STOR566: Introduction to Deep Learning

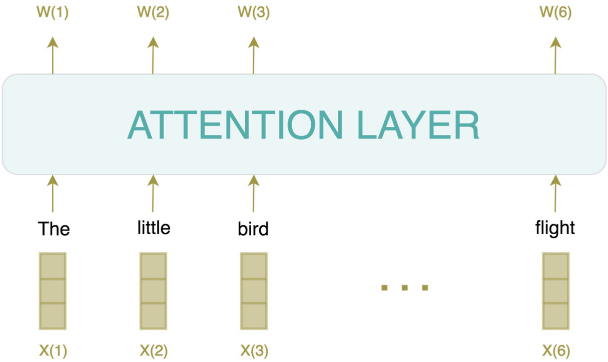
Lecture 18: Transformers for Vision

Yao Li
UNC Chapel Hill

Nov 1, 2022

Materials are from *Deep Learning (UCLA)*

Recap: Transformer for NLP

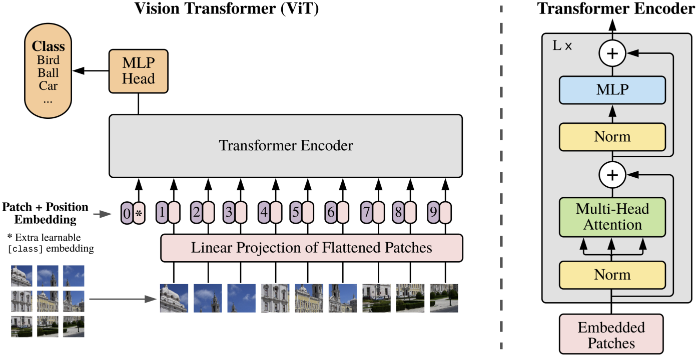


How can we apply it to computer vision?

Vision Transformer (ViT)

Vision Transformer (ViT)

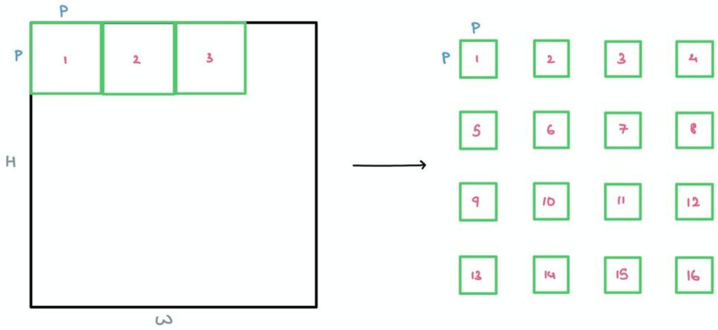
- Partition input image into $K \times K$ patches
- A linear projection to transform each patch to feature (no convolution)
- Pass tokens into Transformer



(Dosovitskiy et al., 2020, "An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale")

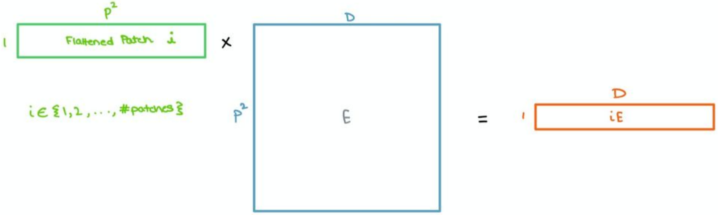
ViT: Image processing

- Partition input image into $K \times K$ patches



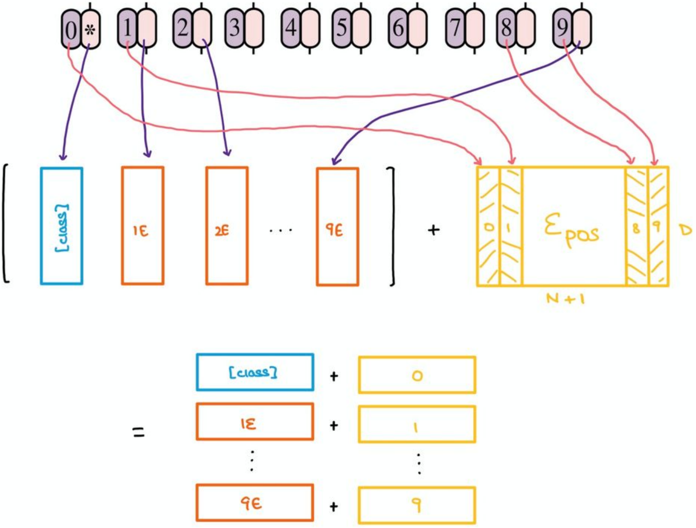
ViT: Projection

- Flatten and projection to feature vector (no convolution)



