Machine Learning (CE 477)

Fall 2024

Ali Sharifi-Zarchi

CE Department
Sharif University of Technology

December 29, 2024

Recap: Self-Attention Mechanism

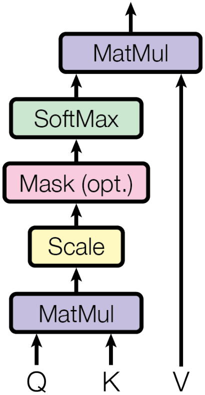


Figure 1: Scaled Dot-Product Attention.

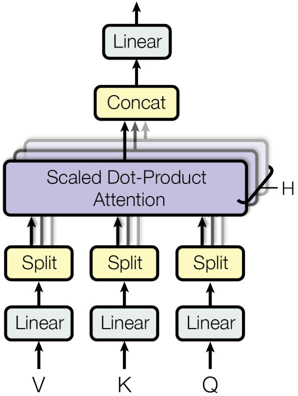


Figure 2: Multi-Head Attention consists of several attention layers running in parallel.

Strengths and Limitations of CNNs

- CNNs excel at:
  - *Local Feature Extraction*
  - *Translation Invariance*
  - *Efficient Computation*
- However, they have limitations due to:
  - *Limited Receptive Field*
  - *Pooling Information Loss*
  - *Struggle with Global Context*

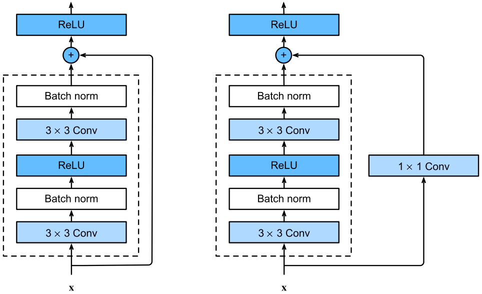


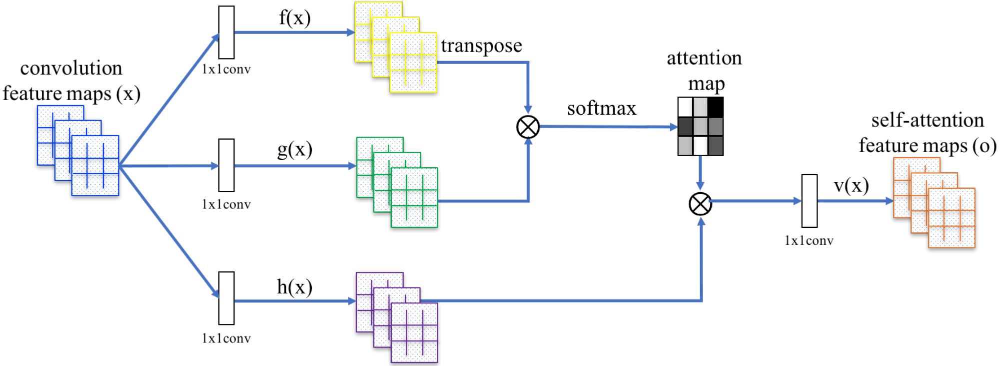
Figure 3: Block diagram of ResNet

Transformers: From Language to Vision

- The Transformer architecture was designed for **sequence-to-sequence learning**.
- Transformers have revolutionized NLP and become the model of choice.
- Is it possible to adapt the Transformer for image data?

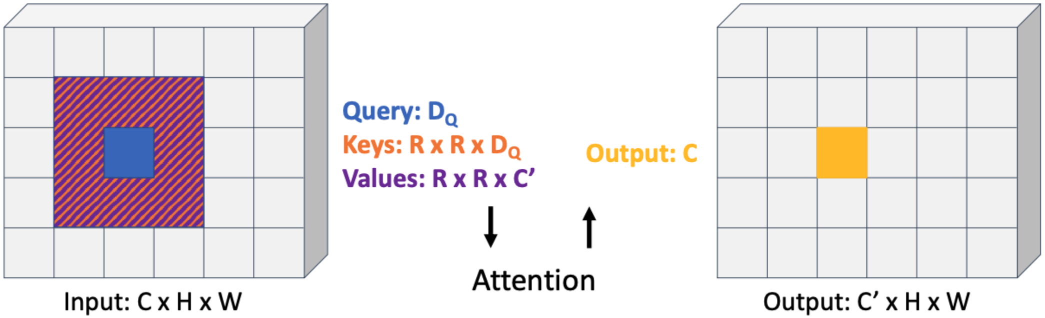
Idea #1: Add Attention to Existing CNNs

- Model is still a CNN!



