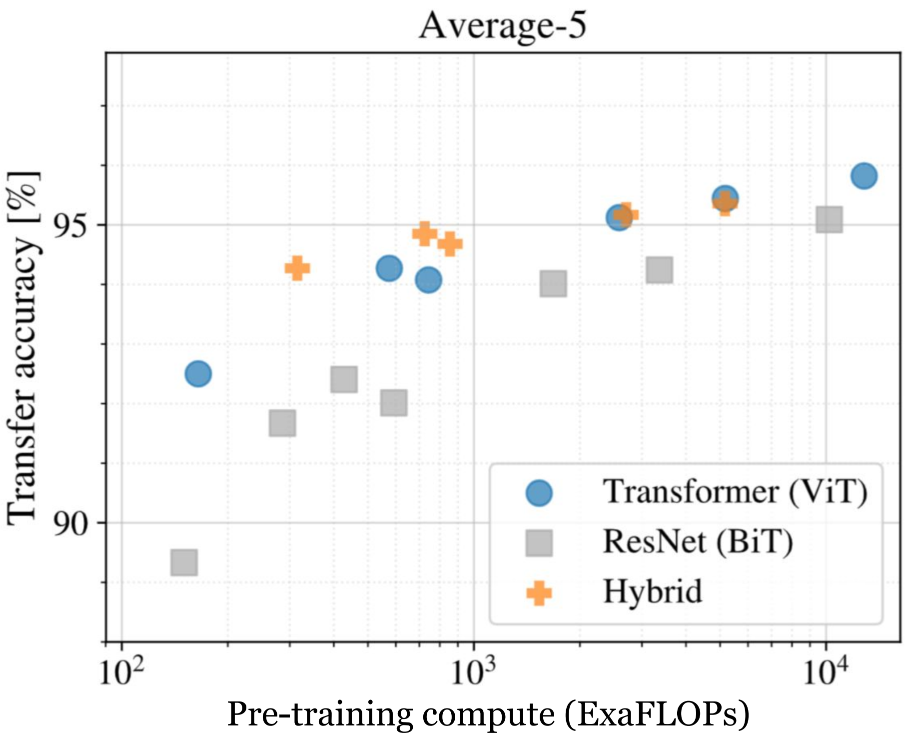


Dosovitskiy et al., *An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale*, ICLR 2021
Xiangning Chen et al., *When Vision Transformers Outperform ResNets without Pretraining or Strong Data Augmentations*, arXiv 2021
Touvron et al., *Training data-efficient image transformers & distillation through attention*, arXiv 2020

Scaling with Compute

Given sufficient data, ViT gives good performance/FLOP

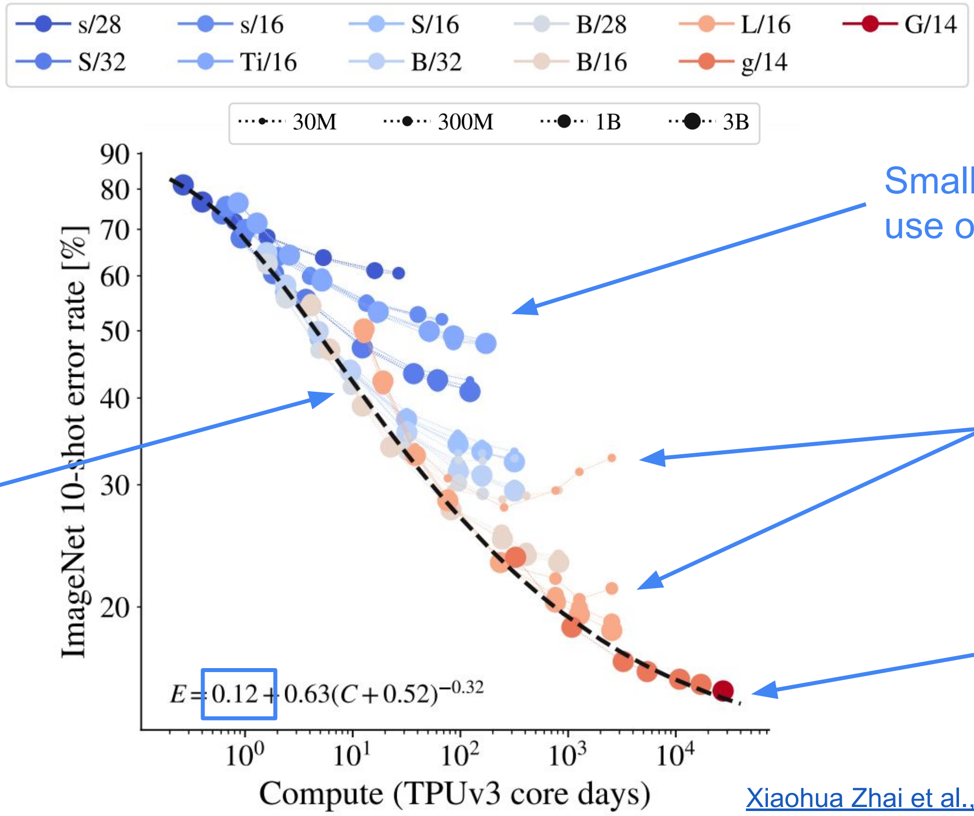
Hybrids yield benefits only for smaller models



