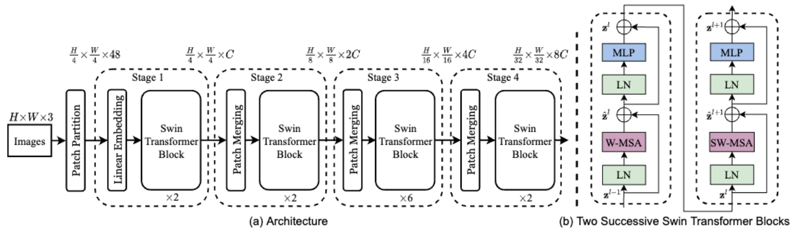
Swin Transformer: Hierarchical Vision Transformer using Shifted Windows

- [[📚 2103.14030v2.pdf]]
- **Basic Idea**:
  - Can we use explicit hierarchical aggregation (progressively reducing the spatial dimension and increasing the channel-wise depth as we move deeper in the network) of information used in CNNs also in Transformers based architectures of vision?
  - Can we reduce the cost of self-attention from quadratic to linear (on the input sequence length/ input image resolution)? Yes, use local self-attention using shifted windows.
  - How to extend usage of ViTs (or transformers) to other computer vision tasks such as object detection and semantic segmentation?
  - Swin-B (88M parameters, 224 × 224 pre-training resolution, 384 × 384 fine-tuning resolution) performance (top1 accuracy) on ImageNet-1k (Table 1 of the paper):
    - *Pre-trained on ImageNet-1k*: 83.5
    - *Pre-trained on ImageNet-21k*: 86.4

- **Need for above properties**
  - *Why hierarchical aggregation?*
    - Objects (visual elements) in an image are of different scales (something object detection community deeply cares about).
    - Whereas, the patches in ViTs are of fixed size (as are words in NLP).
  - *Why linear attention cost?*
    - Dense prediction tasks require pixel level prediction. For such tasks, if we use high resolution images or reduce the patch size, the quadratic self-attention will make the computations intractable.
  - 🟡 **P3** The results of ViT on image classification are encouraging, but its architecture is unsuitable for use as a general-purpose backbone network on dense vision tasks or when the input image 🟡 **P3** resolution is high, due to its low-resolution feature maps and the quadratic increase in complexity with image size.
    - ViTs have a patch-size of 16×16 which results in feature maps of size $\frac{224}{16} \times \frac{224}{16} = 14 \times 14$.
    - Swin Transformers have a downsampling ratio of $32x$, therefore the resulting feature map is of the size $\frac{224}{32} \times \frac{224}{32} = 7 \times 7$.
    - Therefore, it's not that Swin Transformers are producing higher-resolution feature maps. However, Swin Transformers allows us to control the size of the feature map without bloating up the self-attention computation cost.

- **Swin Transformer**
  - 🟡 **P4**



(a) Architecture

(b) Two Successive Swin Transformer Blocks

Figure 3. (a) The architecture of a Swin Transformer (Swin-T); (b) two successive Swin Transformer Blocks (notation presented with Eq. (3)). W-MSA and SW-MSA are multi-head self attention modules with regular and shifted windowing configurations, respectively.

  - *Generating Patches* is similar to ViT.
    Patch size: 4×4. Concatenate RGB channels, therefore the input feature dimension = $4 \times 4 \times 3 = 48$. Project this feature to a arbitrary dimension (C). Number of patches generated: $\frac{H}{4} \times \frac{W}{4}$ where H, W are the height and width of the input image.

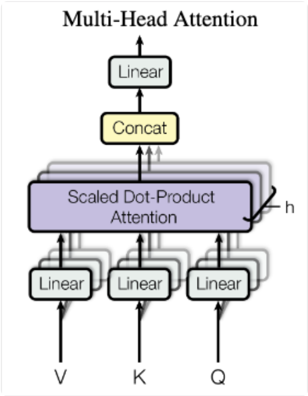
  - *Patch Merging Layer*
    - 🟡 **P3** To produce a hierarchical representation, the number of tokens is reduced by patch merging layers as the network gets deeper.
    - *Merging*: Concatenate the features (each of dimension C) of each group of 2×2 neighbouring patches $\rightarrow$ Features of dimension $4C$ and number of tokens reduced by a factor of 2 along each dimension (therefore, the resulting number of tokens are $\frac{H}{8} \times \frac{W}{8}$)
    - *Dimensionality Reduction*: Apply a linear layer on the concatenated feature (of $4C$-dimension) $\rightarrow$ Features of dimension $2C$.
    - ***Food for thought***
      - Max-pooling operation in a typical CNN is replaced by dimension reducing matrix multiplication (linear transformation).
      - Also, compare it with 1×1 convolutions used in CNNs for dimensionality reduction (reducing the number of channels).

- *Swin Transformer Block*
  - Replace the standard multi-head self-attention (MSA) module with a module based on shifted windows.
  - In the figure, MLP in swin transformer block is a 2-layer MLP with GELU non-linearity in between.

