Funnel Vision Transformer
for image classification

yujingz@stanford.edu

Background

Prior work

Transformer

- Dominant in NLP
- Good at capturing distant dependencies
- Parallel self-attention compuation over different token paris
- Easy to scale up

Vision Transformer

- Transformer architecture on the image classification tasks.
- Entirely replace the convolutions with self-attentions
- Outperform the state-of-the-art convolutional networks

Funnel-Transformer

- Efficient Transformer architecture for NLP pre-training in terms of computation FLOPs and memory
- Remove redundant information in sequence dimension

Problem

- Model performance improves with a large model size and more training data, but it's computational more ***expensive***.

Motivation

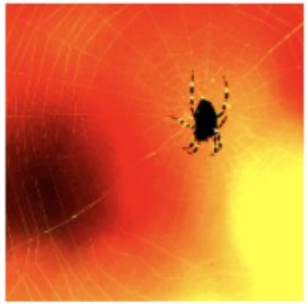
- Build an efficient Vision Transformer by removing ***redundant spatial information*** in deep Transformer layers

  Funnel Vision Transformer

  = Funnel Transformer + Vision Transformer

Problem statement

garden spider, Aranea diademata



- Task: image classification
  - input: an image and a pre-labeled class
  - output: predict a class
  - architecture: Base-Transformer
- Goal: improve the training **efficiency** of Vision Transformer
  - Reach comparable accuracy to Vision Transformer with less computation resource (faster speed and less memory)
  - improve accuracy by re-investing the saved resources to a larger model capacity
- Eval metrics
  - quality: top-1 accuracy
  - resource: memory usage, steps/sec

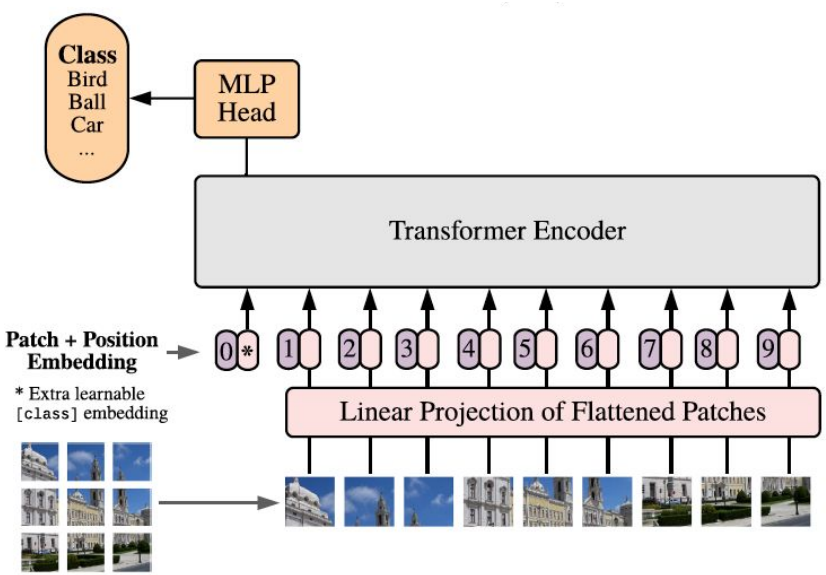
Dataset

ImageNet

- 1000 classes
- 1,281,167 training images, 50,000 validation images and 100,000 test images.
- input resolution: 224 for pre-training, 384 for fine-tuning
- preprocessing: convert to tfrecords to re-use the ViT implementation provided by tensorflow official models.

Methods (baseline)

Vision Transformer (ViT)



Copied from Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. ArXiv, abs/2010.11929, 2021

