Towards Smaller and Faster GPTs

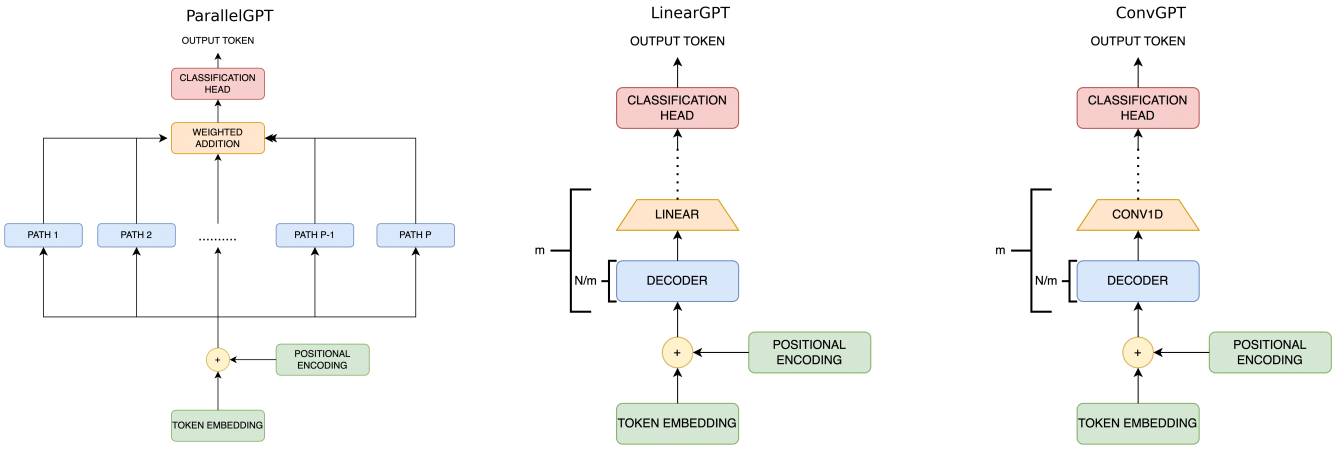

Figure 1: Proposed architectural variants

ABSTRACT

Contemporary efforts in this field of Large Language Models primarily aim to enhance model capabilities by scaling up both the architecture and data volumes. However, there has been little exploration into reducing model sizes while maintaining their effectiveness. In this study, we introduce three modifications to the decoder-only transformer architecture—namely ParallelGPT (*pgpt*), LinearGPT (*lgpt*), and ConvGPT (*cgpt*). These variants demonstrate comparable performances to the conventional architecture, with *lgpt* outperforming it in **4 out of 7 benchmarks** with **less than half the parameters**. We open-source the model weights and the complete codebase for these implementations for further research.

CCS CONCEPTS

• **Computing methodologies** → **Natural language processing**; *Natural language generation.*

KEYWORDS

gpt, parallelism, faster processing

ACM Reference Format:

. 2018. Towards Smaller and Faster GPTs. In *Proceedings of (The Sixth International Conference on Distributed Artificial Intelligence).* ACM, New York, NY, USA, 2 pages. https://doi.org/XXXXXXX.XXXXXXX

The Sixth International Conference on Distributed Artificial Intelligence, Dec 18th – Dec 22nd, 2024, Singapore

ACM ISBN 978-1-4503-XXXX-X/18/06
https://doi.org/XXXXXXX.XXXXXXX

1 INTRODUCTION

Research on Large Language Models (LLMs) has traditionally focused on either scaling the model size to billions or the refinement of pretraining data for relatively smaller models of size 1B to 8B. Though the results are promising, there is an inherent problem that the current research has not explored yet - exploring variants of the GPT architecture that are compute-efficient and effective across multiple tasks. In this study we present 3 architectural variants of the traditional GPT architecture that are either competitive with the traditional architecture or outperforms it. The subsequent sections provides an overview on the architectures and their performances on different benchmarks.

2 ARCHITECTURAL VARIANTS

In this section, we introduce three variant architectures derived from the traditional GPT architecture to address various limitations in training and inference. These architectures are designed to enable faster training and inference. The three proposed architectures are ParallelGPT (pgpt), LinearGPT (lgpt), and ConvGPT (cgpt).

