Vrije Universiteit Amsterdam

Bachelor's Thesis

Investigating the Impact of Recurrent Connections on Pre-Trained Transformer-based Models for Modeling Long-Range Dependencies in Sequential Tasks

Author: Filip Muntean (2663515)

1st supervisor: Peter Bloem
daily supervisor: supervisor name (company, if applicable)
2nd reader: supervisor name

A thesis submitted in fulfillment of the requirements for VU Bachelor of Science degree in Computer Science

June 4, 2023

"I am the master of my fate, I am the captain of my soul"

from Invictus*, by William Ernest Henley*

Abstract

Recent advancements in transformer-based models, such as XL Networks, have achieved remarkable performance by leveraging self-attention mechanisms instead of recurrent connections. In this paper, we present a comprehensive analysis of integrating a recurrent connection into a GPT-2 pre-trained model. Our study involves modeling the recurrent connection, training the model on the Wikipedia Hutter Challenge data set, and conducting thorough performance evaluations using the DAS-5 SLURM Cluster. By addressing this research question we examine the potential benefits and trade-offs associated with incorporating a recurrent connection into the XLNet architecture. Through our study, we shed light on the impact of integrating recurrent connections, digging into modeling sequential patterns, handling complex text tasks, and capturing nuanced dependencies. Overall, we have evaluated the computational requirements and training dynamics of the modified XLNet with recurrent connections, encompassing factors such as memory usage, training time, and convergence properties. make ending more conclusive These findings contribute to understanding of the interplay between recurrent connections and transformer-based models, guiding future developments in natural language processing.

To my parents, thank you for choosing to invest in me everyday.

Contents

List of Figures **v**

List of Tables **vii**

1 Introduction **1**

2 Background **3**

2.1 Pre-trained models 3

2.2 Recurrent connections 3

3 Overview **5**

3.1 Model 5

3.2 Dataset 5

4 Design **7**

5 Evaluation **9**

6 Discussion **11**

7 Threats To Validity **13**

7.1 Internal Validity 13

7.2 External Validity 13

7.3 Construct Validity 13

7.4 Conclusion Validity 13

8 Related Work **15**

9 Conclusion **17**

References **19**

10 Appendix **21**

10.1 Defining terms 21

10.1.1 Recurrent Connection 21

10.1.2 Self-Attention 21

10.1.3 XLNet 21

10.1.4 Baseline XLNet 21

10.1.5 Evaluating Metrics 22

10.1.6 Computational Requirements 22

10.1.7 Effectively Handling Complex Sequential Tasks 22

10.2 Other works worth mentioning 22

10.2.1 Spike GPT 22

List of Figures

LIST OF FIGURES

List of Tables

LIST OF TABLES

1

Introduction

In recent years, Recurrent Neural Networks (RNNs) have demonstrated their efficacy in capturing sequential dependencies and effectively modeling sequential data (1, 2). However, they have faced challenges in capturing long-range dependencies effectively, as the information flow is constrained by the sequential nature of processing(3), (4). While recurrent connections have been traditionally used for this purpose, they suffer from limitations such as vanishing gradients and slow convergence.

rewrite so it's more on topic:

Modify: Nevertheless, there have been significant advancements in transformer-based models, such as XL Networks , among others, which have achieved remarkable performance across various natural language processing tasks (5). XL Networks leverage self-attention mechanisms as opposed to traditional recurrent connections, enabling them to excel in modeling bidirectional context and capturing dependencies spanning entire sentences (6).

Nevertheless, pre-trained models have significantly changed the deep learning environment, with researchers making significant progress in constructing increasingly complex and powerful models. This has been due to the introduction of larger datasets and more powerful computational resources, allowing models to be trained on huge volumes of unlabeled data. The availability of pre-trained models and large-scale pretrained language models like BERT, GPT, and RoBERTa has revolutionized deep learning, allowing researchers and practitioners to build on top of learned representations and reach cutting-edge outcomes with fewer data and computer resources. This has thus democratized deep learning, making it more accessible to a wider spectrum of users and speeding up development in a variety of disciplines.

Hence, in this study, we propose addressing the following main research question: *How does the integration of recurrent connections into pre-trained models impact their ability to model long-range dependencies and improve performance on sequential tasks?* The latter then yields the following subquestions:

(1) How can we evaluate the introduction of recurrent connections with regards to the self-attention mechanisms of pre-trained models and whether these can enhance their sequential context modeling capabilities?

