# A image is Worth 16x16 words - VIT

## SECTION 1

→ Originally the transformer is applied to NLP (texts processing tasks but it was never been applied to image recognition.]

→ They say that it doesn't do well on mid size data but does and scales really well with a large size data.]

## SECTION 2

* $x \in R^{H \times W \times C}$ but transfor expects a 1D vector of inputs.

* Splits the images into pixels. For example for a $224 \times 224 \times 3$

$P = 16$ means $16 \times 16 \times 3$ is a patch.

* $N = HW/P^2$ = Flatten each patch into a vector. $[x_1, x_2 \ldots x_N]$

* Transformers uses constant vector (D) through all of its layers.

(at the end project back to the old layers)

$$E = x_p W_{proj} \rightarrow (N \times D)$$

↳ $[x_1, x_2 \ldots x_N]$

$$(N \times P^2C) \times (P^2C \times D)$$

↓ $x_p$ ↓ $W_{proj}$

The MLP contains two layers with a GELU non-linearity.

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

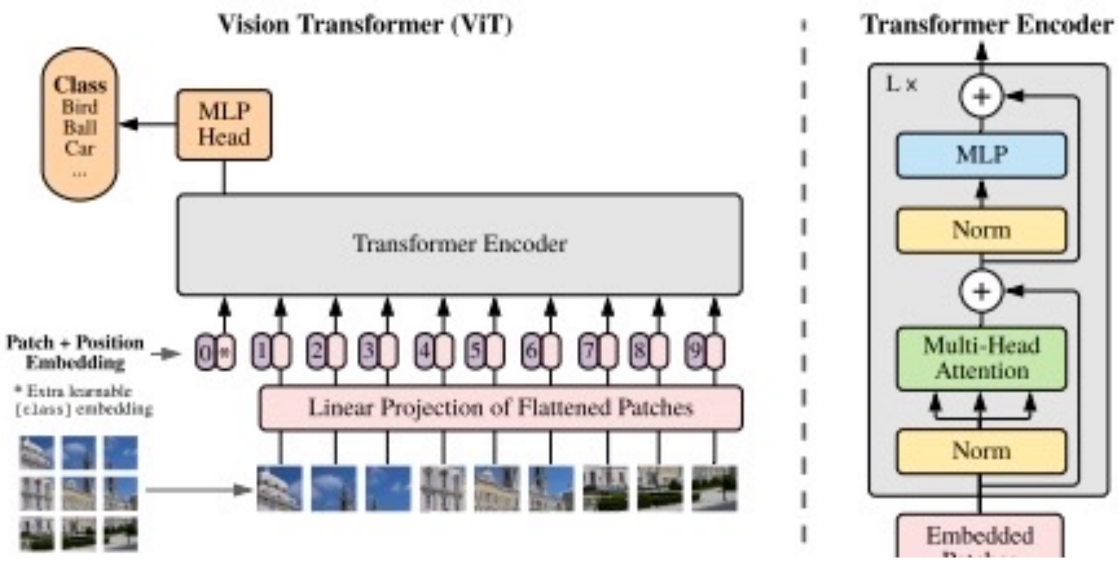
$$\mathbf{z'}_\ell = \text{MSA}(\text{LN}(\mathbf{z}_{\ell-1})) + \mathbf{z}_{\ell-1}, \qquad \ell = 1 \ldots L \tag{2}$$

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

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

**Inductive bias.** We note that Vision Transformer has much less image-specific inductive bias than CNNs. In CNNs, locality, two-dimensional neighborhood structure, and translation equivariance are baked into each layer throughout the whole model. In ViT, only MLP layers are local and translationally equivariant, while the self-attention layers are global. The two-dimensional neighborhood structure is used very sparingly: in the beginning of the model by cutting the image into patches and at fine-tuning time for adjusting the position embeddings for images of different resolution (as described below). Other than that, the position embeddings at initialization time carry no information about the 2D positions of the patches and all spatial relations between the patches have to be learned from scratch.

* To fine tune they replace the classification head with a new head and thus initialize to zeroes with no. of classes.




↳ They add a CLASSTOKEN that pretty much learns to all 'N' patches.

↳ pay attention to

the embedding of transfor

the size of the ...

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

Table 1: Details of Vision Transformer model variants.

EXAMPLE

↳ layers = 12
↳ hidden size = 768
↳ MLP size = 3072
↳ heads = 12

For a 224x224x3 image. With a 16x16x3 patch. $N = (224)^2/(196) = 196$ → tokens

196 + 1 → cls token

→ (197x768) this's the input to the transformer

1. Layer NORM
2. Now MULTI HEAD ATTENTION: (197x768)
   1. $Q = XW_Q$ → (768x768) ↳ (197x768)
   2. $K = XW_K$ → (768x768) ↳ (197x768)
   3. $V = XW_V$ → (768x768) ↳ (197x768)
3. Now we split this into 12 heads

Assume this's done for the batch

↳ 1. Q = (Bx197x768)
2. K = (Bx197x768)
3. V = (Bx197x768)

$\boxed{12 \times 197 \times 64}$

→ split into 12 heads → (Bx12x197x64)

197x64x64 ...

*Now $Q \cdot K^T$ produces our scores so in this case

↳ (197×768)(768×197) → (197×197)

↳ (B×12×197×197) → this the score of how attention (i) should pay to (j)

→ softmax on this → this are the scores attention weights for each token.

* A =


Now = The attention weights are used to weight what each word contributes to feature (i). Thus the tokens just got some embedding. Final step → (B, 12, 197, 64)

↳ (B, 197, 768)

One more output projection → (B, 197, 768) $W_o$ = (B, 197, 768)

* Adds the residual connection: F(x) + x

↓ (197, 768) ↳ (197, 768)

*This outputs $z^{n}o_{l}$ →

Now this is passed to neural net → 768 → 3072

→ (B, 197, 3072) → GELU

→ (B, 197, 768)