# Discussion Quiz #8

Ashutosh Tiwari (ashutiwa@iu.edu)

March 28, 2022

## 1 Papers

### 1.1 Paper 1

Training Image Transformers Distillation

### 1.2 Paper 2

Swin Transformer

## 2 Questions

### 2.1 As discussed in paper 2, visual elements vary substantially in scale. Explain why the variation in the scale of visual elements makes it difficult to adapt a language-based transformer model to a vision task? (3 pts)

As authors point out, it is difficult to figure out the basic elements of construction in case of images, unlike word tokens. Reason being that the visual elements vary substantially in scale, unlike in case of text which can be divided to word token or alphabet level and scale has more or less no relevant meaning to the problem statement.

### 2.2 What is the relationship between the computational complexity of self-attention in vision-transformer (ViT) model with the image size? What is the solution proposed by swin-transformer to tackle this problem? (2 pts)

In case of ViT, computational complexity of self attention is quadratic to the dimensions of image. This makes using ViT problematic for high resolution images where number of pixels can be very large.

Swin transformers solve this problem by creating hierarchical feature maps. This is achieved by creating these feature maps using non overlapping windows that are partitions of image. Because number of these patches is fixed, computational complexity becomes linear in time.

## 2.3 How does Swin Transformer achieves linear computational complexity for self-attention? Explain in a few sentences. (3 pts)

Swin transformer computes self attention locally on non-overlapping patches which are fixed in number, i.e. are linearly related to size of image. A hierarchical representation is built by merging these feature patches in transformer layers further deep.

Because number of these patches is fixed complexity of these operations is linear in time with respect to size of image. This is also beneficial that the previous architectures in the sense that feature maps are captured not in just one resolution but on a varied number of them.

## 2.4 Discuss the difference in computation complexity of a global multi-head self-attention (MSA) and a window-based MSA on an image of h x w patch size. (2 pts)

As discussed in paper, when a window contains $M \times M$ patches, the computational complexity of a global MSA module is

$$\Omega(MSA) = 4hwC^2 + 2(hw)^2C$$

.

In case of window based module, time complexity is

$$\Omega(W - MSA) = 4hwC^2 + 2M^2hwC$$

.

## 2.5 What is the patch merging layer in Swin transformer architecture? Describe how it operates? (3 pts)

Patch merging layers play a very important role in capturing the hierarchical representation of images. The first patch merging layer concatenates the features of each group of $2 \times 2$ neighboring patches and applies a linear layer on $4C$ dimensional concatenated features. Therefore the input dimension is $2 \times 4C$ while the output dimension to $2C$, reducing the total dimension by a factor of 4.

## 2.6 How DeiT model is different from the original ViT model. List at least two differences. (2 pts)

DieT builds upon the ViT transformer block. Having said that there are considerable differences between the two. Few differences between DeiT and ViT are,

1. First of all images are decomposed into fixed size patches, rather than that of fixed number in case of ViT.
2. Then because these layers are projected onto a linear layer, this becomes invariant to the position of patches in image, which is not the case in ViT.
3. DeiT uses linear classifier for pre training instead of the MLP head used in case of ViT.

### 2.7 In paper 1, many versions of DeiT is trained, explain the difference in architecture between DeiT-B, DeiT-S, and DeiT-Ti in terms of D, h, and d as discussed in paper 1. (3 pts)

There are three different DeiT versions trained by authors. They also list down the differences between all of them. They are tabulated below:

| Model | D | h | d = D/h |
|---|---|---|---|
| DeiT-Ti | 192 | 3 | 64 |
| DeiT-S | 384 | 6 | 64 |
| DeiT-B | 768 | 12 | 64 |

Table 1: Comparison of models

So as we can see, their D and h are different but d remains the same.

### 2.8 Define distillation token as used in paper 1. How does the proposed distillation token is used for training DeiT? Explain in detail. (3 pts)

Distillation token introduced is used in similarity as the class token. This so called distillation token is added to initial embeddings which include patches and class tokens. It then passes through self attention and is taken into consideration by the network learning the representations. The objective of using this token is to be able to reproduce the hard label $y_t$ produced by the teacher network as it facilitates the model to learn from the the teacher's output while remaining complementary to the class token that is tasked to reproduce the true label $y$.

### 2.9 What reasons does the author provide to argue that ConvNets are better teacher models for distillation based training of DeiT model? (3 pts)

Authors point out that convnets are better teachers (than using transformers) because transformers have this inductive bias inherited by the transformers through distillation. After stating this authors also state results of their experiments which corroborate this hypothesis.

### 2.10 ViT needed huge amount of data to reach at par with state-of-the-art ConvNet models, but DeiT does not need that much data. How did the authors tackle this problem? What specific methods were used to achieve this?, name them. (3 pts)

Authors employ a lot of techniques to resolve this data issue. They are listed below:

1. They rely heavily on data augmentation. They try Auto-Augment, Rand-Augment and random erasing to improve their results.
2. Learning from convnets instead of directly learning from data.

## 2.11 Describe briefly. (3 pts)
## (a) Relative position bias (paper 2)
## (b) Soft distillation (paper 1)
## (c) Hard-label distillation (paper 1)

1. **Relative position bias**

   This is a learnt matrix included in the attention formulation. The final formulation is

   $$Attention(Q, K, V) = SoftMax(QK^T/\sqrt{d} + B)V$$

   Authors saw major improvements when this term was included vs when it was not included.

2. **Soft distillation**

   Soft distillation minimizes the Kullback-Leibler divergence between the teacher model's softmax and the student model's softmax. Let $Z_t$ define the logits of the teacher model, $Z_s$ define the logits of the student model. Then the distillation objective is given by

   $$\mathcal{L}_{global} = (1-\lambda)\mathcal{L}_{CE}(\Psi(Z_s), y) + \lambda\tau^2 KL(\Psi(Z_s/\tau), \Psi(Z_t/\tau))$$

3. **Hard-label distillation**

   They introduce this variant of distillation where a hard decision by teacher is taken as true label. It is worth noticing that hard labels can easily be converted into soft labels with label smoothing. Its expression is given below,

   $$\mathcal{L}_{global}^{hardDistill} = \frac{1}{2}\mathcal{L}_{CE}(\Psi(Z_s), y) + \frac{1}{2}\mathcal{L}_{CE}(\Psi(Z_s), y_t)$$

# References