ViT: Positional encoding

- Add positional encoding

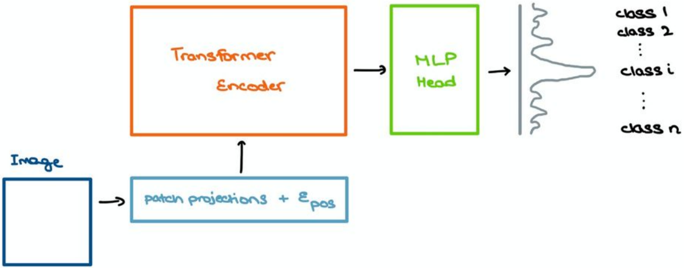


Vision Transformer (ViT)

- Patches are non-overlapping in the original ViT
- $N \times N$ image $\Rightarrow (N/K)^2$ tokens
- Smaller patch size $\Rightarrow$ more input tokens
  - Higher computation (memory) cost, (usually) higher accuracy
- Use 1D (learnable) positional embedding
- Inference with higher resolution:
  - Keep the same patch size, which leads to longer sequence
- Use learnable class embedding

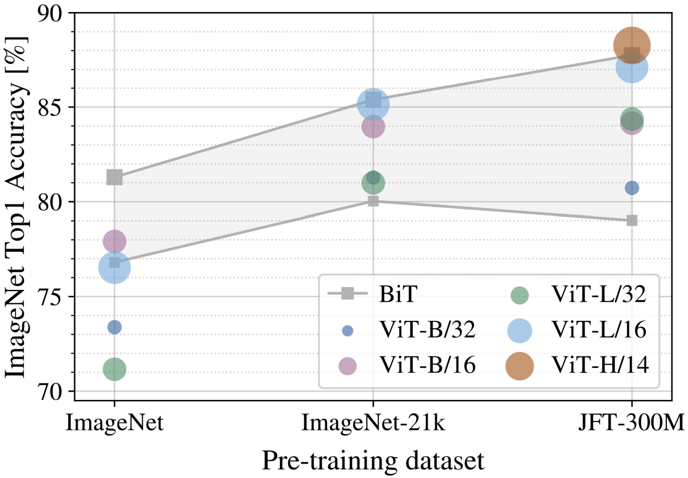
ViT: Class embedding

- Only outputs related to class embedding are fed into the MLP head



ViT Performance

ViT outperforms CNN with large pretraining

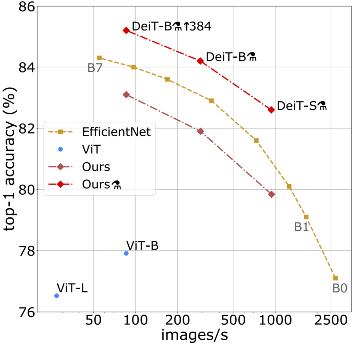


BiT (2020): a SOTA CNN architecture

Attention maps of ViT (to input)

ViT v.s. ResNet

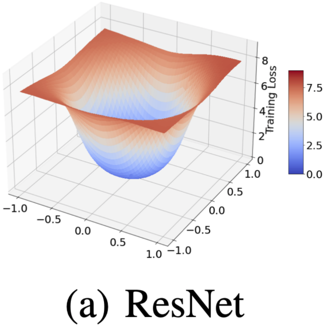
- Can ViT outperform ResNet on ImageNet without pretraining?
- Deit (Touvron et al., 2021):
    - Use very strong data augmentation
    - Use a ResNet teacher and distill to ViT