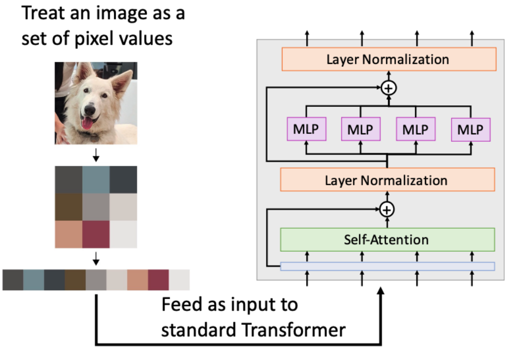
Idea #2: Replace Convolution with "Local Attention"

- Lots of tricky details, hard to implement, only marginally better than ResNets.



Idea #3: Standard Transformers on Pixels

- Insane memory usage!



Idea #4: Standard Transformer on Patches

- Fixes idea #1 by entirely replacing convolutional layers.
- Fixes idea #2 by simplifying implementation.
- Fixes idea #3 by reducing memory usage.
- This is the main idea behind **Vision Transformers**!

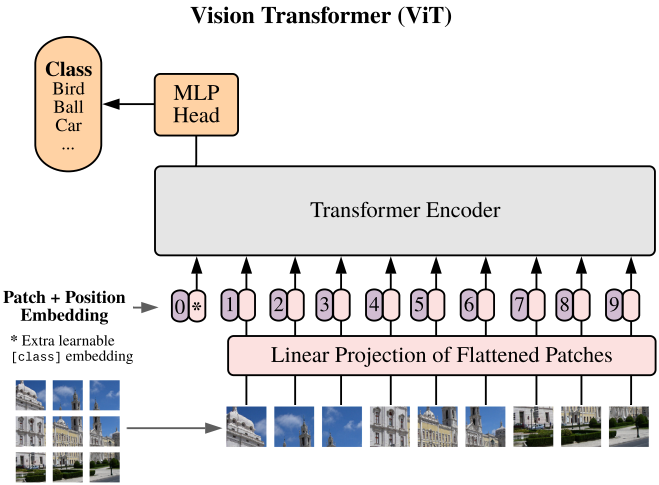
Processing Text

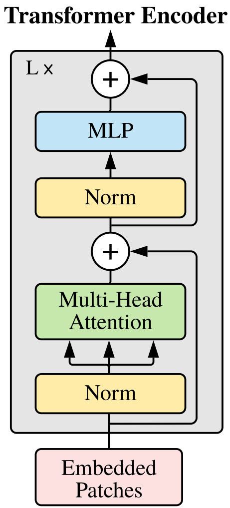
- **Token Embeddings**: The input text is broken down into individual tokens.
- **Word Embeddings**: Each token is converted into a fixed-length vector representation.
- **Self-Attention on Tokens**: The transformer's self-attention mechanism is applied to these word embeddings.

Processing Images

- **Patch Embeddings**: ViT splits an image into fixed-size patches (e.g., 16×16 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





Patch Sizes

- Patch size determines the granularity of the image representation.
- Smaller patches capture finer details but increase computational cost.
- The self-attention mechanism has $\mathcal{O}(N^2)$ complexity due to pairwise interactions between N patches.
- Larger patches reduce computational complexity but might miss finer details.
- In practice:
  - take 224x224 input image,
  - divide it into a 16x16 grid of 14x14 pixel patches (or 14x14 grid of 16x16 patches)
- **An Image is Worth 16x16 Words!**

Positional Embeddings

- We need to inject spatial information into the model to ensure the model understands the order of image patches.
- There are several options:
  - Providing no positional information: *bag of patches*
  - 1-dimensional positional embedding: sequence of patches in the raster order
  - 2-dimensional positional embedding: concatenate embedding of different axes
  - Relative positional embeddings: relative distance instead of absolute position
- In practice 1-dimensional positional embedding work best.

Combining Patches and Embeddings

1. **Divide Image into Patches**
   - Image of size $H \times W$ divided into patches of size $P \times P$.
   - Number of patches: $N = \frac{H \times W}{P \times P}$.
2. **Flatten and Project Patches**

$$\text{patch\_embedding}_i = \text{Linear}(\text{flatten}(x_i))$$

3. **Add Positional Embeddings**

$$\text{combined\_embedding}_i = \text{patch\_embedding}_i + PE_i$$

4. **Form Input Sequence**

$$\text{Input} = [\text{CLS}; \text{combined\_embedding}_1; \ldots; \text{combined\_embedding}_N]$$

Transformer Encoder

- Now that we have an **Input Sequence**, we can use a simple Transformer Encoder.
- A Transformer encoder consists of:
  - Multi-Head Attention: Learns relationships between all tokens (image patches) in the sequence.
  - Feedforward Network: Processes each token independently in a higher-dimensional space.
  - Residual Connections & Layer Normalization: Helps with optimization.
- Finally we can pass the output of the encoders to an MLP classifier head in order to predict the correct class.

