## Visual Transformers (ViT)

## by Francesco Pelosin

We dissect, visually, how a Visual Transformer works. We will consider the ViT Tiny architecture, that is, a model composed by 12 layers each of them with 3 heads with embedding size of 192.

The model has been introduced in: https://arxiv.org/pdf/2012.12877.pdf and constitutes the "smallest" ViT architecture available. We consider the input images to be $224 \times 224$ pixels, with 3 channels and patch size of $16 \times 16$.

This is the general overview of a ViT. Each image is splitted in patches and then goes through a Linear Embedder. Here, we add a positional encoding and we append an extra [class] token which is the fundamental vector that is then used in the MLP Head to perform the classification.


## (pre_logits): Identity()



```
(blocks): Sequential(
    (0): Block(
        (norm1): LayerNorm((192,), eps=1e-06, elementwise_affine=True)
        (attn): Attention(
        (qkv): Linear(in_features=192, out_features=576, bias=True)
        (attn_drop): Dropout(p=0.0, inplace=False)
        (proj): Linear(in_features=192, out_features=192, bias=True)
        (proj_drop): Dropout(p=0.0, inplace=False)
    )
    (drop_path): Identity()
    (norm2): LayerNorm((192,), eps=1e-06, elementwise_affine=True)
    (mlp): Mlp(
        (fc1): Linear(in_features=192, out_features=768, bias=True)
        (act): GELU()
        (drop1): Dropout(p=0.0, inplace=False)
        (fc2): Linear(in_features=768, out_features=192, bias=True)
        (drop2): Dropout(p=0.0, inplace=False)
    )
    )
```


## VisionTransformer(

(patch_embed): PatchEmbed(
(proj): Conv2d(3, 192, kernel_size=(16, 16), stride=(16, 16))
(norm): Identity()

## Linear Embedder



Here we unravel the image in patches, each patch is $16 \times 16$, there are in total 14 patches for images 224x224


We add a class token embedding which will be the used at the head to determine the class.
$\oplus^{18}$


Here, a positional matrix is added to the embeddings.

Multi-Head Attention


Head 1
$\mathbf{z}=\operatorname{softmax}\left(\frac{\mathbf{Q K}^{T}}{\sqrt{d_{q}}}\right) \mathbf{V}$

Head 2

$$
\mathbf{z}=\operatorname{softmax}\left(\frac{\mathbf{Q} \mathbf{K}^{T}}{\sqrt{d_{q}}}\right) \mathbf{V}
$$

Head 3
$\mathbf{z}=\operatorname{softmax}\left(\frac{\mathbf{Q} \mathbf{K}^{T}}{\sqrt{d_{q}}}\right) \mathbf{V}$


192 Attention


## Attention Mechanism <br> $$
\mathbf{z}=\operatorname{softmax}\left(\frac{\mathbf{Q K}^{T}}{\sqrt{d_{q}}}\right) \mathbf{V}
$$



Here we report the self
attention mechanism that is performed for each head, in this case we have 3 heads. We report only Head 2


## MLP Head

At the last layer the MLP Head will only consider the cls token


