Generative AI Transformer

Mrunal Vairagade

December 2024

Abstract

This report explores the foundational concepts behind Generative AI Transformers. We discuss neural networks, activation functions, backpropagation, and attention mechanisms in the context of transformer architectures. These concepts are crucial for understanding modern deep learning models used in natural language processing (NLP) and other domains.

1 Introduction

Generative AI has revolutionized the field of artificial intelligence, particularly in tasks involving text generation, translation, and image synthesis. Transformers form the backbone of modern generative models, relying on attention mechanisms to process input sequences efficiently. This report provides a detailed exploration of key components involved in transformer architectures.

2 Neural Networks

A neural network consists of layers of interconnected neurons that process and learn patterns from data. The basic components include:

- **Input Layer**: Receives raw data inputs.
- **Hidden Layers**: Contain neurons that transform inputs through learned weights and activation functions.
- **Output Layer**: Produces final predictions or classifications.

Feedforward neural networks (FNNs) serve as the foundation for deep learning models, including transformers.

3 Activation Functions

Activation functions introduce non-linearity in neural networks, enabling them to model complex relationships. Common activation functions include:

- **ReLU (Rectified Linear Unit)**: $f(x) = \max(0, x)$, widely used due to its efficiency and ability to mitigate the vanishing gradient problem.
- **Sigmoid**: $f(x) = \frac{1}{1+e^{-x}}$, useful for binary classification but suffers from saturation.
- **Softmax**: Used in classification problems to normalize outputs into probabilities.
- **GELU (Gaussian Error Linear Unit)**: Used in transformers for improved gradient flow.

4 Backpropagation

Backpropagation is the key algorithm for training neural networks. It consists of:

1. **Forward Pass**: Compute predictions using current weights.
2. **Loss Calculation**: Compare predictions with actual outputs using a loss function.
3. **Backward Pass**: Compute gradients of the loss function with respect to weights using the chain rule.
4. **Weight Update**: Adjust weights using gradient descent or its variants (e.g., Adam, RMSprop).

Backpropagation allows networks to learn from data iteratively. The weight update rule using gradient descent is given by:

$$w^{(t+1)} = w^{(t)} - \eta \frac{\partial L}{\partial w} \tag{1}$$

where $w^{(t)}$ is the weight at iteration t, η is the learning rate, and $\frac{\partial L}{\partial w}$ is the gradient of the loss function with respect to the weight.

5 Attention Mechanism

The attention mechanism enables transformers to process long-range dependencies in sequences efficiently. The core components of attention are:

- **Query** (Q), **Key** (K), and **Value** (V) matrices.
- **Scaled Dot-Product Attention**:

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right) V \tag{2}$$

- **Multi-Head Attention**: Multiple attention heads allow the model to focus on different parts of the input.

6 Transformer Architecture

Transformers consist of encoder and decoder blocks:

- **Encoder**: Processes input sequences using self-attention and feedforward layers.
- **Decoder**: Generates output sequences while attending to encoder outputs.
- **Layer Normalization and Residual Connections**: Help stabilize training and improve convergence.

The transformer model eliminates the need for recurrence, making it highly parallelizable and efficient.

7 Conclusion

This report covered key concepts behind Generative AI Transformers, including neural networks, activation functions, backpropagation, and attention mechanisms. These elements contribute to the success of modern NLP models such as GPT and BERT.