Dosovitskiy et al., *An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale*, ICLR 2021

Scaling Laws

How many images do you need for a big model & vice-versa?



Small models can't make use of >30M examples

Even 300M examples are insufficient for large models

Saturates before y=0

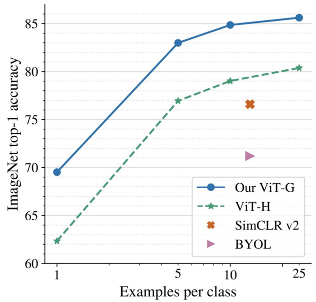
Power-law behaviour $E=aC^{-b}$

[Xiaohua Zhai et al., *Scaling Vision Transformers*, arXiv 2021](#)

Scaling Laws

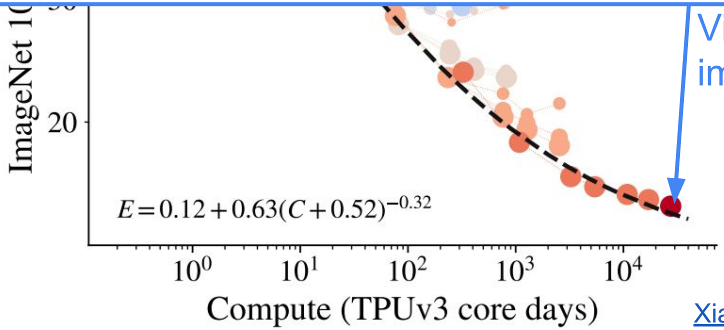
How many images do you need for a big model & vice-versa?

84.86% top-1 accuracy on 10-shot ImageNet (**<1% of train set**)