(2) How can we compare whether the performance of the modified pre-trained model is superior in terms of modeling sequential patterns, and how effectively can the modified architecture handle complex tasks compared to the baseline and other state-of-the-art models?

(3) Assess the appropriate computational requirements and trade-offs associated with training recurrent connections in a pre-trained model, considering factors such as memory usage, training time, and inference speed.

In the 2^{nd} chapter, we will delve into the background of our research topic, providing the necessary background and reviewing relevant existing models related to our topic. Later, in the 3^{rd}, we will present an overview of the model's architecture and the dataset used for this taks. The following chapter will focus on the design of our proposed modification to pre-trained models, which incorporates recurrent connections. We will discuss the specific architectural changes and adaptations made to enhance the model's sequential context modeling capabilities. Following the design, the 5^{th} part will present our comprehensive evaluation of the modified architecture. Chapter 6 will provide a detailed discussion of the results obtained from our evaluation. We will analyze the implications of incorporating recurrent connections into pre-trained models, discussing the benefits, trade-offs, and areas for further improvement. In the 7^{th} chapter we will address potential threats to the validity of our findings, including limitations and potential biases that may have influenced the outcomes of our research. Section 8 will explore the related work in the field, highlighting previous studies and advancements in the integration of recurrent connections and transformer-based models for language understanding. Finally, we will review the significant findings, examine their implications, and outline prospective future research areas in the realm of recurrent transformers.

2

Background

This section provides the necessary context to help the reader understand the remainder of the thesis.

2.1 Pre-trained models

Pre-trained transformer-based models are unsupervisedly trained on massive volumes of text data, learning to anticipate missing words or create coherent text. These models collect extensive language knowledge and may be modified for specific downstream applications. These models tend, however, to be quite data hungry, meaning that their large number of hyperparameters makes them more likely to overfit and to have a poor generalization ability(7).

2.2 Recurrent connections

Recurrent connections are an important component of recurrent neural networks (RNNs). RNNs feature connections that create loops, enabling information to be transmitted from one step to the next within a sequence, as opposed to feed forward neural networks, which process input data in a strictly forward fashion. This enables RNNs to successfully represent sequential data and capture temporal relationships. The recurrent connections keep a hidden state or memory that remembers past steps and impacts the calculation at each time step. This memory allows the network to learn long-term dependencies and collect context over time. The inclusion of recurrent connections in such RNNs enables them to utilize past context and adjust to stretched or compressed input patterns by altering the pace at which their internal state changes(8). are more robust to temporal warping than non-recursive models.

3

Overview

This section provides a high-level outline of the proposed system or solution. It typically illustrates the system architecture or the interactions between the different solution components (via a "boxes-and-arrows" diagram) from a user's perspective.

3.1 Model

3.2 Dataset

The enwik8 dataset, used in the Hutter Prize Challenge, is a commonly referenced benchmark dataset for testing lossless compression algorithms. It consists of the first 100 million bytes of the English Wikipedia text, excluding XML tags, URLs, and other non-content data. The dataset contains a diverse range of textual information, including articles, discussions, and various types of formatting. It is often used to evaluate the effectiveness of compression algorithms in capturing and representing the complexity and redundancy present in natural language data. This dataset is valuable for assessing the compression performance of algorithms (9) because it provides a standardized and representative set of text data. Its large size challenges algorithms to effectively capture long-range dependencies, statistical patterns, and linguistic structures to achieve high compression ratios. Researchers and developers often use this dataset to compare different compression methods, assess improvements in algorithms, and push the boundaries of lossless compression techniques.

write more details about the dataet

4

Design

In this section, you would provide a high-level description of the system or solution and explain your design choices.

5

Evaluation

Discuss the design of your experiments, the results you obtained, and how they help in evaluating the claims you made in the introduction. You may also use the evaluation results in this section to justify your design choices or assess the contributions of different aspects of your design towards the overall goals.

6

Discussion

Here you put your results in context (possibly grouped by research question). Usually, this section focuses on analyzing the implications of the proposed work for current and future research and for practitioners.

7

Threats To Validity

Report about each type of threat to the validity of the experiment, according to the classification framework proposed by Wohlin *et al.* (**?**).