Image pixels: H x W
Patch size: P x P

$$Z_0 = [X_{class}; X_p^1 E; X_p^2 E; \cdots; X_p^N E] + E_{pos}$$

$$Z_l' = Self-Attention(LayerNorm(z_{l-1})) + z_{l-1}, l = 1 \cdots L$$

$$Z_l = MLP(LayerNorm(z_l)) + z_l', l = 1 \cdots L$$

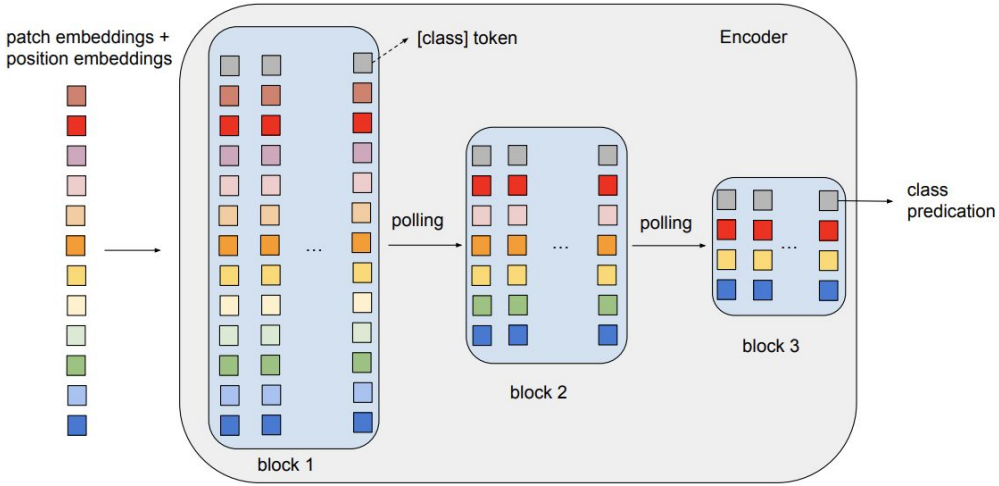
$$y = LayerNorm(Z_L^0)$$

Fine-tuning with different image resolution

- different patch sequence lengths
- apply 2D interpolation of the pre-trained position embeddings

Methods

Funnel Vision Transformer (Funnel-ViT)



- Compress the patch sequence dimension by half through a pooling layer
  - patch dim: N -> (N - 1) / 2 + 1 (no pooling on *class* token)
  - weights shapes are not changed
- "*pool-query-only*" for self-attentions of the first layer after pooling

$$Attn(Q = h', KV = h)$$

- Different re-invest strategies
  - different Transformer configurations across blocks

Experiments and analysis

Pre-training:

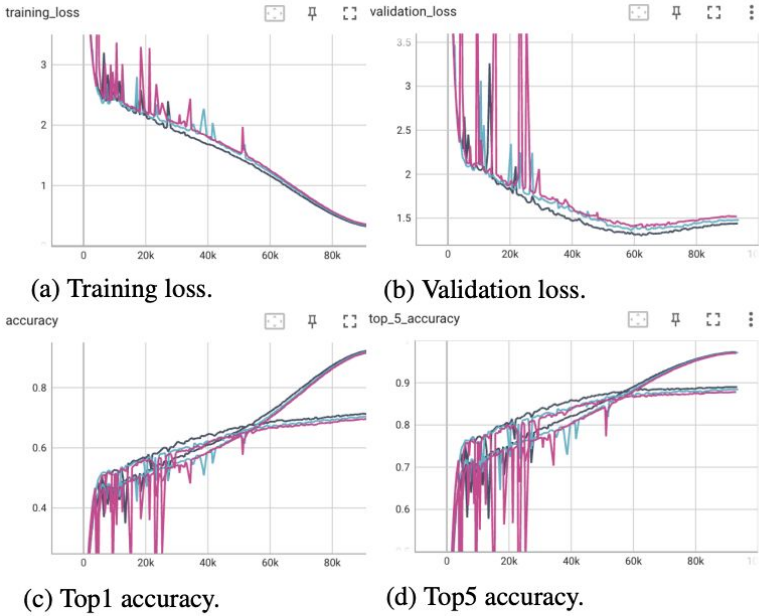
- with a small pre-training accuracy compromise (< 1%)
  - 25.8% less memory, 23% speedup with two funnel blocks
  - 40% less memory, 37.5% speedup with three funnel blocks
- Mean pooling performs better than max pooling

block layout	train top1	val top1	train top5	val top5	memory usage	steps/sec time
B12(197) (ViT)	92.4%	71.41%	97.28%	89%	12.38G	2.0 (12 hours to converge)
B6(197)-6(99), max	92.05%	70.42%	97.15%	88.41%	9.18G	2.6
B6(197)-6(99), mean	92.12%	71.19%	97.17%	88.9%	9.18G	2.6
B4(197)-4(99)-4(50), max	91.66%	69.71%	97.01%	87.8%	7.38G	3.2
B6(197)-6(50), max	92.02%	70.42%	97.14%	88.26%	8.04G	3.1

Table 2. Different block layouts of pre-training on ImageNet.

Bn(t) means that there are n Transformer layers in a block with patch length t.