- *Window based Self-Attention*
  - To reduce the quadratic time-complexity to linear, the authors propose to compute self-attention within local windows.
  - **P4** The windows are arranged to evenly partition the image in a non-overlapping manner.
  - Suppose a window contains $M \times M$ patches and there are $h \times w$ patches on the image,
    - Computational complexity of a global MSA:

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

      P4

      This is because for self-attention $Q \in \mathbb{R}^{hw \times C}$, $K \in \mathbb{R}^{hw \times C}$ and $V \in \mathbb{R}^{hw \times C}$. Now the matrix multiplication QK^T has complexity $\mathcal{O}((hw)^2C)$. The multiplication of this term with V has again the complexity $\mathcal{O}((hw)^2C)$. Now, in MSA, incoming input is linearly transformed k times where k is the number of heads. Assume $k = 1$ or consider the computational complexity being calculated for each head individually. Each transformation is done using a matrix multiplication of shape $C \times C$ which increases the computational complexity by hwC^2. This transformation is done for each Q, K, V, thereby amounting to $3hwC^2$. Moreover, once the scalar dot-product attention is computed then the output of all k transformations are concatenated and then again transformed using a matrix multiplication of shape $C \times C$, thereby adding another hwC^2 to the entire computational complexity. Therefore, the total computational complexity is $(hw)^2C + (hw)^2C + 3hwC^2 + hwC^2 = 4hwC^2 + 2(hw)^2C$.

    - Computational complexity of Window-based MSA:

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

      Here $Q \in \mathbb{R}^{M^2 \times C}$, $K \in \mathbb{R}^{M^2 \times C}$ and $V \in \mathbb{R}^{M^2 \times C}$. Therefore, computational complexity of each window is $2(M^2)^2C$. There are $\frac{h}{M} \times \frac{W}{M}$ such windows and hence the total computational complexity of scaled dot-product attention is $2 \cdot M^4 \cdot C \cdot \frac{h}{M} \cdot \frac{W}{M} = 2M^2hwC$. Rest of the computation are similar to global MSA.
    - Note that in the above calculations, the complexity of softmax computation has been ignored.
    - The paper uses $M = 7$.
  - ***Food for thought***
    - This window based self-attention is similar to the restricted self-attention proposed in the original Transformers paper. However, since images have a 2D topology, therefore the windows explicitly exploits the neighbourhood (local) structure while computing attention which the global self-attention usually avoids.
    - Restricted self-attention in original Transformers paper:
      P6 To improve computational performance for tasks involving very long sequences, self-attention could be restricted to considering only a neighborhood of size r in6 the input sequence centered around the respective output position. This would increase the maximum path length to O(n/r).

- *Shifted Window partitioning in successive blocks*
  - **P4** The window-based self-attention module lacks connections across windows, which limits its modeling power. To introduce cross-window connections while maintaining the efficient computation of non-overlapping windows, we propose a shifted window

partitioning approach which alternates between two partitioning configurations in consecutive Swin Transformer blocks.

- 🟡 **P2**

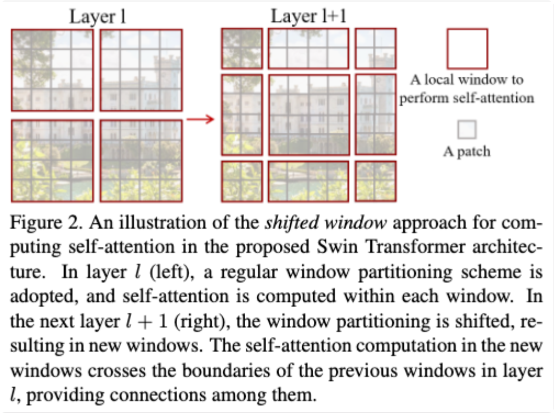
  

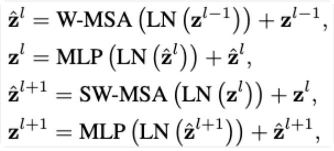
  Figure 2. An illustration of the *shifted window* approach for computing self-attention in the proposed Swin Transformer architecture. In layer l (left), a regular window partitioning scheme is adopted, and self-attention is computed within each window. In the next layer $l + 1$ (right), the window partitioning is shifted, resulting in new windows. The self-attention computation in the new windows crosses the boundaries of the previous windows in layer l, providing connections among them.

- *Regular Partitioning followed by Shifted Windows*

  As shown above, 🟡 **P4** he first module uses a regular window partitioning strategy which starts from the top-left pixel, and the 8 × 8 feature map is evenly partitioned into2 × 2 windows of size 4 × 4 (M = 4). Then, the next module adopts a windowing configuration that is shifted from that of the preceding layer, by displacing the windows by $(\lfloor \frac{M}{2} \rfloor, \lfloor \frac{M}{2} \rfloor)$ 🟡 **P4** pixels from the regularly partitioned windows.

