6. Attention-based networks

6.3. Transformers for vision

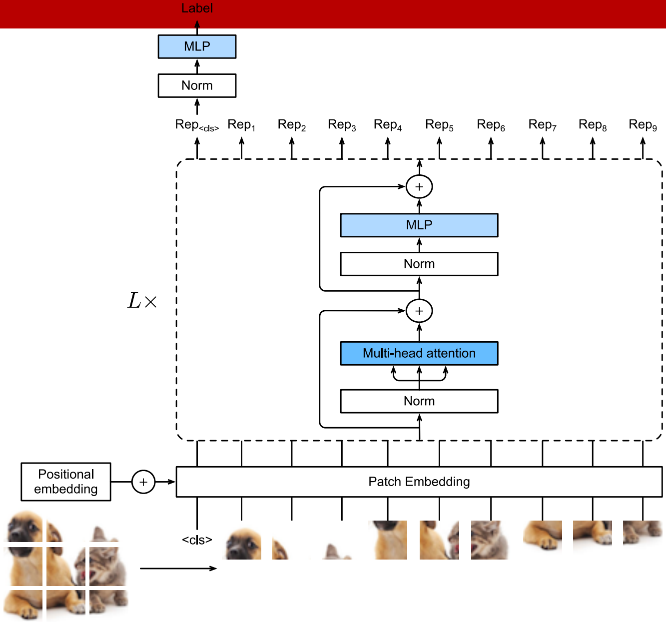
Manel Martínez-Ramón
Meenu Ajith
Aswathy Rajendra Kurup

Introduction

- Transformers were initially developed for natural language processing (NLP), with impressive performance in machine translation and text generation.
- Also recently applied to computer vision tasks as image classification, object detection, semantic segmentation, and others.
- Transformers can potentially overcome some of the limitations of traditional CNNs, such as limited ability to handle long-range dependencies and global context.
- Recent advancements have paved the way for developing models that capture local and global features while handling large variations in object scales and spatial configurations.

Introduction

- Convolution can be replaced with self-attention
- Self-attention can learn to behave similarly to convolution.
- If no constraints in patch size are applied, vision transformers can extract patches from images and use to encode them.
- Transformers show better scalability than convolutions.
- They outperform ResNets.



Label
MLP
Norm
Rep<cls>
Rep1
Rep2
Rep3
Rep4
Rep5
Rep6
Rep7
Rep8
Rep9
+
MLP
Norm
L×
+
Multi-head attention
Norm
Positional embedding
+
Patch Embedding
<cls>

Vision transformer

Operation

- Images as tokens:
    - Input image with height h, width w, and C channels.
    - patch size with dimension p
    - The image is split into a sequence of $m = hw/p^2$ flattened patches with length Cp^2
    - A special <cls> (class) token.
- Sequences are added to learnable positional encodings
- The transformer produces m output vector representations of the same length.
- The <cls> token attends to all the image patches via self-attention. Its representation from the Transformer encoder output is transformed into the output label.

- The transformer consists of alternating multi-head attention and MLP blocks.
- The first layer is a

$$\mathbf{z}_0 = [\mathbf{x}_{class}, \mathbf{E}\mathbf{x}_1, \cdots, \mathbf{E}\mathbf{x}_n] + \mathbf{E}_{pos} \tag{1}$$

  where $\mathbf{x}_n$ is a patch of the image, $\mathbf{E}$ is an embedding matrix and $\mathbf{E}_{pos}$ is a positional embedding
- The expression of the next layers are

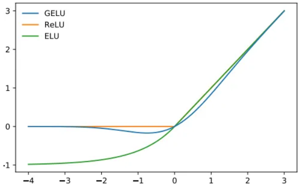
$$\begin{aligned} \mathbf{z}'_l &= \mathrm{MSA}\left(\left(\mathrm{LN}\left(\mathbf{z}_{l-1}\right)\right) + \mathbf{z}_{l-1} \\ \mathbf{z}_l &= \mathrm{MLP}(\mathrm{LN}(\mathbf{z}_l)) + \mathbf{z}'_l \end{aligned} \tag{2}$$

  where MSA stands for multihead self-attention and LN is layer normalization.

- The MLP has two layers and Gaussian Error Linear Activation (GELU)

$$\text{GELU}(u) = x\Phi(u) \approx 0.5x\left(1 + \tanh\left[\sqrt{\frac{2}{\pi}}\left(x + 0.044715x^3\right)\right]\right) \quad (3)$$

where Φ is the error function.



The first element of the sequence at the output is used to code the image class

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

The input sequence can be constructed from feature maps of a CNN.

Vision transformer Experiments

Datasets

- Training
  - ILSVRC-2012 ImageNet dataset with 1000 classes nd 1.3M images.
  - ImageNet-21k with 21000 classes and 14M images
  - JFT with 18000 classes and 303M high-resolution images.
- Test
  - ImageNet with original validation labels
  - ImageNet with ReaL labels
  - CIFAR-10/100
  - Oxford-IIIT Pets
  - Oxford Flowers-102
  - VTAB

Models and results

Model	layers	Hidden size D	MLP size	Heads	Parameters
ViT-Base	12	768	3072	12	86M
ViT-Large	24	1024	4096	12	307M
ViT-Huge	32	1280	5120	16	632M

	(ViT-H/14)	(ViT-L/16)	(ViT-L/16)	(ResNet152x4)	(EfficientNet-L2)
ImageNet	**88.55** ± 0.04	87.76 ± 0.03	85.30 ± 0.02	87.54 ± 0.02	88.4/88.5*
ImageNet ReaL	**90.72** ± 0.05	90.54 ± 0.03	88.62 ± 0.05	90.54	90.55
CIFAR-10	**99.50** ± 0.06	99.42 ± 0.03	99.15 ± 0.03	99.37 ± 0.06	–
CIFAR-100	**94.55** ± 0.04	93.90 ± 0.05	93.25 ± 0.05	93.51 ± 0.08	–
Oxford-IIIT Pets	**97.56** ± 0.03	97.32 ± 0.11	94.67 ± 0.15	96.62 ± 0.23	–
Oxford Flowers-102	99.68 ± 0.02	**99.74** ± 0.00	99.61 ± 0.02	99.63 ± 0.03	–
VTAB (19 tasks)	**77.63** ± 0.23	76.28 ± 0.46	72.72 ± 0.21	76.29 ± 1.70	–
TPUv3-core-days	2.5k	0.68k	0.23k	9.9k	12.3k

14 and 16 stand for the patch size. First L model trained with JFT and second L model trained with Imagenet21K.