ImageNet **SOTA** | ImageNet (**OOD**/re-label) variants | 19 **diverse** tasks

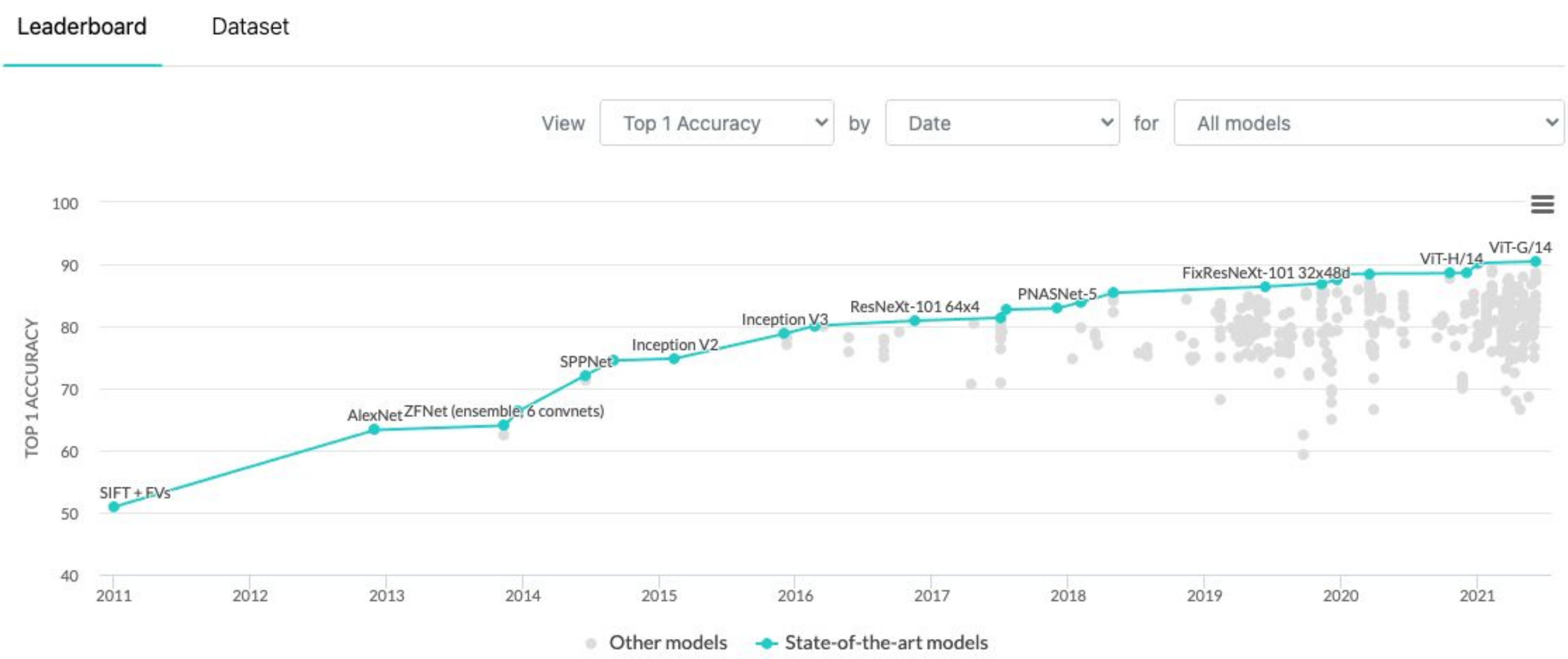
Benchmark	ImageNet	INet V2	INet ReaL	ObjectNet	VTAB (light)
NS (Eff.-L2) [39]	88.3	80.2	-	68.5	-
MPL (Eff.-L2) [24]	90.2	-	**91.02**	-	-
CLIP (ViT-L/14) [26]	85.4	75.9	-	**72.3**	-
ALIGN (Eff.-L2) [16]	88.6	70.1	-	-	-
BiT-L (ResNet) [18]	87.54	-	90.54	58.7	76.29
ViT-H/14 [11]	88.55	-	90.72	-	77.63
Our ViT-G/14	**90.45±0.03**	**83.33±0.03**	90.81±0.01	70.53±0.52	**78.29±0.53**



ViT-G/14: 2B params, 3B images

[Xiaohua Zhai et al., *Scaling Vision Transformers*, arXiv 2021](#)

Image Classification on ImageNet



paperswithcode.com

Summary

Transformer model:

- Alternating layers of self-attention & MLP
- Very few assumptions built into model
- Trained end-to-end
- Easy to scale to be very wide & deep
- Originally applied to NLP (sequences of words)

- Lots of variants in architecture & application

Transformers in vision:

- How to represent image pixels?
    - Too many, given quadratic scaling of model
    - Position in 2D array

- Below SOTA for small models/data (Convnet/Resnets superior)
- SOTA at very large scale (100M-1B images)

Admin Interlude

HPC situation:
- Everyone should now have an HPC account
- Come and see me after if not!

HPC staff have setup GCP account that we can use through Greene login
- Class TAs will hold session to explain this

Projects
- Time to start on projects
- Google doc with some ideas posted in Piazza
    - Will be adding more ideas
- Feel free to come up with your own
- Teams of 2 or 3 people (no teams of 1)
- Every team must chat with me about their proposed idea
    - I will tell you if it is feasible/realistic or not.