7.1 Internal Validity

7.2 External Validity

7.3 Construct Validity

7.4 Conclusion Validity

8

Related Work

Describe here scientific papers similar to your experiment, both in terms of goal and methodology. One paragraph for each paper (we expect about 5-8 papers to be discussed). Each paragraph contains: (i) a brief description of the related paper and (ii) a black-on-white description about how your work differs from the related paper. You may place this section immediately after the Background section, if necessary. write how these papers are related to the topic add the other papers here since they are more relevant

Lately, a number of research papers have been published that present attempts to include recurrent connections within a transformer. Namely, "Adding Recurrence to Pretrained Transformers for Improved Efficiency and Context Size"(10) explores the latter, aiming to enhance their ability to model long-range dependencies and sequential patterns. By adding recurrent connections through recurrent neural networks (RNNs), the authors combine the strengths of transformers and RNNs. Additionally, a hybrid sequential model(11) is proposed, combining recurrent and transformer architectures to effectively handle variable-length sequences. Similarly, a study(12) investigates the benefits and efficiency of incorporating recurrence into the Transformer architecture. Results show improved performance without low-level optimizations, demonstrating the potential of this approach for enhancing natural language processing models. These findings have important implications for future research in deep learning architectures.

add Modelling recurrence for transformers (13)

Look into Segmented Recurrent Transformer: An Efficient Sequence-to-Sequence Model Look into Simple Recurrence Improves Masked Language Models

add transformer citation: @inproceedingswolf-etal-2020-transformers, title = "Transformers: State-of-the-Art Natural Language Processing", author = "Thomas Wolf and Lysandre Debut and Victor Sanh and Julien Chaumond and Clement Delangue and Anthony Moi and Pierric Cistac and Tim Rault and Rémi Louf and Morgan Funtowicz and Joe Davison and Sam Shleifer and Patrick

von Platen and Clara Ma and Yacine Jernite and Julien Plu and Canwen Xu and Teven Le Scao and Sylvain Gugger and Mariama Drame and Quentin Lhoest and Alexander M. Rush", booktitle = "Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations", month = oct, year = "2020", address = "Online", publisher = "Association for Computational Linguistics", url = "https://www.aclweb.org/anthology/2020.emnlp-demos.6", pages = "38–45"

9

Conclusion

Briefly summarize your contributions, and share a glimpse of the implications of this work for future research.

References

[1] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. **Attention Is All You Need**. *Advances in Neural Information Processing Systems*, **30**:5998–6008, 2017. 1

[2] Andrej Karpathy. **The Unreasonable Effectiveness of Recurrent Neural Networks**. 2015. 1

[3] Yankai Lin, Zhiyuan Liu, Huanbo Luan, and Maosong Sun. **A Survey of Recent Advances in Deep Learning Models for Natural Language Processing**. *Information Fusion*, **58**:1–12, 2020. 1

[4] Yoshua Bengio, Patrice Simard, and Paolo Frasconi. **Learning Long-Term Dependencies with Gradient Descent is Difficult**. *IEEE Transactions on Neural Networks*, **5**(2):157–166, 1994. 1

[5] Kevin Clark, Minh-Thang Luong, and Quoc V Le. **ELECTRA: Pre-training Text Encoders as Discriminators Rather Than Generators**, 2020. 1

[6] Zihang Dai, Zhilin Yang, Yiming Yang, Jaime Carbonell, and Quoc V Le. **Transformer-XL: Attentive Language Models Beyond a Fixed-Length Context**, 2019. 1

[7] Xu Han, Zhengyan Zhang, Ning Ding, Yuxian Gu, Xiao Liu, Yuqi Huo, Jiezhong Qiu, Yuan Yao, Ao Zhang, Liang Zhang, Wentao Han, Minlie Huang, Qin Jin, Yanyan Lan, Yang Liu, Zhiyuan Liu, Zhiwu Lu, Xipeng Qiu, Ruihua Song, Jie Tang, Ji-Rong Wen, Jinhui Yuan, Wayne Xin Zhao, and Jun Zhu. **Pre-Trained Models: Past, Present and Future**, 2021. 3

[8] title= Graves et al. 3

REFERENCES

[9] Sergey Edunov, Alexei Baevski, and Michael Auli. **Lossy Compression of Neural Machine Translation Models**. In *Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2018. 5

[10] Davis Yoshida, Allyson Ettinger, and Kevin Gimpel. **Adding Recurrence to Pretrained Transformers**. *arXiv preprint arXiv:2104.08691*, 2021. 15