Experiments and analysis



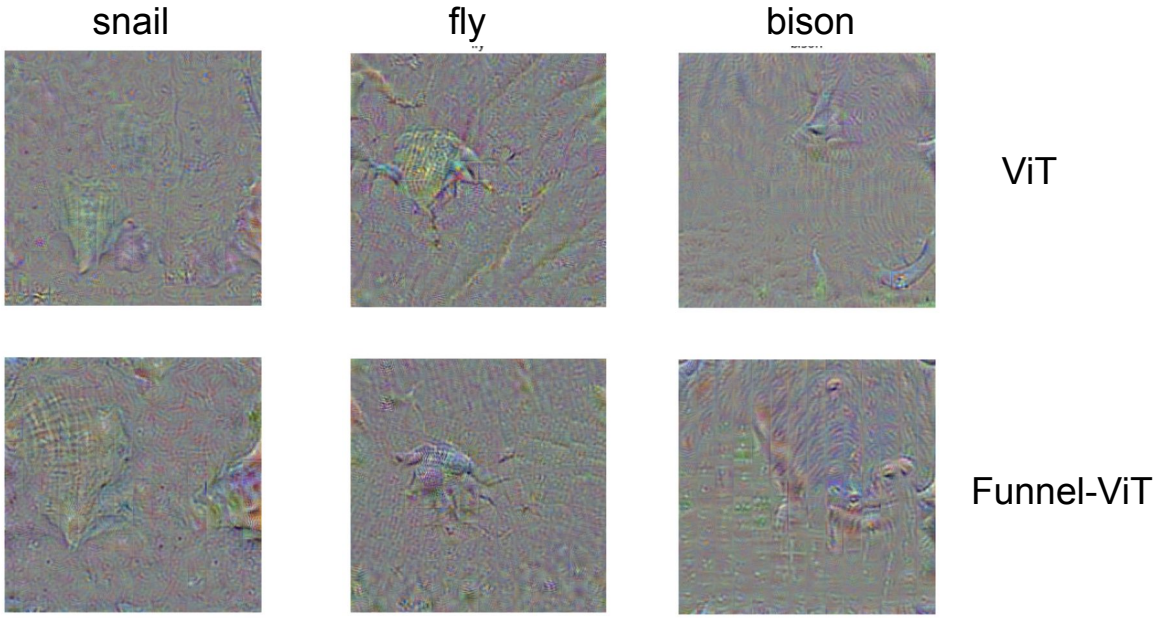
(a) Training loss. (b) Validation loss.

(c) Top1 accuracy. (d) Top5 accuracy.

Figure 4. Loss and accuracy pretraining curves of ViT and Funnel-Vit
(black: ViT; blue: Funnel-ViT B6(197)-6(99); red: Funnel-ViT B4(197)-4(99)-4(50)

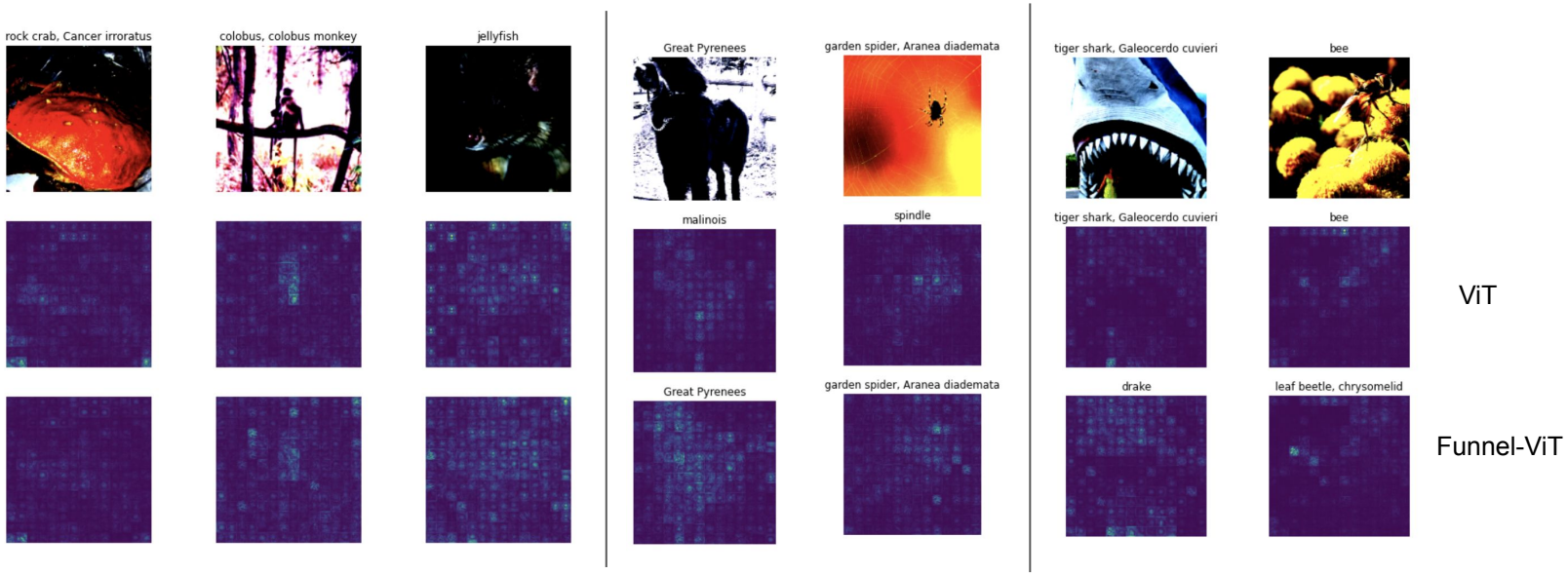
- Converge at the same number of steps.
- If compress the spatial dimensions too much each time (e.g. stride = 4), it would make the training more unstable.

