# Machine Learning (CE 477)

## Fall 2024

Ali Sharifi-Zarchi

CE Department
Sharif University of Technology

November 29, 2024

# Recap: Self-Attention Mechanism

**Inputs:**

Input vectors: $\mathbf{x}$ (shape: $N \times D$)

**Operations:**

$$\begin{aligned}
\text{Key vectors:} \quad & \mathbf{k} = \mathbf{W}_K^T \mathbf{x} \\
\text{Value vectors:} \quad & \mathbf{v} = \mathbf{W}_V^T \mathbf{x} \\
\text{Query vectors:} \quad & \mathbf{q} = \mathbf{W}_Q^T \mathbf{x} \\
\text{Alignment:} \quad & e_{i,j} = \frac{\mathbf{q}_j \cdot \mathbf{k}_i}{\sqrt{D}} \\
\text{Attention:} \quad & a = \text{softmax}(e) \\
\text{Output:} \quad & \mathbf{y}_j = \sum_i a_{i,j} \mathbf{v}_i
\end{aligned}$$

**Outputs:**

- Self-attention-based architectures, in particular Transformers, have become the model of choice in natural language processing (NLP).
- The dominant approach is to pre-train on a large text corpus and then fine-tune on a smaller task-specific dataset.
- In computer vision, however, convolutional architectures remain dominant.

## Strengths of CNNs

- *Local Feature Extraction:* CNNs excel at extracting local patterns through convolutions, detecting edges, textures, and shapes in a hierarchical manner.
- *Translation Invariance:* The use of convolution and pooling layers helps CNNs detect features regardless of their position in the image.
- *Efficient Computation:* Weight sharing in convolutional layers makes CNNs computationally efficient compared to fully connected networks for images.

# Limitations of CNNs

- *Limited Receptive Field:* Early convolutional layers only see a small portion of the image at a time (local features), while later layers expand the receptive field, but still require deep architectures to model long-range dependencies.
- *Pooling Information Loss:* Pooling layers (e.g., max-pooling) help reduce computational costs but can lose important spatial details, especially for global image understanding.
- *Struggle with Global Context:* CNNs focus more on local spatial hierarchies and have difficulty capturing global relationships between distant parts of an image.

## CNNs Struggle with Global Context

- Why it matters: For tasks that require understanding the entire image (e.g., image classification, where the relationship between distant parts of the image may be important), CNNs often require deep architectures to get the full picture.
- Feature Hierarchy: CNNs build a hierarchical structure of features but rely on many layers to achieve global understanding, leading to more complexity.

# Why Look Beyond CNNs?

- From Local to Global
- NLP to Vision
- Global Dependency

## What Makes Transformers Powerful?

- Self-Attention Mechanism: Explain the core of the Transformer architecture: the self-attention mechanism, which allows each token (or patch in ViT) to weigh its relationship with every other token. This helps the model learn global context.
- Global Context from the Start: Unlike CNNs, which build local-to-global context layer by layer, Transformers have global context awareness from the very first layer.
- Parallelization: Unlike sequential processing in CNNs, Transformers allow parallel processing of image patches, speeding up training on large datasets.

## Transformers: From NLP to Vision

- No Need for Hand-Crafted Convolutions: Explain that Transformers don't rely on convolutions or pooling, which are manually crafted for images. Instead, they apply the same attention mechanism used in NLP.
- Adaptability to Different Domains: Discuss how the Transformer's architecture is flexible and adaptable to multiple domains (e.g., vision, audio, NLP) without the need for task-specific layers.

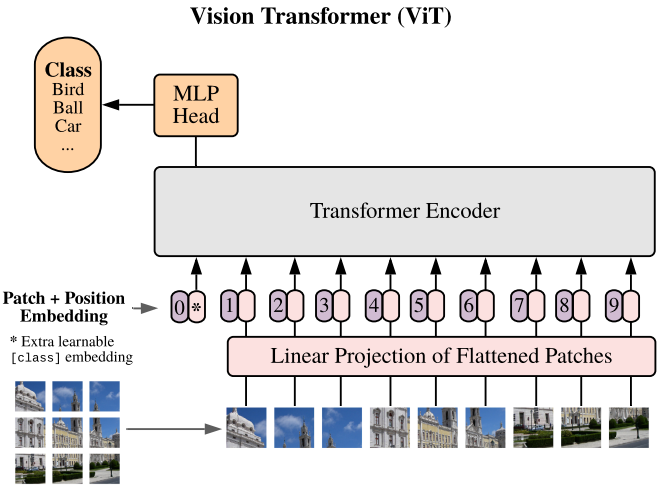
## Processing Text

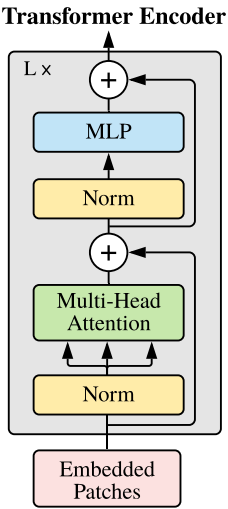
- **Token Embeddings**: The input text is broken down into individual tokens (e.g., words, sub-words). Each token is analogous to a patch in an image.
- **Word Embeddings**: Each token is converted into a fixed-length vector representation (embedding) using a technique like Word2Vec, GloVe, or learned embeddings within the transformer model itself.
- **Self-Attention on Tokens**: The transformer's self-attention mechanism is applied to these word embeddings. This allows the model to capture relationships between different words in the sentence, understanding context and dependencies.