  🟡 **P4** With the shifted window partitioning approach, consecutive Swin Transformer blocks are computed as

  🟡 **P4**

  $$\begin{aligned}\hat{\mathbf{z}}^{l} &= \text{W-MSA}\left(\text{LN}\left(\mathbf{z}^{l-1}\right)\right) + \mathbf{z}^{l-1},\\ \mathbf{z}^{l} &= \text{MLP}\left(\text{LN}\left(\hat{\mathbf{z}}^{l}\right)\right) + \hat{\mathbf{z}}^{l},\\ \hat{\mathbf{z}}^{l+1} &= \text{SW-MSA}\left(\text{LN}\left(\mathbf{z}^{l}\right)\right) + \mathbf{z}^{l},\\ \mathbf{z}^{l+1} &= \text{MLP}\left(\text{LN}\left(\hat{\mathbf{z}}^{l+1}\right)\right) + \hat{\mathbf{z}}^{l+1},\end{aligned}$$

  where $\hat{z}^l$ and z^l denote the output features of the (S)W-MSA module and the MLP module for block l, respectively; W-MSA and SW-MSA denote window based multi-head self-attention using regular and shifted window partitioning configurations, respectively.

- 🟡 **P5** The shifted window partitioning approach introduces connections between neighboring non-overlapping windows in the previous layer.

- 🟡 **P2** This strategy is also efficient in regards to real-world latency: all query patches within a window share the same key set1, which facilitates memory access in hardware. In contrast, earlier sliding window based self-attention approaches [33, 50] suffer from low latency on general hardware due to different key sets for different query pixels2.

  This could be because of the small number of patches in a window, all query and key set can be accommodated in the cache at the same time. However, since global MSA works on all the token which may not entirely fit in the cache which results in a higher access time.

- *Efficient computation for batch computation*

  - 🟡 **P5**

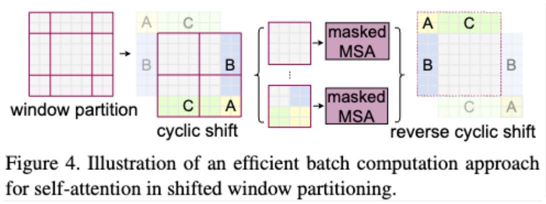
    

    Figure 4. Illustration of an efficient batch computation approach for self-attention in shifted window partitioning.

  - Issue with shifted window partitioning: Results in more windows (from $\lceil \frac{h}{M} \rceil \times \lceil \frac{h}{M} \rceil$ to $(\lceil \frac{h}{M} \rceil + 1) \times (\lceil \frac{h}{M} \rceil + 1)$) and some windows are smaller than $M \times M$. Note that to make the window size (M, M) divisible by the feature map size of (h, w), bottom-right padding is done on the feature map if required.

  - *Naive Solution*: Pad the windows which are smaller than $M \times M$ and mask out the padded values while computing the self-attention. However, 🟡 **P5** When the number of windows in regular partitioning is small, e.g.2 × 2, the increased computation with this naive solution is considerable (2 × 2 → 3 × 3, which is 2.25 times greater = 9/4).

  - *Cyclic Shift solution*: 🟡 **P5** After this shift, a batched window may be composed of several sub-windows that are not adjacent in the feature map, so a masking mechanism is employed to limit self-attention computation to within each sub-window. With the cyclic-shift, the number of batched windows remains the same as that of regular window partitioning, and thus is also efficient.

  - The masking mechanism can be thought of being implemented as

$$softmax\left(\frac{QK^T}{\sqrt{d_k}} * M'\right)V$$

where * represents element-wise multiplication and M' is a mask with values set to $-\infty$ for tokens which do not contribute to the final attention score. Therefore, if we implement the naive solution this MSA will calculated for 3×3 windows whereas if we do a cyclic shift it will be calculated only 2×2.

- *Relative Positional Bias*
  - Add a relative positional bias $B \in \mathbb{R}^{M^2 \times M^2}$ to each head in computing similarity:

    $$\text{Attention}(Q, K, V) = softmax\left(\frac{QK^T}{\sqrt{d}} + B\right)V$$

    where $Q, K, V \in \mathbb{R}^{M^2 \times d}$, d is the query/key dimension and M^2 is the number of patches in a window.
  - Since the relative position along each axis lies in the range $[-(M-1), (M-1)] = [-M+1, M-1]$ (which is the signed distance between any two tokens), the authors parameterise a smaller-size bias matrix $\hat{B} \in \mathbb{R}^{(2M-1)\times(2M-1)}$ and values in B are taken from $\hat{B}$.
  - Note that this is similar to a combination of relative and axial attention mentioned in the ViT paper.
  - Note that an added advantage of the local-window attention is that the number of parameters involved in relative positional embedding get reduced significantly as compared to global self-attention.
- *Fine-tuning*
  - 🟡 **P5** The learnt relative position bias in pre-training can be also used to initialize a model for fine-tuning with a different window size through bi-cubic interpolation [20, 63]

- **Results**
  - 🟡 **P8** While the recent ViT/DeiT models abandon translation invariance in image classification even though it has long been shown to be crucial for visual modeling, we find that inductive bias that encourages certain translation invariance is still preferable for general-purpose visual modeling, particularly for the dense prediction tasks of object detection and semantic segmentation.
  - Since Swin Transformer has stronger inductive biases then the ViT or Mixer models, it is able to perform better even in low data regime (they haven't trained on the gigantic JFT-300M dataset).