ParallelGPT: The deeper layers in a transformer architecture have very little information to work with and hence do not contribute to the performance of a language model as much as the earlier layers as discussed in [1]. Also the linear flow of information constrains the knowledge to be learned from a single direction. To address these problems, we introduce **ParallelGPT** (*pgpt*), where the N decoder blocks in a traditional GPT architecture are split across P parallel paths with each path containing N/P decoder blocks. The outputs of the parallel paths are combined using a vector of weights $W \epsilon \mathbb{R}^P$, which are learned during training. Each parallel path has its own embedding layer to make sure that each path learns the training data from a different dimension compared to other paths.

ConvGPT and LinearGPT: CNNs while processing an image, progressively downsample the image as the earlier layers learn basic

Table 1: Benchmark Results for Different Models

Benchmark	*gpt*	*pgpt*	*lgpt*	*cgpt*	*pgpt-1*
HellaSwag	**0.2625**	0.2604	0.2517	0.2492	0.2527
WinoGrande	**0.5036**	0.4925	0.4870	0.4751	0.4933
CommonSenseQA	0.1376	**0.1605**	0.1458	0.1540	0.1572
ANLI	0.3300	**0.3350**	0.3290	0.3240	0.3340
PIQA	0.5071	0.5131	0.5185	**0.5256**	0.5125
COPA	0.5300	0.5200	0.5500	0.5000	**0.5600**
ARC Easy	0.2333	0.2439	**0.2491**	0.2298	0.2509

information while the deeper layers learn abstractions over the knowledge of the earlier layers. Similar to that, we hypothesize that when processing text, the embeddings of the tokens need to be downsampled so that the earlier layers learn sentence structures while the deeper layers learn abstractions like word ordering, dependency between words among others. To validate our hypothesis we introduce two architectures called **ConvGPT** and **LinearGPT** which progressively downsample the vector representations of the tokens by introducing conv-1d and linear layers after every n decoder blocks in a traditional GPT architecture. The downsampling makes the architectures faster and controls overfitting.

3 EXPERIMENTS AND RESULTS

We train four different models *gpt*, *pgpt*, *lgpt* and *cgpt* and the common parameters are: *context_size* is 1024, *vocab_size* is 50304, *n_layer* is 8, *n_head* is 8 and *embedding_dim* is 512. For *pgpt*, we set P to 2 and for *lgpt* and *pgpt* we set n to 2. The model size and hardware requirements are listed in Table 2, where *pgpt-1* refers to *pgpt*, with one path dropped during inference. Each of the variants were trained for 5000 steps (25%) on the fineweb-edu [2] dataset (which consists of 10B tokens) using `4xA5000` GPUs.

3.1 Results on various benchmarks

Table 1 presents the various benchmarks on which the models were evaluated on and it can be clearly seen that the architectural variants are either competitive with the traditional *gpt* architecture or outperform it. *lgpt* is the best performing model, outperforming *gpt* on **4 out of the 7 benchmarks**. The comparison of results between *pgpt* and *pgpt-1* is quite surprising since even after dropping a parallel path *pgpt-1*'s performance is almost as similar as *pgpt*. Further research is required on this front, which we leave for future work.

3.2 Advantages of the architectures

The following list of points present the advantages of the proposed architectures:

- pgpt enables parallel training for faster processing.
- pgpt supports dropping of parallel blocks to speed up inference.
- lgpt and cgpt minimize overfitting and increase speed by decreasing block dimensions.
- All the architectures maintain similar performance with fewer parameters.

Table 2: Model Hardware Requirements

Model	# Params	Memory
`gpt`	77.2M	294.53MB
`pgpt`	77.7M	296.53MB
`lgpt`	36.4M	138.95MB
`cgpt`	36.4M	138.95MB
`pgpt-1`	64.8M	247.52MB

4 CONCLUSION

In this paper, we presented architectural variants to the transformer architecture that are faster, have fewer parameters but still perform similarly to the traditional GPT architecture. This opens up several questions and we hope that our work motivates further research along this direction.

REFERENCES

[1] Andrey Gromov, Kushal Tirumala, Hassan Shapourian, Paolo Glorioso, and Daniel A. Roberts. 2024. The Unreasonable Ineffectiveness of the Deeper Layers. *arXiv preprint arXiv:2403.17887* (2024).

[2] HuggingFace. 2024. FineWeb-Edu Dataset. https://huggingface.co/datasets/HuggingFaceFW/fineweb-edu. A dataset of educational web pages curated from the FineWeb corpus, designed for educational and academic applications..

Received 12 October 2024