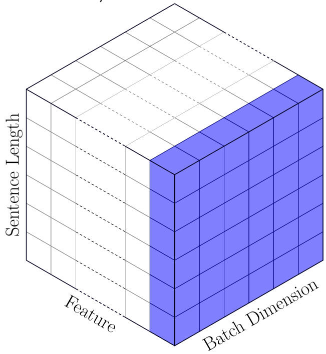
Batch Normalization

- Normalizes inputs across the mini-batch.

$$\hat{x}_{ij} = \frac{x_{ij} - \mu_j}{\sqrt{\sigma_j^2 + \epsilon}}$$

  where μ_j and σ_j^2 are the mean and variance of the j-th feature in the mini-batch.
- Appropriate for CNNs.
- Requires large batch sizes.
- Improves generalization via regularization.

Batch/Power Normalization



Layer Normalization

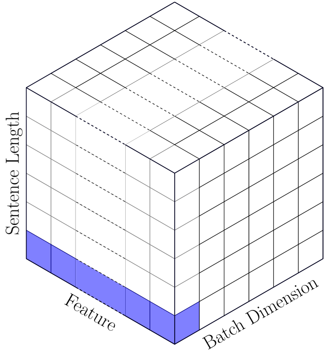
- Normalizes inputs across the features in a single training example.

$$\hat{x}_{ij} = \frac{x_{ij} - \mu_i}{\sqrt{\sigma_i^2 + \epsilon}}$$

where μ_i and σ_i^2 are the mean and variance of the i-th training example.

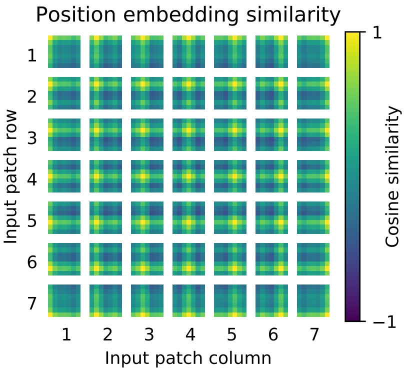
- Appropriate for Transformers, RNNs, or NLP tasks.
- Works with small or dynamic batch sizes.
- Provides batch-independent normalization for flexibility.

Layer Normalization



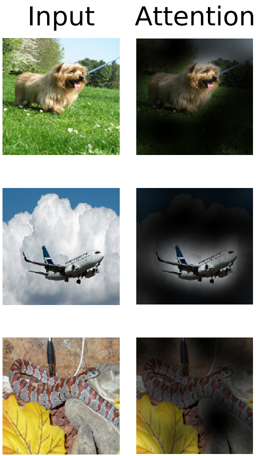
What Do Positional Embeddings Learn?

- The model learns to encode distance within the image in the similarity of position embeddings.
- This means closer patches tend to have more similar position embeddings.
- Patches in the same row/column have similar embeddings.



How Does Attention Help?

- Self-attention allows ViT to integrate information across the entire image even in the lowest layers.
- We can average attention weights across all heads and recursively multiply the weight matrices of all layers to mix attention across tokens through all layers.



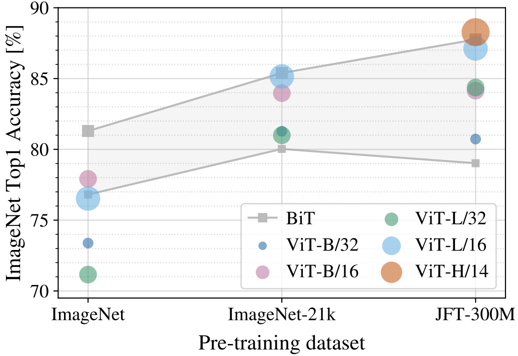
Inductive Bias

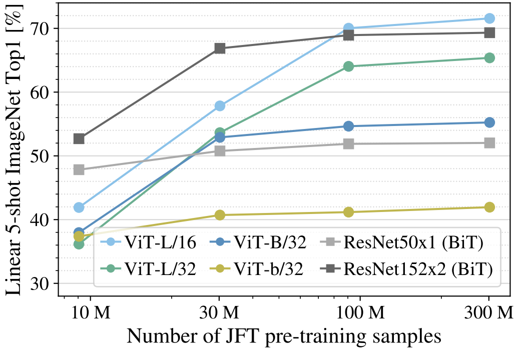
- In CNNs, locality, 2D neighborhood structure, and translation equivariance are baked into each layer throughout the whole model.
- In ViT, only MLP layers exhibit locality and translational equivariance, while self-attention layers capture global context.
- The 2D neighborhood structure is used very sparingly.