[11] Antonio García-García, Sergio Orts-Escolano, Miguel Cazorla, and José García-Rodríguez. **Recurrent-Transformer: A Hybrid Sequential Model for Classifying Variable-Length Data**. In *International Conference on Pattern Recognition*, pages 8777–8782. IEEE, 2019. 15

[12] Tao Lei, Ran Tian, Jasmijn Bastings, and Ankur P. Parikh. **Simple Recurrence Improves Masked Language Models**, 2022. 15

[13] Jie Hao, Xing Wang, Baosong Yang, Longyue Wang, Jinfeng Zhang, and Zhaopeng Tu. **Modeling Recurrence for Transformer**. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 1198–1207, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics. 15

[14] Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. **Language models are unsupervised multitask learners**. *OpenAI Blog*, **1**(8), 2019.

[15] Rui-Jie Zhu, Qihang Zhao, and Jason K. Eshraghian. **SpikeGPT: Generative Pre-trained Language Model with Spiking Neural Networks**, 2023.

10

Appendix

10.1 Defining terms

10.1.1 Recurrent Connection

By recurrent connections, we refer to a type of connection in neural networks that allows information to flow in a loop. The former enables the network to retain and process sequential information by considering the previous states or outputs. In the context of our research question, recurrent connections are being introduced and evaluated.

10.1.2 Self-Attention

Additionally, self-attention is a mechanism used in transformer-based models, such as XL Networks. It allows the model to weigh the importance of different parts of the input sequence when generating output. Self-attention enables the model to capture dependencies between different positions in the sequence, making it effective for modeling long-range dependencies.

10.1.3 XLNet

Similarly, the term XL Networks refers to a specific transformer-based model, which stands for "Extra-Long Networks." This architecture is recognized for its ability to model bidirectional context and capture dependencies across the entire sequence. XL Networks leverage self-attention mechanisms, making them relevant to the evaluation in this research question.

10.1.4 Baseline XLNet

Eventually, by baseline XLNet, we refer to the original XLNet model without any modifications or additions, which serves as the reference or comparison point for evaluating the performance of the modified architecture. The baseline XLNet represents the state-of-the-art model at the time of

comparison, while other state-of-the art models encompass existing models or approaches that have also demonstrated superior performance or achieved notable results in solving complex sequential tasks.

10.1.5 Evaluating Metrics

Consequently, the performance of XL Networks and the impact of recurrent connections are evaluated using multiple standard metrics (14), including accuracy, precision, recall, F1 score, and area under the ROC curve (AUC-ROC). Good performance with regards to sequence models translates to the accurate prediction of the next word in a sentence. If the model consistently generates the correct word given the context and dependencies in the preceding words, it can be considered to have good performance. For example, if the sentence is "The sun is shining and the ... is blue," and the model predicts "sky" as the next word, it demonstrates good performance.

10.1.6 Computational Requirements

Some appropriate requirements translate to, for exmaple, some additional 2GB of memory usage, some additional 16GB of RAM and 5inference speed. The latter implies the required time to process new input and generate predictions. This ultimately renders a 20this transformation yields a 24h training time into one close to the 30h margin. Other factors such as convergence properties should be taken into account as training time advances.

10.1.7 Effectively Handling Complex Sequential Tasks

What is more, effectively handling complex sequential tasks means that the model can successfully tackle tasks that involve intricate dependencies, long-range dependencies, variable-length input sequences, or complex patterns in the sequential data. It implies that the model can effectively capture and utilize the relevant information from the input sequence to generate accurate predictions or perform the task with high precision. The evaluation of how effectively a model handles complex sequential tasks can be based on its ability to handle challenging scenarios, generalize well to diverse input data, provide robust predictions, and achieve competitive results compared to other models or established benchmarks for those specific tasks.

10.2 Other works worth mentioning

10.2.1 Spike GPT

SpikeGPT is a novel approach to language Spiking neural networks are utilized in a unique way in the study "SpikeGPT: Generative Pre-trained Language Model with Spiking Neural Networks"(15)

to represent language. For more effective language processing that is inspired by the brain, it combines the strength of spiking neural networks with generative pre-training. Unsupervised pre-training and supervised fine-tuning are used to train the model in two steps. The research offers experimental findings that show the potency of SpikeGPT, which employs spiking neural networks to attain competitive performance on a number of language modeling benchmarks. The authors also go through SpikeGPT's prospective uses and effects in the context of neuromorphic computing and brain-inspired artificial intelligence.