# TRANSFORMERS - ASSIGNMENT 10

**Mirko Morello**
920601
m.morello11@campus.unimib.it

**Andrea Borghesi**
916202
a.borghesi1@campus.unimib.it

May 31, 2024

## 1 INTRODUCTION

In this assignment, we explore and analyze the performance of Transformers [4] for predicting the reversed sequence of a series of numbers compared with RNNs on the same task.

## 2 TECHNOLOGIES USED

### 2.1 Transformers

Transformers are a type of neural network architecture that has revolutionized various natural language processing (NLP) tasks. Unlike traditional recurrent neural networks (RNNs), transformers rely solely on attention mechanisms, allowing them to capture long-range dependencies more effectively. The transformer architecture, introduced in the seminal paper "Attention is All You Need" by Vaswani et al. [4], consists of an encoder and a decoder, both built using multi-head self-attention and feed-forward layers. Transformers have achieved state-of-the-art performance on tasks such as machine translation, text summarization, and language modeling.

### 2.2 Recursive Neural Networks

Recurrent Neural Networks (RNNs) have emerged as a powerful class of neural network architectures for modeling sequential data, such as text, speech, and time series data. Unlike traditional feedforward neural networks, RNNs are designed to process inputs sequentially, allowing them to capture and model the temporal dependencies and patterns present in sequential data [3].
The core component of an RNN is the recurrent cell, which takes the current input and the previous hidden state as inputs and produces the current hidden state and output. This recurrent structure enables RNNs to maintain a memory of past inputs, making them well-suited for tasks that require capturing long-range dependencies.

## 3 ARCHITECTURE DETAILS

### 3.1 Transformers

The implementation focused solely on the Transformer Encoder component shown in Figure 1, which was coupled with an output head for classification tasks. This approach aimed to showcase the long-term memory capabilities of the attention mechanism employed in Transformers. The architecture consisted of a single-layer Transformer Encoder with a single attention head. During the evaluation process, the length of the input context was kept consistent with the length of the generated sequence, while the embedding length was maintained at approximately 32 dimensions. By adopting this simplified architecture, the emphasis was placed on demonstrating the ability of the attention mechanism to capture long-range dependencies effectively, a key strength of Transformer models.

### 3.2 RNNs

Several RNN architectures have been tested, including a simple RNN, a Long Short-Term Memory (LSTM) [2], and a Gated Recurrent Unit (GRU) [1]. Although the increasing complexity from the basic RNN to LSTM and GRU is significant, the results, as we will demonstrate in the results section, do not differ substantially among these architectures. To maintain a dynamic architecture and adapt to the increasing sequence length, we set the hidden size of each architecture to be equal to the sequence length. Each RNN architecture comprised 6 stacked layers. The motivation behind matching the hidden size to the sequence length was twofold. First, it allowed the

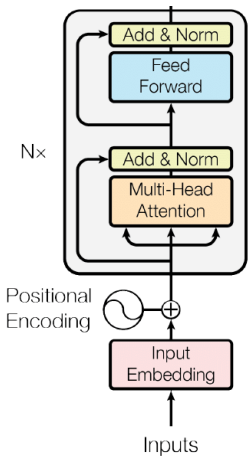


Figure 1: Transformer encoder architecture

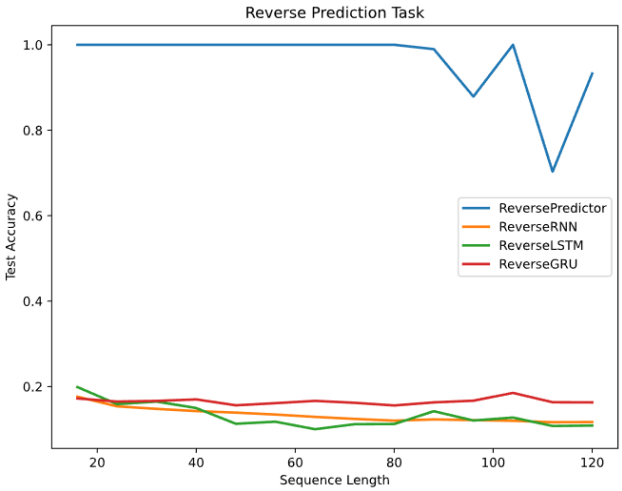


Figure 2: Accuracy of the experimented architecture for a varying sequence length

