My experience with Compact Transformers

Mishka (Michael Bukatin)

Dataflow Matrix Machines project

https://github.com/anhinga

My August “Request for Plot”

Replace

$$\text{Attention}(Q, K, V) = \text{softmax}(cKQ^{\mathsf{T}})V$$

with

$$\text{Attention}(Q, K, V) = \text{softmax}(cKQ^{\mathsf{T}})\text{softmax}^{\mathsf{T}}(V)$$

Does your favorite learning curve improve?

This requires Transformer training experiments, and those are supposed to be really expensive.

Compact Transformers

The beauty of requests for plots is that you often get rapid feedback.

And in response to my RFP a friend pointed me to **Compact Transformers**!

Escaping the Big Data Paradigm with Compact Transformers, `https://arxiv.org/abs/2104.05704`

PyTorch original implementation: `https://github.com/SHI-Labs/Compact-Transformers`

Rephrasing the abstract of their paper

Many have come to believe that Transformers are not suitable for small sets of data. [...] In this paper, we dispel the myth that transformers are ”data hungry” and therefore can only be applied to large sets of data. We show for the first time that with the right size and tokenization, transformers can perform head-to-head with state-of-the-art CNNs on small datasets, often with better accuracy and fewer parameters. Our model eliminates the requirement for class token and positional embeddings through a novel sequence pooling strategy and the use of convolutions. It is flexible in terms of model size, and can have as little as 0.28M parameters while achieving good results. Our model can reach 98.00% accuracy when training from scratch on CIFAR-10, which is a significant improvement over previous Transformer based models.

Rephrasing their abstract, continued

It also outperforms many modern CNN based approaches, such as ResNet, and even some recent NAS-based approaches, such as Proxyless-NAS. Our simple and compact design democratizes transformers by making them accessible to those with limited computing resources and/or dealing with small datasets. Our method also works on larger datasets, such as ImageNet (82.71% accuracy with 29% parameters of ViT), and NLP tasks as well. Our code and pre-trained models are publicly available.

My experiments, rather inconclusive

Compact Transformers worked well, took me an hour to do a training run of their default configuration.

I created a fork in order to fix a version of Compact Transformers I was working with:

`https://github.com/anhinga/Compact-Transformers`

Speaking of my RFP, the results are quite inconclusive so far. I've recorded them in the README here:

`https://github.com/anhinga/JuliaCon2021-poster/tree/main/RFP-draft/experiments_with_compact_transformers`

Remix with classical Transformer attention

Various ways to remix, e.g.

$$\alpha * \text{softmax}^{T}(V) + (1 - \alpha) * (V)$$

I originally thought about training Transformers from scratch when creating this RFP.

But if one starts with $\alpha = 0$ and increases α gradually, then one should be able to try fine-tuning a classical pretrained Transformer while deforming it from V towards $\text{softmax}^{T}(V)$.

Contact

Open an issue in

`https://github.com/anhinga/JuliaCon2021-poster`

or send an e-mail to bukatin @ cs dot brandeis dot edu