Speed of Llama

Inar Timiryasov

2023-11-20

BabyLM Challenge

This summer, I participated in the BabyLM Challenge, which aimed to improve the sample efficiency of language models by training them on a small (10M or 100M words), developmentally-plausible dataset.

Eventually, we trained an ensemble consisting of a GPT-2 (705M parameters) and LLaMA (360M parameters) models and then distilled it into a small, 58M-parameter LLaMA model. This model exceeds in performance both of its teachers as well as a similar model trained without distillation. The models were benchmarked using BLiMP, (Super)GLUE, and MSGS tasks. Our BabyLlama scored in the top 5% and is the best decoder model in the competition! Frankly, encoder models, such as Roberta and BERT, were much stronger in BLiMP. Also, the first-place model implemented a very interesting modification of the usual BERT architecture, using a weighted sum of activations of all previous layers as a layer input. Check the paper for details. I am currently exploring a similar approach for a decoder model.

Our code for training HF Transformers models on the BabyLM dataset as well as for distillation pretraining is available here.

Llamas are Fast

During my experiments, I trained many different models from scratch. I found that Llama trains significantly faster than GPT-2. It reaches the minimum eval loss in nearly half the number of epochs needed for GPT-2.

This made me curious: what is the reason? There are two main differences between the models: GPT uses trainable positional embeddings, while Llama employs Rotary Positional Embedding (RoPE). Additionally, Llama utilizes SwiGLU instead of simple MLP layers.

To try to isolate these two effects, I also trained GPT-J, which uses RoPE (although I used the default settings and didn't attempt to make the RoPE implementations match precisely) but not SwiGLU. To make the comparison with GPT-2 more accurate, I enabled weight tying

in both Llama and GPT-J (this feature is disabled by default). I performed a grid search for the optimal learning rate (which happened to be the same for all three models) using the 10M BabyLM dataset (strict-small task). Then, I trained all the models using the 100M dataset (strict task; see the configs *-strict.yaml). The result is shown below.

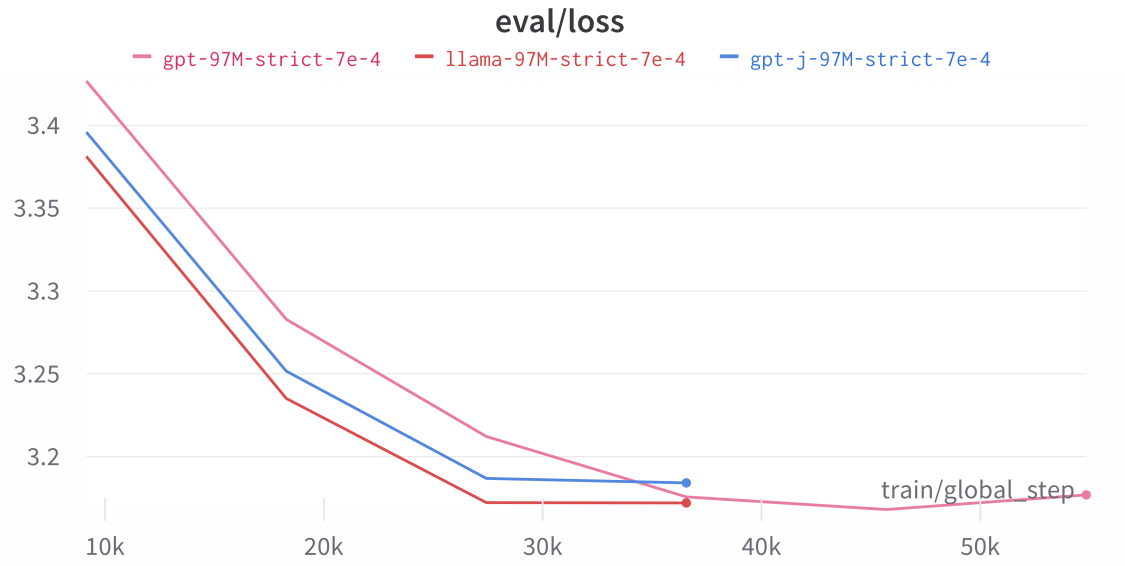

Figure 1: eval-loss

Llama achieves a lower loss than GPT-J and does so more quickly than GPT-2. It seems that SwiGLU—a gated unit that is quadratic in its inputs—is the key to the performance gain.

SwiGLU capacity

Next, I plan to study the capacity of SwiGLU layers. By capacity, I mean—following David MacKay's book—the amount of information a network can store. That is, let's take some random vectors and random binary labels. Then we overfit a network to this data. The capacity is the number of different vectors the model can memorize. This is closely related to Rademacher complexity. What I've found so far is that a SwiGLU network has a larger complexity than a simple MLP.