[Paper Review] Pretrained Transformers As Universal Computation Engines

https://arxiv.org/abs/2103.05247
https://github.com/kzl/universal-computation

Ukjae Jeong, jeongukjae@gmail.com

2021.04.19

Introduction

- Inspired by transformer architecture's successes, the authors address the generalization capabilities in transferring from one modality to another.
- The goal of this paper is investigate finetuning on modalities.
- The authors investigate what pretreated language models (LMs) are capable of in terms of generalizing to other modalities. (Image classification, numerical computations, and protein fold predictions)
- Finetuning linear input and output layers, as well as positional embeddings and layer normalization weights(0.1% of total parameters), the authors show comparable performance in comparison to training full transformer parameters.
- The results suggest that the self-attention layers learned by a language model may have properties amenable to efficient universal computations.

2. Methodology - 2.1 Tasks

- Bit memory
- Bit XOR: `x0 ^ x1 = y`
- ListOps: `[ MAX 4 3 [ MIN 2 3 ] 1 0 ]`
- MNIST: The tokens given to the model are 4 x 4 image patches.(total 64 tokens)
- CIFAR-10: Same with MNIST
- CIFAR-10 LRA: 1 x 1 image patches (total 1024 tokens with dim 1)
- Remote homology detection: predicting protein fold problem. 1024 tokens of dimension 25.

2. Methodology - 2.2 Architecture

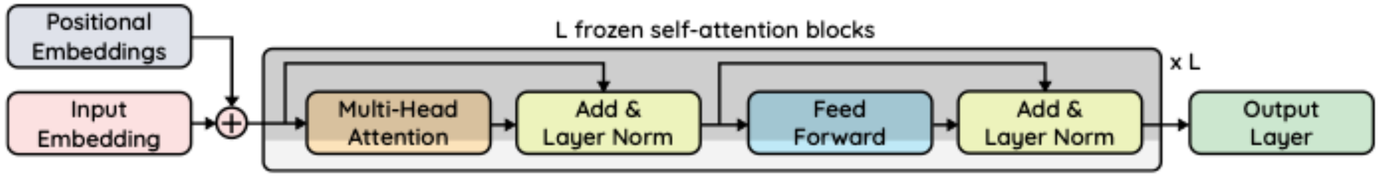

Figure 2: Frozen Pretrained Transformer (FPT). The self-attention & feedforward layers are frozen.

- Output Layer: Single linear layer. $n_{dim} \times d_{out}$ (CIFAR-10: 786 * 10)
- Input Layer: Single linear layer. $n_{in} \times n_{dim}$ (CIFAR-10: 16 * 768)
  - Learning input layer means **learning how to query the transformer.**

3. Empirical Evaluation - Can pretrained language models transfer to different modalities?

- Frozen Pretrained Transformer vs Fully Trained Transformer vs LSTM
- FPT achieves comparable performance than fully trained transformer.
- Because it is difficult to fully train a 12-layer transformer on small datasets, for CIFAR-10, the authors report the full transformer results for a 3-layer model.

Model	Bit Memory	XOR	ListOps	MNIST	CIFAR-10	C10 LRA	Homology
FPT	100%	100%	38.4%	98.0%	72.1%	38.6%	12.7%
Full	100%	100%	38%	99.1%	70.3%	42%	9%
LSTM	60.9%	50.1%	17.1%	99.5%	73.6%	11.7%	12%

Table 1: Test accuracy of FPT vs fully training transformer on downstream task vs fully training LSTM on downstream task (results are transcribed from Figure 1).

3. Empirical Evaluation - What is the importance of the pretraining modality?

- Frozen Pretrained Transformer vs Random initialization vs Bit memory pretraining vs Image pretraining (ViT)

Model	Bit Memory	XOR	ListOps	MNIST	C10	C10 LRA	Homology
FPT	100%	100%	38.4%	98.0%	68.2%	38.6%	12.7%
Random	75.8%	100%	34.3%	91.7%	61.7%	36.1%	9.3%
Bit	100%	100%	35.4%	97.8%	62.6%	36.7%	7.8%
ViT	100%	100%	37.4%	97.8%	72.5%	43.0%	7.5%

Table 2: Test accuracy of language-pretrained (FPT) vs randomly initialized (Random) vs Bit Memory pretraining (Bit) vs pretrained Vision Transformer (ViT) models. The transformer is frozen.

3. Empirical Evaluation - How important is the transformer architecture compared to LSTM architecture?

- Randomly initialized transformer vs Randomly initialized LSTM
- The authors find that the self-attention architecture already serves as an effective inductive bias for universal computation.

