Scaling Laws for Transfer

Zhangir Azerbayev

Fall 2023

1 Motivation

The most compute-efficient way to train a language model is to train only on the target distribution and to never repeat data. But suppose we wish to train a language model on a target distribution $\mathcal{F}$ for which there is no more data available than some fixed number of tokens $d_{\mathcal{F}}$. We can try to compensate for our data constraint in three ways:

- Pretrain on some data distrubtion $\mathcal{P}$ for which an arbitrary amount of data is available.
- Train for R epochs on the $d_{\mathcal{F}}$ target distribution tokens.
- Set your number of model parameters N larger than you would if you were allowed arbitrary $D_{\mathcal{F}}$.

How much accuracy and compute-efficiency do we lose by being constrained to $d_{\mathcal{F}}$ target tokens? The following definition helps us quanify this.

Definition 1.1. Suppose an N parameter model pretrained for D_P tokens and finetuned for D_F unique tokens over R epochs achieves a loss of L. The **effective data** D_E of this model is the amount of tokens required for an N parameter model trained only on unrepeated data from the target distribution to achieve a loss of L.

Our goal is to find a parametric form for D_E, i.e $D_E = f\left(N, D_{\mathcal{P}}, D_{\mathcal{F}}, R\right)$. More realistically, given our limited compute budget, we might try to fix a few values of D_F and try to estimate $D_E = f\left(N, D_P, R\right)$ for each of these fixed values. One question we are particularly interested in is for some fixed $D_{\mathcal{F}}$, what is the maximum attainable D_E?

Our scaling law for transfer is therefore $L(N, D_E) = S + \frac{A}{N^{\alpha}} + \frac{B}{D_E^{\beta}}$. Using our scaling law, we can answer questions about compute efficiency. For example, how much more compute is required to train a model to a given loss L when you are restricted to $D_{\mathcal{F}} = d_{\mathcal{F}}$ than if you are allowed arbitrary $D_{\mathcal{F}}$?

2 Estimation

The following is a procedure for estimating $D_E(N, D_{\mathcal{P}}, R)$ for some fixed $D_{\mathcal{F}} = d_{\mathcal{F}}$.

1. First, obtain a Chinchilla-style scaling law $L(N, D_{\mathcal{F}})$.
2. For each N, do a grid sweep over $(D_{\mathcal{P}}, R)$. Each of these runs will yield a loss L, and use the Chinchilla law $L(N, D_{\mathcal{F}}$ to map each loss to a D_E.
3. You now have a set of correspondences $(N, D_{\mathcal{P}}, R) \mapsto D_E$. From this, derive $D_E\left(N, D_{\mathcal{P}}, R\right)$.

3 Related Work

Compute optimal scaling. [HBM+22] introduce three approaches for finding the compute optimal frontier when scaling model parameters and training data. We use their approach three as one step of the estimation procedure in section 2. Approaches one and two are in principle applicable to our transfer learning setting. However, they depend on obtaining a fine grid of samples around the compute optimal frontier. This is simple when your isoFLOP contour is 1-dimensional, but extremely difficult when it is 3-dimensional.

Scaling laws for transfer. [HKHM21] derive a scaling law for transfer learning. However, they only investigate the small data limit, and do not vary the amount of pretraining steps nor do they train for multiple epochs.

References

[HBM+22] Jordan Hoffmann, Sebastian Borgeaud, Arthur Mensch, Elena Buchatskaya, Trevor Cai, Eliza Rutherford, Diego de Las Casas, Lisa Anne Hendricks, Johannes Welbl, Aidan Clark, Tom Hennigan, Eric Noland, Katie Millican, George van den Driessche, Bogdan Damoc, Aurelia Guy, Simon Osindero, Karen Simonyan, Erich Elsen, Jack W. Rae, Oriol Vinyals, and Laurent Sifre. Training compute-optimal large language models, 2022.

[HKHM21] Danny Hernandez, Jared Kaplan, Tom Henighan, and Sam McCandlish. Scaling laws for transfer, 2021.