Transformers and Multi-Head Attention: An Implementation and Analysis

Giovanni Mantovani and Andrea Palmieri

June 8, 2024

1 Introduction

Transformers have emerged as a powerful architecture for various sequence-to-sequence tasks, leveraging the self-attention mechanism to effectively capture long-range dependencies within sequences. This report focuses on the application of transformers to a specific sequence-to-sequence task: *reversing the order of elements in an input sequence.*

Given a sequence of N numbers between 0 and M, the task is to reverse the input sequence. Although this task sounds straightforward, Recurrent Neural Networks (RNNs) can struggle with such problems that require long-term dependencies. Transformers, on the other hand, are designed to handle long-range dependencies effectively, and we expect them to perform well on this task. The performances of the Transformer network with increasingly longer sequence length will be evaluated against a RNN with the same sequence lengths.

2 Methodology

2.1 Model Architecture

We implement a transformer-based encoder-decoder model, consisting of a multi-headed self-attention module, feed-forward layers, and positional encodings. The encoder processes the input sequence and generates a contextualized representation, while the decoder generates the output sequence one token at a time, attending to the encoder's output and the previously generated tokens.

We also implemented a RNN with 32 as hidden dimension and 5 layers, followed by a fully connected layer.

2.1.1 Encoder

The encoder consists of a stack of identical layers, each containing a multi-head self-attention sublayer and a position-wise feed-forward sublayer. The multi-head attention mechanism allows the encoder to attend to different representations of the input sequence in parallel, capturing dependencies among tokens. The feed-forward sublayer applies a non-linear transformation to each position independently, enabling the encoder to capture more complex features.

2.1.2 Decoder

The decoder also consists of a stack of identical layers, each containing a multi-head self-attention sublayer, a multi-head encoder-decoder attention sublayer, and a position-wise feed-forward sublayer. The self-attention sublayer allows the decoder to attend to the previously generated tokens, while the encoder-decoder attention sublayer enables the decoder to attend to the encoder's output. The feed-forward sublayer applies a non-linear transformation, similar to the encoder.

2.2 Training and Evaluation

The model is trained using teacher forcing, where the ground truth output sequence is fed as input to the decoder during training. The loss function is the cross-entropy between the predicted output sequence and the ground truth sequence.

3 Results and Discussion

In this section, we will present and discuss the results.

Table 1: Results with different sequence lengths

Network	Sequence Length	Validation Accuracy	Test Accuracy
Transformer	50	100.00%	100.00%
Transformer	100	100.00%	100.00%
Transformer	150	95.83%	95.82%
Transformer	200	87.93%	87.91%
Transformer	250	65.85%	65.94%
RNN	50	16.47%	16.38%
RNN	100	13.44%	13.52%
RNN	150	11.90%	11.83%
RNN	200	10.90%	10.87%
RNN	250	10.66%	10.63%

The **transformer model demonstrates excellent performance on shorter sequences** (50 and 100 elements), achieving perfect accuracy on both validation and test sets. However the **performance** of the transformer model **declines as the sequence length increases**. The most significant drop is observed for sequences of length 250, where the model achieves only around 65.9% accuracy. This degradation in performance with increasing sequence length suggests that while the transformer model is highly capable of handling shorter sequences, whereas it struggles with longer sequences.

In contrast to the transformer model, the **RNN model shows consistently poor performance** across all sequence lengths. The validation and test accuracies for sequences of length 50 start at around 16.4% and decrease further with increasing sequence length. For sequences of length 250, the accuracies are as low as approximately 10.6%.

This contrast in performance between the RNN and transformer models highlights the limitations of traditional RNNs in handling long-range dependencies and capturing complex patterns in sequences. RNNs often suffer from issues such as vanishing and exploding gradients, which hinder their ability to learn effectively from long sequences.

4 Conclusion

Despite the simplicity of the task, this experiment served as a valuable benchmark for evaluating the transformer architecture's ability to model sequences and capture dependencies. By successfully reversing the input sequence, the model demonstrates its capability to understand and manipulate sequential data, a crucial skill for more complex sequence-to-sequence tasks such as machine translation and text summarization.