Model	Bit Memory	XOR	ListOps	MNIST	CIFAR-10	C10 LRA	Homology
Trans.	75.8%	100%	34.3%	91.7%	61.7%	36.1%	9.3%
LSTM	50.9%	50.0%	16.8%	70.9%	34.4%	10.4%	6.6%

Table 3: Test accuracy of randomly initialized transformers vs randomly initialized LSTM models. Note that unlike in Figure 1, the LSTM here is frozen. Frozen LSTMs perform very poorly.

3. Empirical Evaluation - Does language pretraining improve compute efficiency over random initialization

Model	Memory	XOR	ListOps	MNIST	C10	C10 LRA	Homology
FPT	1×10^4	5×10^2	2×10^3	5×10^3	4×10^5	3×10^5	1×10^5
Random	4×10^4	2×10^4	6×10^3	2×10^4	4×10^5	6×10^5	1×10^5
Speedup	$4\times$	$40\times$	$3\times$	$4\times$	$1\times$	$2\times$	$1\times$

Table 4: Approximate number of gradient steps until convergence for pretrained (FPT) vs randomly initialized (Random) models. Note that we use the same batch size and learning rate for both models.

3. Empirical Evaluation - Do the frozen attention layers attend to modality-specific tokens?

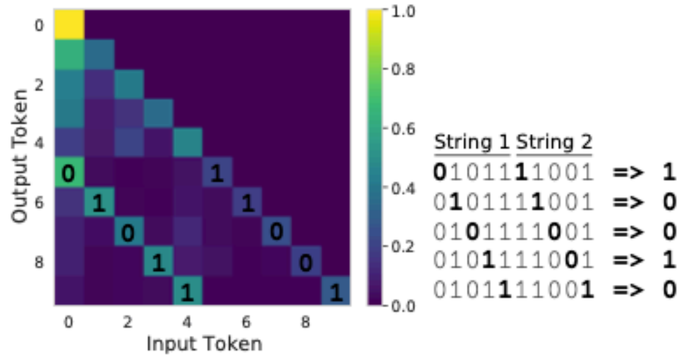

Figure 3: On Bit XOR, the model must produce the element-wise XOR of two bitstrings presented sequentially (inputs 0-4 are the first bitstring, inputs 5-9 are the second). Each token is one bit. FPT learns to attend positionally to the two bits that are XOR'ed by the output token.

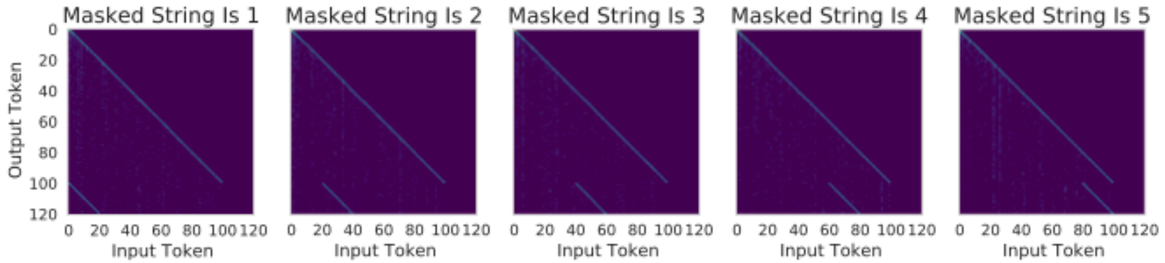

Figure 4: On Bit Memory, the model must return one of five strings (inputs 0-99) given a masked version of one of the strings (inputs 100-119). Each token is 50 bits. FPT learns to attend to the correct string based on finding similarity to the inputs, not relying solely on position as in Bit XOR.

3. Empirical Evaluation - Does performances scale with model size?

Model Size	# Layers	Total Params	Trained Params	FPT	Random
Small (Base)	12	117M	106K	68.2%	61.7%
Medium	24	345M	190K	69.8%	64.0%
Large	36	774M	300K	72.1%	65.7%

Table 6: Test accuracy of larger frozen transformer models on CIFAR-10.

3. Empirical Evaluation - Does fine-tuning the self-attention and feedforward layers further improve performance?

Model	Memory	XOR	ListOps	MNIST	C10	C10 LRA	Homology
FPT	100%	100%	38.4%	98.0%	68.2%	38.6%	12.7%
+ Feedforward	100%	100%	36.0%	98.3%	76.6%	38.2%	13.1%
+ Attention	100%	100%	36.8%	89.0%†	47.7%†	23.0%	10.9%
+ Both	100%	100%	35.8%	93.1%†	32.9%	21.0%	10.5%

Table 8: Additionally finetuning either the feedforward layers, attention layers, or both. We do not use a per-layer learning scheme/etc. †training diverged, number reported before divergence.

Conclusion

- The authors proposed transferring a pertained transformer language model for downstream tasks is non-language modalities.
- The authors believe this work can serve as the foundation for future work investigating transfer between modalities.
- For real-world problem, there are potential upsides with FPT models being able to better exploit representative datasets from one or more modalities.