Combining CNNs and Transformers

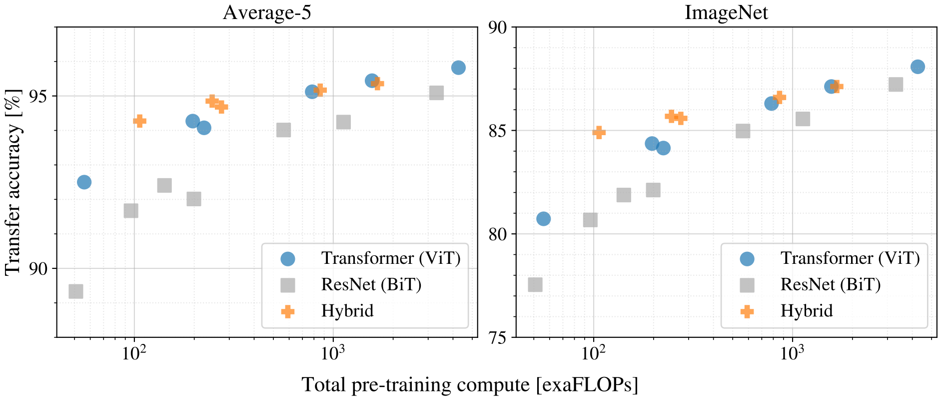
- As an alternative to raw image patches, the input sequence can be formed from feature maps of a CNN.
- These hybrid models combine the local feature extraction capabilities of CNNs with the global context awareness of Transformers.
- These models can perform well even on smaller datasets.

Dataset Size



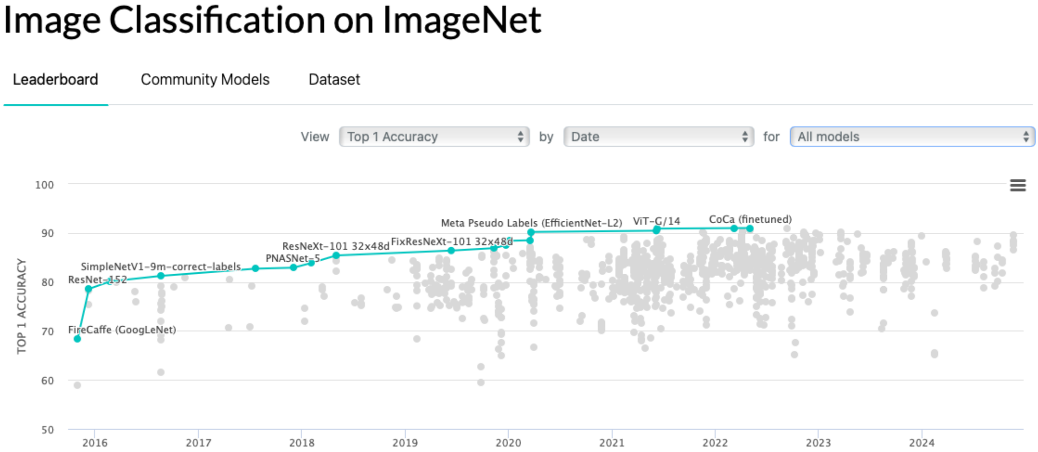


Performance vs. Pre-training



State-of-the-Art

- Transformers achieve State-of-the-Art results in a variety of computer vision tasks, including **Classification**, **Segmentation**, **Detection**, and more!



Conclusion

- **Advantages of ViTs**
  - **Better with Scale**: Performance improves significantly as dataset size and model size grow.
  - **Unified Architecture**: The same Transformer blocks can be applied to text, images, and other modalities.
  - **Interpretability**: Attention maps can provide insights into which patches are influencing the final decision.
- **Disadvantages of ViTs**
  - **Data Hungry**: ViTs typically require large pretraining datasets.
  - **Computational Cost**: Multi-head self-attention scales quadratically with the number of tokens. For high-resolution images, this can be expensive.
  - **Overfitting**: High capacity can lead to overfitting.

Contributions

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

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.

[2] H. Zhang, I. Goodfellow, D. Metaxas, and A. Odena, “Self-attention generative adversarial networks,” 2019.

[3] J. Johnson, “Eecs 498.008 / 598.008: Deep learning for computer vision, winter 2022, lecture 18: Vision transformers,” 2022.
Accessed: 2024-12-14.

[4] A. Zhang, Z. C. Lipton, M. Li, and A. J. Smola, *Dive into Deep Learning*. Cambridge University Press, 2023.
https://D2L.ai.

[5] S. Shen, Z. Yao, A. Gholami, M. W. Mahoney, and K. Keutzer, “Rethinking batch normalization in transformers,” *CoRR*, vol. abs/2003.07845, 2020.