Transformers: Attention Mechanisms and Architecture

AI/ML Learning Notes

October 5, 2025

Contents

1 Introduction

Transformers represent a paradigm shift in sequence modeling, moving away from recurrent architectures to attention-based mechanisms. Introduced by Vaswani et al. in 2017, transformers have become the foundation of modern natural language processing and are increasingly applied to computer vision and other domains.

2 Mathematical Foundations

2.1 Attention Mechanism

The core innovation of transformers is the scaled dot-product attention mechanism:

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V \tag{1}$$

where:

- $Q \in \mathbb{R}^{n \times d_k}$ is the query matrix
- $K \in \mathbb{R}^{m \times d_k}$ is the key matrix
- $V \in \mathbb{R}^{m \times d_v}$ is the value matrix
- d_k is the dimension of keys/queries
- d_v is the dimension of values
- n is the number of queries
- m is the number of key-value pairs

The scaling factor $\frac{1}{\sqrt{d_k}}$ prevents the dot products from growing too large, which would push the softmax function into regions with extremely small gradients.

2.2 Multi-Head Attention

Multi-head attention allows the model to jointly attend to information from different representation subspaces:

$$\text{MultiHead}(Q, K, V) = \text{Concat}(\text{head}_1, \ldots, \text{head}_h)W^O \tag{2}$$

where each head is computed as:

$$\text{head}_i = \text{Attention}(QW_i^Q, KW_i^K, VW_i^V) \tag{3}$$

The projection matrices are:

- $W_i^Q \in \mathbb{R}^{d_{model} \times d_k}$
- $W_i^K \in \mathbb{R}^{d_{model} \times d_k}$
- $W_i^V \in \mathbb{R}^{d_{model} \times d_v}$
- $W^O \in \mathbb{R}^{hd_v \times d_{model}}$

2.3 Positional Encoding

Since transformers don't have inherent notion of sequence order, positional encodings are added to input embeddings:

$$PE_{(pos,2i)} = \sin\left(\frac{pos}{10000^{2i/d_{model}}}\right) \tag{4}$$

$$PE_{(pos,2i+1)} = \cos\left(\frac{pos}{10000^{2i/d_{model}}}\right) \tag{5}$$

where:

- pos is the position in the sequence
- i is the dimension index
- d_{model} is the model dimension

3 Architecture Details

3.1 Encoder Layer

Each encoder layer consists of:

1. Multi-head self-attention mechanism
2. Add & Norm (residual connection + layer normalization)
3. Position-wise feed-forward network
4. Add & Norm

The feed-forward network is defined as:

$$\text{FFN}(x) = \max(0, xW_1 + b_1)W_2 + b_2 \tag{6}$$

3.2 Decoder Layer

Each decoder layer extends the encoder with:

1. Masked multi-head self-attention
2. Add & Norm
3. Multi-head cross-attention (attending to encoder output)
4. Add & Norm
5. Position-wise feed-forward network
6. Add & Norm

3.3 Layer Normalization

Layer normalization normalizes across features:

$$\text{LayerNorm}(x) = \gamma \odot \frac{x - \mu}{\sqrt{\sigma^2 + \epsilon}} + \beta \tag{7}$$

where μ and σ^2 are the mean and variance computed across the feature dimension.

4 Training Considerations

4.1 Loss Function

For language modeling, the cross-entropy loss is used:

$$\mathcal{L} = -\sum_{t=1}^{T} \log P(y_t | y_{<t}, x) \tag{8}$$

4.2 Optimization

The Adam optimizer is typically used with learning rate warm-up:

$$lr = d_{model}^{-0.5} \cdot \min(step^{-0.5}, step \cdot warmup\_steps^{-1.5}) \tag{9}$$

5 Complexity Analysis

5.1 Computational Complexity

- Self-attention: $O(n^2 \cdot d)$ where n is sequence length, d is model dimension
- Feed-forward: $O(n \cdot d^2)$
- Total per layer: $O(n^2 \cdot d + n \cdot d^2)$

5.2 Memory Complexity

The attention mechanism requires $O(n^2)$ memory for the attention matrix, which becomes a bottleneck for long sequences.

6 Variants and Extensions

6.1 BERT (Bidirectional Encoder Representations from Transformers)

Uses only the encoder stack with masked language modeling and next sentence prediction objectives.

6.2 GPT (Generative Pre-trained Transformer)

Uses only the decoder stack with causal language modeling for autoregressive generation.

6.3 Vision Transformers (ViT)

Applies transformers to image patches, treating them as tokens in a sequence.

7 Practical Considerations

7.1 Hyperparameters

Common configurations:

- $d_{model} = 512$ or 768
- $h = 8$ or 12 heads

- $d_k = d_v = d_{model}/h$
- $d_{ff} = 2048$ or 3072 (feed-forward dimension)
- Number of layers: 6-24

7.2 Regularization

- Dropout applied to attention weights and feed-forward outputs
- Label smoothing for regularization
- Weight decay in optimizer

8 Conclusion

Transformers have revolutionized deep learning by demonstrating that attention mechanisms alone, without recurrence or convolution, can achieve state-of-the-art results across various domains. Their parallel processing capability and ability to capture long-range dependencies make them the architecture of choice for modern AI systems.

9 References

1. Vaswani, A., et al. (2017). Attention is all you need. In NIPS.
2. Devlin, J., et al. (2018). BERT: Pre-training of deep bidirectional transformers for language understanding. arXiv preprint.
3. Radford, A., et al. (2019). Language models are unsupervised multitask learners.
4. Dosovitskiy, A., et al. (2020). An image is worth 16x16 words: Transformers for image recognition at scale. ICLR.