class visualization



Funnel-ViT is able to capture similar class features as ViT.

Saliency maps



Funnel-ViT slightly enlarges regions with a large effect on the classification score. The noise introduced by the additional pixels would disturb the feature extraction.

Results and analysis

Fine-tuning:

- Funnel-ViT outperforms ViT
- Fine-tuning ViT with a pre-trained Funnel-ViT yields better accuracy than the baseline
- Fine-tuning Funnel-ViT with a pre-trained ViT yields worse accuracy than baseline

pre-training	fine-tuning	train top1	val top1	train top5	val top5	memory usage	steps/sec time
ViT	ViT	73.84%	70.95%	90.06%	89.45%	7.32G	4.4
Funnel-ViT	**Funnel-ViT**	75.28%	71.53%	90.8%	89.97%	5.38G	5.8
Funnel-ViT	ViT	75.18%	71.35%	90.74%	89.8%	7.32G	4.4
ViT	Funnel-ViT	73.09%	70.44%	89.7%	89.14%	5.38G	5.8

Table 3. Fine-tuning performance on ImageNet.

Results and analysis

Re-investment:

- Accuracy can be improved slightly with a deeper and wider model
- Overfitting
  - Mitigate overfitting by tuning model width and depth
    - slightly increase model depth
    - increase width of shallow layers (MLP dim); decrease width of deep layers (MLP dim)
  - Avoid overfitting requires more training data (e.g. Imagenet-21k)

block layout	train top1	val top1	train top5	val top5	memory usage	steps/sec time
B12(197) (ViT)	92.4%	71.41%	97.28%	89%	12.38G	2.0
B6(197)-6(99)	92.12%	71.19%	97.17%	88.9%	9.18G	2.6
B6(197)-6(99)-4(50)	93.7%	70.88%	97.76%	88.32%	10.19G	2.3
B6(197, h1024)-6(99, h1024)	93.45%	70.7%	97.69%	88.42%	10.54G	1.76
B6(197, m4096)-6(99, m4096)	93.29%	70.27%	97.62%	88.11%	10.04G	2.35
B6(197)-6(99)-4(50, m1536)	93.04%	71.17%	97.53%	88.79%	9.78G	2.4
B6(197)-6(99)-2(50, m1536)	92.65%	71.13%	97.36%	88.69%	9.44G	2.5
B6(197, m4096)-6(99)-2(50, m1536)	93.05%	71.31%	97.51%	88.69%	10.97G	2.1

Table 4. Different re-investment configurations of pre-training on ImageNet.
(h: hidden dim; m: MLP dim)

Conclusions

- Demonstrate ***spatial information redundancy*** in deeper Transformer layers
- Compression
  - For pre-training, with a small accuracy loss
    - 25.8% less memory, 23% speedup with two funnel blocks (0.22% accuracy loss)
    - 40% less memory, 37.5% speedup with three funnel blocks (0.7% accuracy loss)
  - For fine-tuning, improves accuracy by 0.6%
  - For the whole pre-training and fine-tuning pipeline, ***attain accuracy*** but with much ***less computation resource***
- Re-investment
  - ***Improve accuracy*** slightly by re-investing some saved resources to a deeper and wider model.

Future work

- Try Large-Transformer on a larger pre-training dataset to further explore different re-investment strategies (e.g. ImageNet-21k)
- Experiment with other datasets for fine-tuning to consolidate the conclusion (e.g. CIFAR-10, CIFAR100)