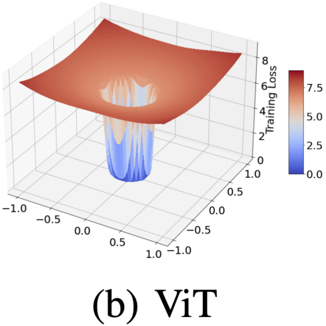
Ablation on ↓	Pre-training	Fine-tuning	Rand-Augment	AutoAug	Mixup	CutMix	Erasing	Stoch. Depth	Repeated Aug.	Dropout	Exp. Moving Avg.	top-1 accuracy: pre-trained 224^2	top-1 accuracy: fine-tuned 384^2
none: DeiT-B	adamw	adamw	✓	✗	✓	✓	✓	✓	✓	✗	✗	81.8 ±0.2	83.1 ±0.1
optimizer	SGD	adamw	✓	✗	✓	✓	✓	✓	✓	✗	✗	74.5	77.3
	adamw	SGD	✓	✗	✓	✓	✓	✓	✓	✗	✗	81.8	83.1
data augmentation	adamw	adamw	✗	✗	✓	✓	✓	✓	✓	✗	✗	79.6	80.4
	adamw	adamw	✗	✓	✓	✓	✓	✓	✓	✗	✗	81.2	81.9
	adamw	adamw	✓	✗	✗	✓	✓	✓	✓	✗	✗	78.7	79.8
	adamw	adamw	✓	✗	✓	✗	✓	✓	✓	✗	✗	80.0	80.6
	adamw	adamw	✓	✗	✗	✗	✓	✓	✓	✗	✗	75.8	76.7
regularization	adamw	adamw	✓	✗	✓	✓	✗	✓	✓	✗	✗	4.3*	0.1
	adamw	adamw	✓	✗	✓	✓	✓	✗	✓	✗	✗	3.4*	0.1
	adamw	adamw	✓	✗	✓	✓	✓	✓	✗	✗	✗	76.5	77.4
	adamw	adamw	✓	✗	✓	✓	✓	✓	✓	✓	✗	81.3	83.1
	adamw	adamw	✓	✗	✓	✓	✓	✓	✓	✗	✓	81.9	83.1

ViT v.s. ResNet

- ViT tends to converge to sharper regions than ResNet



Leading eigenvalue of Hessian: 179.8



Leading eigenvalue of Hessian: 738.8

(Chen et al., 2021, "When vision transformers outperform ResNets without pre-training or strong data augmentations")

"Sharpness" is related to generalization

- Testing can be viewed as a slightly perturbed training distribution
- Sharp minimum $\Rightarrow$ performance degrades significantly from training to testing

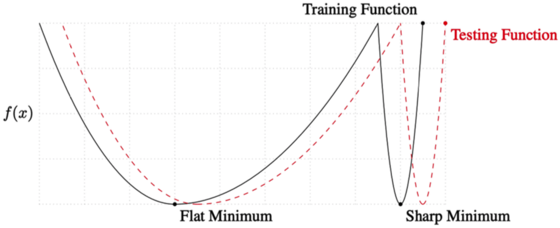


Figure from (Keskar et al., 2017)

Sharpness Aware Minimization (SAM)

- Optimize the worst-case loss within a small neighborhood

$$\min_{w} \max_{\|\delta\|_2 \le \epsilon} L(w + \delta)$$

ϵ is a small constant (hyper-parameter)

- Use 1-step gradient ascent to approximate inner max:

$$\hat{\delta} = \arg \max_{\|\delta\|_2 \le \epsilon} L(w) + \nabla L(w)^T \delta = \epsilon \frac{\nabla L(w)}{\|\nabla L(w)\|}$$

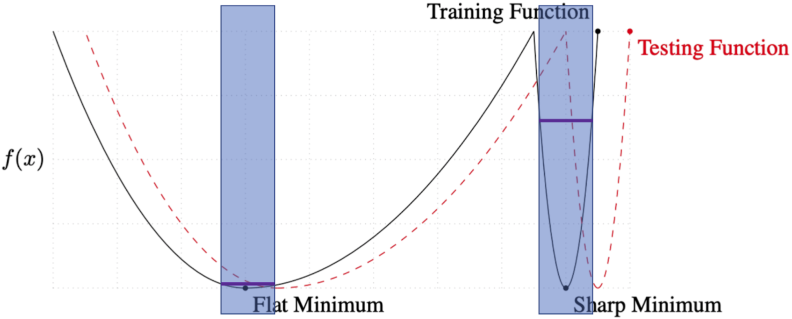
- Conduct the following update for each iteration:

$$w \leftarrow w - \alpha \nabla L(w + \hat{\delta})$$

(Foret et al., 2020, "Sharpness-Aware Minimization for Efficiently Improving Generalization")

