Note:AN IMAGE IS WORTH 16X16 WORDS: TRANSFORMERS FOR IMAGE RECOGNITION AT SCALE

Sinkoo

April 3, 2022

1 Abstract

Transformer's application in limited in CV. A pure transformer applied directly to sequences of image patches can perform very well on image classification tasks.
But if pure Transformer is applied, the accuracy in middle size dataset is discouraging because Transformers lack some of the inductive biases inherent to CNNs.But ViT attains excellent results when pre-trained at sufficient scale and transferred to tasks with fewer datapoints.

2 Method

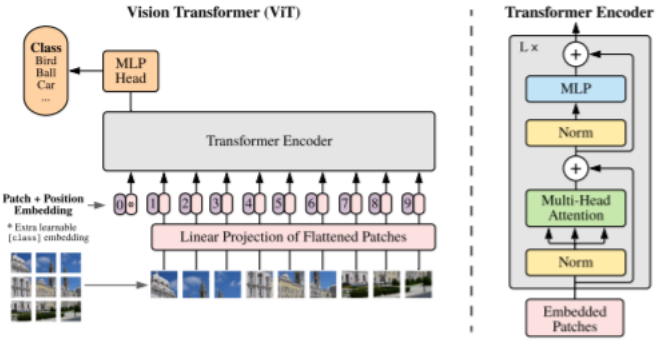


Figure 1: Model overview. We split an image into fixed-size patches, linearly embed each of them, add position embeddings, and feed the resulting sequence of vectors to a standard Transformer encoder. In order to perform classification, we use the standard approach of adding an extra learnable "classification token" to the sequence. The illustration of the Transformer encoder was inspired by Vaswani et al. (2017).

Vision Transformer (ViT) Reshape the image $x \in R^{H*W*C}$ into a sequence of flattened 2D patches $x_p \in R^{N*(P^2*C)}$,(H, W) is the resolution of the original image, C is the number channels, (P, P) is the resolution of each image patch, and $N = HW/P^2$is the resulting number of patches. The Transformer uses constant latent vector size D through all of its layers, so then flatten the patches and map to D dimensions with a trainable linear projection.
Author uses standard learnable 1D position embeddings.
And CNN-ViT can be implemented through using feature map from CNN as input of ViT.

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

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

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

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

Eq1: image 2 sequence; E is that trianable linear projection.
Eq2,3: MSA,MLP;
Eq4:output

Fine-tuning and Higher Resolution Typically, pre-train ViT on large datasets, and fine-tune to (smaller) downstream tasks. Remove the pre-trained prediction head and attach a zero-initialized D $\times$ K feedforward layer, where

K is the number of downstream classes.
And it is often beneficial to fine-tune at higher resolution than pre-training. However, the pre-trained position embeddings may no longer be meaningful. Therefore perform 2D interpolation of the pre-trained position embeddings, according to their location in the original image.

3 Experiments

Data Requirements Author pre-trained ViT in ImageNet-21k, and JFT300M. And optimize three basic regularization parameters – weight decay, dropout, and label smoothing. In ImageNet, ViT-Large models underperform compared to ViT-Base models. With ImageNet-21k pre-training, their performances are similar. With JFT-300M, ViT works better.

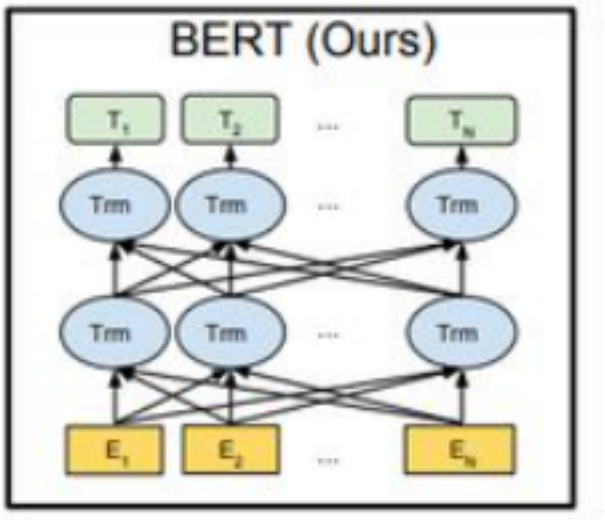
4 Summary

This paper finds directly application of Transformers to Image-based tasks. Divide image into a sequence of patches and process it using Transformer. If dataset is large enough, ViT can reach state-of-the-art or even better. Author gives challenge to implement Transformer in segmentation and detection, and application in segmentation had been solved in [1].

5 Reference

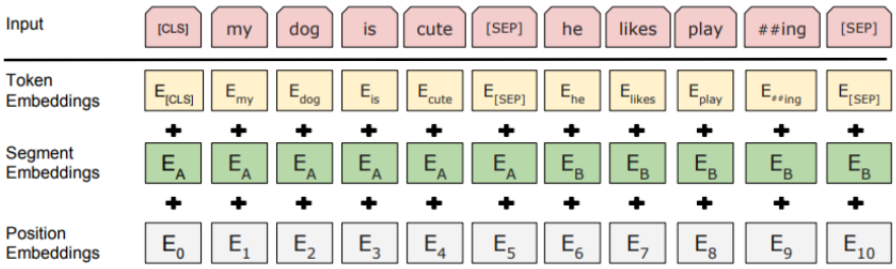
[1],2021"Rethinking Semantic Segmentation from a Sequence-to-Sequence Perspective with Transformers"
[2],LN(Layer normalization) "https://zhuanlan.zhihu.com/p/54530247"

6 Other knowledge: BERT



Bidirectional Encoder Representation from Transformers(BERT) is composed by stacked Transformer encoders.

embedding Token embeddings is a word vector, and the first word is the CLS label(Used in ViT as x_{class} aforementioned), which can be used for subsequent classification tasks.
Segment embeddings is used to distinguish two sentences.
Position embeddings.



Pre-training Task 1: Masked LM In the training process, randomly mask 15% of the tokens instead of predicting every word. The final loss function only cares about tokens that is masked. Not all selected token will be replaced by

[mask], 10% of words will be replaced with other words, 10% of words will not be replaced, and the remaining 80% will be replaced with [mask].