Tensor Programming V

Denis Sapozhnikov

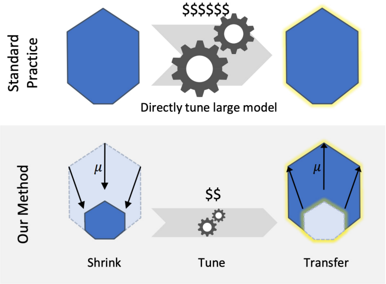
HSE AMI 202

January 15, 2024

Large Models problems

- Speed up of hyperparameter tuning (HP)
- Cost of HP
- Quality of HP

Solution



μP training scheme

Central Limit Theorem

Kind reminder about the CLT formula. If $x_1, x_2, \ldots, x_n$ "look like" sample of random independent variable X, then

$$\frac{1}{\sqrt{n}} \sum_{i=1}^{n} (x_i - \mathbb{E}[X]) \to \mathcal{N}(0, \sigma(X))$$

We will call $\frac{1}{\sqrt{n}}$ the right scaling factor c_n, because with its value the formula yields non-trivial distribution.

Minimization Task

Now define some minimization problem $F_n(c) \to \min$ like following:

$$F_n(c) = \mathbb{E}_{x_1,\ldots,x_n} f\left(c(x_1 + x_2 + \ldots + x_n)\right),$$

where $x_1, x_2, \ldots, x_n$ are hidden variables and f is a bounded function.
Here we can define the right scaling factor $c_n = \frac{\alpha}{\sqrt{n}}$, because we would minimize something non-trivial in infinite-width case:

$$\lim_{n\to\infty} F_n(c_n) \to f(\mathcal{N}(0, \alpha^2)) =: G_n(\alpha) \qquad (1)$$

Ok, and?

- The equation (1) means that for sufficiently large n, the optimal $\alpha_n^* = \arg\min G_n(\alpha)$ should be close to α_N^* for any $N > n$.
- So, applying this to the ideas of machine learning, we can select the scaling factor c_n (learning rate, width, etc.) so that for all larger models the model quality is optimal without HP.

MLP with one hidden layer

Let's define a standard MLP with one hidden layer v of width n, input layer u ($u, v \in \mathbb{R}^n$), and 0 biases for one scalar sample x:

$$f(x) = v^T ux$$

with a standard parametrization (SP): $v_i \sim \mathcal{N}(0, \frac{1}{n})$ and $u_i \sim \mathcal{N}(0, 1)$ and learning rate η.

Why SP is bad

After the first step of SGD, the updated weights will look like $v \leftarrow v + \theta u, u \leftarrow u + \theta v$. From now on the function $f(x)$ is following:

$$f(x) = (v + \theta u)^T (u + \theta v)x = (v^T u + \theta u^T u + \theta v^T v + \theta^2 u^T v)x$$

As you can mention, $u^T u \in \Theta(n)$ which is blown up with the width of the network. Hence, the model's weights will explode in an infinite-width case.

A brand new parametrization

Let's fix our parametrization with a new one (μP):

- $v_i \sim \mathcal{N}(0,1), u_i \sim \mathcal{N}(0, \frac{1}{n^2})$
- $\eta_v = \frac{1}{n}\eta, \eta_u = n\eta$

Then after updating the weights formula will look like this:

$$f(x) = (v^T u + \theta n^{-1} u u^T + \theta n v^T v + \theta^2 u^T v)x$$

Why is it better?

Proof

- $n^{-1}\mathbb{E}[u^T u] = n^{-1} \cdot n \cdot \mathbb{E}u_0^2 = 1 \cdot 1 = 1$
- $n\mathbb{E}[v^T v] = n^2\mathbb{E}v_0^2 = n^2 \cdot \frac{1}{n^2} = 1$
- $\mathbb{E}[u^T v] = n \cdot \mathbb{E}[u_0 v_0] \leq$ [Cauchy–Schwarz] $\leq$
  $\leq n \cdot \sqrt{Var(v_0)}\sqrt{Var(u_0)} = n \cdot 1 \cdot \frac{1}{n} = 1$

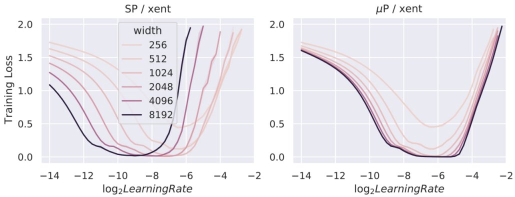
So the weights in $\Theta(1)$, therefore there are no vanishing or exploding of model weights.

How to fix any layer?

	Input weights & all biases	Output weights	Hidden weights
Init. Var.	1/fan_in	$1/\text{fan\_in}^2$ $(1/\textit{fan\_in})$	1/fan_in
SGD LR	fan_out (1)	1/fan_in (1)	1
Adam LR	1	1/fan_in (1)	1/fan_in (1)

But many details about other parameters and special layers (e.g. Transformers) are in the appendix.

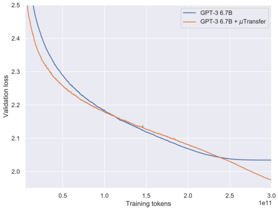
MLP



2-hidden layers MLP

- μP works on "trivial" networks
- Wider is better for any training step
- Unlike SP, optimum has no shift with a rising width

GPT-3



GPT-3 train

- The proxy model is 168 times smaller than the target model by reducing the width by a factor of 16
- The proxy-model was trained only on 4 or 16 billion tokens while the target used the 300 billion
- Tuning cost only 7% of total pretraining cost
- The model exceeded the results of the original work and is comparable to models 2 times larger

Results

- Better performance: μP outperforms SP
- Theory is working on practice with all tested families of models
- Unlike previous works, there is quite a bit family of "broken" layers in scaling rules
- Semi-automatic framework by authors could reduce your pain
- Now more researchers can afford to experiment with large models, and comparing the quality of models will be much easier and more convenient with the same approach to finding hyperparameters

Criticism

- Theory and practice still require a model of a "sufficiently large" size, while there is no understanding of what size is enough
- Based on the plots, the optimal SP models can learn faster than μP
- The optimal HP still shifts slightly for smaller models
- Initialization does not transfer well across depth, and depth transfer generally still does not work for post-layernorm Transformers