Sharpness Aware Minimization (SAM)

SAM is a natural way to penalize sharpness region (but requires some computational overhead)



ViT v.s. ResNet

- When both trained by SAM, ViT outperforms ResNet on ImageNet (without pretraining, strong augmentation, distillation)

Model	#params	Throughput (img/sec/core)	ImageNet	Real	V2	ImageNet-R	ImageNet-C
ResNet							
ResNet-50-SAM	25M	2161	76.7 (+0.7)	83.1 (+0.7)	64.6 (+1.0)	23.3 (+1.1)	46.5 (+1.9)
ResNet-101-SAM	44M	1334	78.6 (+0.8)	84.8 (+0.9)	66.7 (+1.4)	25.9 (+1.5)	51.3 (+2.8)
ResNet-152-SAM	60M	935	79.3 (+0.8)	84.9 (+0.7)	67.3 (+1.0)	25.7 (+0.4)	52.2 (+2.2)
ResNet-50x2-SAM	98M	891	79.6 (+1.5)	85.3 (+1.6)	67.5 (+1.7)	26.0 (+2.9)	50.7 (+3.9)
ResNet-101x2-SAM	173M	519	80.9 (+2.4)	86.4 (+2.4)	69.1 (+2.8)	27.8 (+3.2)	54.0 (+4.7)
ResNet-152x2-SAM	236M	356	81.1 (+1.8)	86.4 (+1.9)	69.6 (+2.3)	28.1 (+2.8)	55.0 (+4.2)
Vision Transformer							
ViT-S/32-SAM	23M	6888	70.5 (+2.1)	77.5 (+2.3)	56.9 (+2.6)	21.4 (+2.4)	46.2 (+2.9)
ViT-S/16-SAM	22M	2043	78.1 (+3.7)	84.1 (+3.7)	65.6 (+3.9)	24.7 (+4.7)	53.0 (+6.5)
ViT-S/14-SAM	22M	1234	78.8 (+4.0)	84.8 (+4.5)	67.2 (+5.2)	24.4 (+4.7)	54.2 (+7.0)
ViT-S/8-SAM	22M	333	81.3 (+5.3)	86.7 (+5.5)	70.4 (+6.2)	25.3 (+6.1)	55.6 (+8.5)
ViT-B/32-SAM	88M	2805	73.6 (+4.1)	80.3 (+5.1)	60.0 (+4.7)	24.0 (+4.1)	50.7 (+6.7)
ViT-B/16-SAM	87M	863	79.9 (+5.3)	85.2 (+5.4)	67.5 (+6.2)	26.4 (+6.3)	56.5 (+9.9)
MLP-Mixer							
Mixer-S/32-SAM	19M	11401	66.7 (+2.8)	73.8 (+3.5)	52.4 (+2.9)	18.6 (+2.7)	39.3 (+4.1)
Mixer-S/16-SAM	18M	4005	72.9 (+4.1)	79.8 (+4.7)	58.9 (+4.1)	20.1 (+4.2)	42.0 (+6.4)
Mixer-S/8-SAM	20M	1498	75.9 (+5.7)	82.5 (+6.3)	62.3 (+6.2)	20.5 (+5.1)	42.4 (+7.8)
Mixer-B/32-SAM	60M	4209	72.4 (+9.9)	79.0 (+10.9)	58.0 (+10.4)	22.8 (+8.2)	46.2 (12.4)
Mixer-B/16-SAM	59M	1390	77.4 (+11.0)	83.5 (+11.4)	63.9 (+13.1)	24.7 (+10.2)	48.8 (+15.0)
Mixer-B/8-SAM	64M	466	79.0 (+10.4)	84.4 (+10.1)	65.5 (+11.6)	23.5 (+9.2)	48.9 (+16.9)

(Chen et al., 2021, "When vision transformers outperform ResNets without pre-training or strong data augmentations")

ViT v.s. ResNet (representation power)

- Let's compare one ViT layer vs one convolution layer
- Reception field: (which input neurons can affect an output neuron)
  - CNN: some subarea of image (kernel size)
  - Self-attention: the whole image
  - $\Rightarrow$ there exists self-attention function that cannot be captured by convolution

Conclusions

- A brief introduction of Vision Transformer.

Questions?