RNN architectures to dynamically adjust their capacity based on the complexity of the task, potentially improving their ability to capture long-range dependencies. Second, it prevented the Transformer architecture from having an unfair advantage over the RNNs due to its larger context size for longer sequences by design. Despite the increasing complexity from the basic RNN to LSTM and GRU, which aim to mitigate the vanishing gradient problem and better capture long-term dependencies, their performance remained comparable, as we will showcase in the results section. This observation suggests that the inherent limitations of recurrent architectures in handling very long sequences may persist, even with more advanced variants like LSTMs and GRUs.

## 4 Data

The model was trained on an artificial dataset of random sequences of numbers where the labels are the reversed sequences. Both the range of integer numbers and the length of the generated sequences are parametrized but for our experimentations we went for 10 unique numbers, from 0 to 9 and we tested different sequence lengths.

Since the data is completely synthetic there was no lack of data as we could generate as much as needed.

## 5 Results

To comprehensively evaluate the performance of the proposed architectures, we conducted tests on a range of sequence lenghts from 16 to 128 with a step of 8.

### 5.1 Transformers

As anticipated, the Transformer architecture achieved accuracies always very close to 1.0. While there was a slight deterioration in performance with longer sequences, we suspect that this behavior can be attributed to insufficient training, as we trained the model for a fixed 20 epochs, regardless of the sequence length under consideration. The remarkable performance of the Transformer can be attributed to its attention mechanisms, which enable it to effectively capture long-term dependencies with ease. Additionally, the fact that the context length was always equal to the sequence length contributed to the high accuracy. However, it is important to note that if the context length were shorter than the sequence length, the first `seq_len` - `context_len` predictions would have to be made randomly, potentially leading to incorrect predictions.

### 5.2 RNN

On the other hand, RNNs exhibited significant difficulties in keeping track of long-term dependencies as the sequences grew longer, resulting in substantially lower accuracy scores.

The poor performance of RNNs in predicting the reverse sequence of a random sequence of numbers can be attributed to the inherent limitations of their architecture. Unlike Transformers, which can effectively capture long-range dependencies through their attention mechanisms, RNNs process sequences in a sequential manner, making it challenging to maintain information from earlier

time steps as the sequence length increases. As the sequence length grows, the RNN model has to retain and propagate information through its hidden state over multiple time steps. This process can lead to the vanishing or exploding gradient problem, where the gradients either become too small or too large, hindering the model's ability to learn long-term dependencies effectively. Furthermore, the random nature of the input sequences exacerbates the difficulty for RNNs. Without any inherent patterns or structures to leverage, the model must rely solely on its internal representations to memorize and reverse the sequence, a task that becomes increasingly challenging as the sequence length increases. The remarkably low accuracy scores, particularly for longer sequences, highlight the significant struggles faced by RNNs in this task. While RNNs can perform reasonably well on shorter sequences, their performance degrades rapidly compared to Transformers as the sequence length grows, further emphasizing the advantage of the attention mechanism in capturing long-range dependencies.

## REFERENCES


[1] Kyunghyun Cho, Bart van Merrienboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. Learning phrase representations using rnn encoder-decoder for statistical machine translation, 2014.

[2] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9:1735–80, 12 1997.

[3] David E Rumelhart, Geoffrey E Hinton, and Ronald J Williams. Learning internal representations by error propagation, parallel distributed processing, explorations in the microstructure of cognition, ed. de rumelhart and j. mcclelland. vol. 1. 1986. *Biometrika*, 71:599–607, 1986.

[4] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need, 2023.