## Processing Images

- **Patch Embeddings**: ViT splits an image into fixed-size patches (e.g., 16x16 pixels). Each patch is treated like a token in NLP, much like words in a sentence.
- **Linear Embedding**: Each patch is flattened into a 1D vector and then linearly projected to a fixed-length embedding.
- **Self-Attention on Patches**: The Transformer's self-attention mechanism is then applied to these patch embeddings, allowing the model to learn relationships between patches.

# ViT Overview



## Positional Encoding

- Positional Encoding: Since Transformers process all patches simultaneously (losing the inherent spatial structure of the image), positional encodings are added to the patch embeddings to maintain the relative position of each patch.
- Global Awareness with Position Information: These encodings allow the Transformer to understand not just the content of the patches, but where each patch is located within the image.

# Inspecting Vision Transformer

- Self-attention allows ViT to integrate information across the entire image even in the lowest layers.

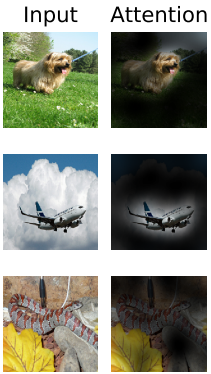



Figure 1: Representative examples of attention from the output token to the input space.

## Inductive Bias

- 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

# ViT vs. CNN

- Local vs. Global Understanding: While CNNs are good at local feature extraction, ViT captures global relationships from the very first layer.
- Efficiency on Large Datasets: ViT shines on large datasets, while CNNs, due to their inductive biases (local patterns, translation invariance), perform better on smaller datasets without needing as much data.
- Simplicity in Architecture: ViT uses a simpler architecture with fewer specialized layers, while CNNs rely on task-specific layers like convolutions and pooling.

## Combining CNNs and Transformers

- As an alternative to raw image patches, the input sequence can be formed from feature maps of a CNN.
- These hybrid models combine the local feature extraction capabilities of CNNs with the global context awareness of Transformers.
- These models can perform well even on smaller datasets.

# Dataset Size

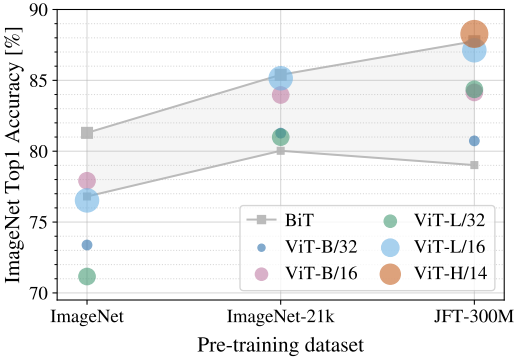



Figure 2: While large ViT models perform worse than BiT ResNets when pre-trained on small datasets, they shine when pre-trained on larger datasets.

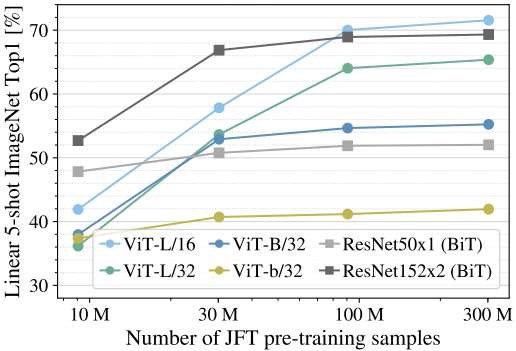



Figure 3: ResNets perform better with smaller pre-training datasets but plateau sooner than ViT, which performs better with larger pre-training.

# Performance versus Pre-training

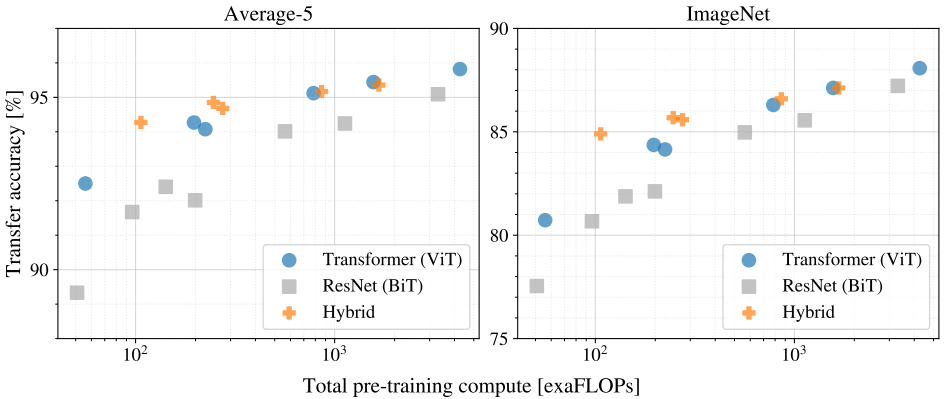



Figure 4: Vision Transformers generally outperform ResNets with the same computational budget. Hybrids improve upon pure Transformers for smaller model sizes, but the gap vanishes for larger models.

# Contributions

- These slides have been prepared thanks to:
  - Ramtin Moslemi
  - Ashkan Majidi

# References

[1] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit, and N. Houlsby, "An image is worth 16x16 words: Transformers for image recognition at scale," 2021.