Simple Derivation of Scaling Law Exponents in Linear Models

Alexander Atanasov

1 Setup

The goal of this note is to summarize the main result of our model of neural scaling laws [3] (with Blake Bordelon and Cengiz Pehlevan) in the setting most of interest to practitioners. We consider a simple supervised task. Take a fixed set of features $\boldsymbol{\psi} \in \mathbb{R}^M$. M should be thought of as infinite, and respresents the space of the infinite-width NTK of a model. The targets are generated by a linear combination of the features plus noise.

$$y = \bar{\boldsymbol{w}} \cdot \boldsymbol{\psi} + \boldsymbol{\epsilon}. \tag{1}$$

For many neural scaling law models, the noise term $\boldsymbol{\epsilon}$ is dropped as we view text and image data as "clean". We will drop this in this note, though there are an interesting class of scaling laws under noise that one can study [2, 7]. The fact that the targets are a linear combination of the infinite-width NTK features comes from the fact that they span the space of functions on input space. The features $\boldsymbol{\psi}$ have a covariance

$$\mathbb{E}[\boldsymbol{\psi}\boldsymbol{\psi}^\top] = \boldsymbol{\Sigma}. \tag{2}$$

In many treatments we will assume that $\boldsymbol{\psi} \sim \mathcal{N}(0, \boldsymbol{\Sigma})$. This assumption seems to hold for how features in the NTK space are distributed. It is known as **gaussian universality** or gaussian equivalence. Wei and Steinhardt [10] show that gaussian equivalence empirically holds even for NTKs of pretrained models.

We take the following conditions on the spectrum λ_k of $\boldsymbol{\Sigma}$ and the decay of $\bar{\boldsymbol{w}}$. Writing $\bar{w}_k$ as the signal in the kth eigenmode:

$$\lambda_k \sim k^{-\alpha}, \quad \sum_{\ell > k} \lambda_k \bar{\boldsymbol{w}}_\ell^2 \sim k^{-\alpha\beta}. \tag{3}$$

The exponents α, β are called **capacity** and **source** respectively. They control how the PCA of the data decays in the feature space of the NTK, as well as how the target weights $\bar{\boldsymbol{w}}$ decay along those eigenspaces.

One can observe these power laws empirically for models on real datasets. To measure the eigenspectrum of $\boldsymbol{\Sigma}$, one can study the kernel gram matrix $K_{\mu\nu} = \boldsymbol{\psi}(\boldsymbol{x}_\mu) \cdot \boldsymbol{\psi}(\boldsymbol{x}_\nu)$ where μ, ν run over a large dataset of point. This is computable without explicitly using the infinite-dimensional feature space. We then observe power law decay in the eigenvalues of K, which are the same as the eigenvalues of $\boldsymbol{\Sigma}$. Similarly, to measure β, one can also project the labels y_μ onto the first k principal components of K and study how the remaining variance decays.

For real datasets, we observe small values for β, always < 1, and usually around 0.05. Smaller β corresponds to "harder" tasks. Usually α is measured to be slightly above 1. If $\beta > 1$ (i.e. very easy tasks) other novel scaling laws can emerge, but I do not think they are relevant for practitioners.

2 Studying The Model

2.1 Population Gradient Flow: Finite Time Effects

We are interested in the MSE test loss:

$$\mathcal{L} = \frac{1}{2}\mathbb{E}_{\boldsymbol{x}}[(\boldsymbol{\psi}(\boldsymbol{x}) \cdot \bar{\boldsymbol{w}} - \boldsymbol{\psi}(\boldsymbol{x}) \cdot \hat{\boldsymbol{w}})^2] = \frac{1}{2}(\bar{\boldsymbol{w}} - \hat{\boldsymbol{w}})^\top \boldsymbol{\Sigma} (\bar{\boldsymbol{w}} - \hat{\boldsymbol{w}}) = \frac{1}{2}\sum_k \lambda_k (\bar{w}_k - \hat{w}_k)^2. \tag{4}$$

For a linear model under population gradient flow, the learned weights $\hat{\boldsymbol{w}}$ satisfy the differential equation

$$\frac{d}{dt}\hat{\boldsymbol{w}} = -\nabla_{\boldsymbol{w}} \mathcal{L} = -\boldsymbol{\Sigma}(\bar{\boldsymbol{w}} - \hat{\boldsymbol{w}}) \Rightarrow \hat{w}_k = (1 - e^{-t\lambda_k})\bar{w}_k. \tag{5}$$

The time-scale to learn mode k is therefore $t \sim 1/\lambda_k \sim k^\alpha$. This means that at time t, all the modes below k^α are learned, and all the modes above k^α are not learned. From the definition of the test loss, we get

$$\mathcal{L} = \frac{1}{2}\sum_k \lambda_k (\bar{w}_k - \hat{w}_k)^2 \approx \frac{1}{2}\sum_{\ell > t^{1/\alpha}} \lambda_k \bar{w}_k^2 \sim t^{-\beta}. \tag{6}$$

This is the first, and most basic scaling law. Several authors have derived it [5, 6, 9].

2.2 Finite Model Size

We now consider training a linear model that *does not have access to the full set of features*, but rather a projection of them. We take $\boldsymbol{A} \in \mathbb{R}^{M\times N}$ to project from the (infinite) M-dimensional space to the (finite, but still large) N-dimensional space. The analogy is that the infinite width NTK is replaced the (finite-width) empirical NTK. The learned function is then:

$$\hat{y} = \boldsymbol{\psi}^\top \boldsymbol{A}\boldsymbol{v}. \tag{7}$$

Here, $\boldsymbol{v}$ is trained with gradient flow on the population loss

$$\frac{d}{dt}\boldsymbol{v} = -\frac{1}{2}\nabla_{\boldsymbol{v}}\mathbb{E}_{\boldsymbol{x}}|\boldsymbol{\psi}(\boldsymbol{x})\cdot\bar{\boldsymbol{w}} - \boldsymbol{\psi}(\boldsymbol{x})^\top\boldsymbol{A}\boldsymbol{v}|^2 = \boldsymbol{A}^\top\boldsymbol{\Sigma}(\bar{\boldsymbol{w}} - \boldsymbol{A}\boldsymbol{v}). \tag{8}$$

The exact structure of $\boldsymbol{A}$ is not precisely known for real models, but the predictions are pretty universal as long as it has the form of a random projection. Here, we will assume the elements of $\boldsymbol{A}$ are i.i.d. gaussian distributed with mean zero and variance $1/N$. Define $\hat{\boldsymbol{w}} = \boldsymbol{A}\boldsymbol{v}$. The equation for SGD then becomes:

$$\frac{d}{dt}\hat{\boldsymbol{w}} = -\boldsymbol{A}\boldsymbol{A}^\top\boldsymbol{\Sigma}(\bar{\boldsymbol{w}} - \hat{\boldsymbol{w}}) \Rightarrow \hat{\boldsymbol{w}} = (1 - e^{-t\boldsymbol{A}\boldsymbol{A}^\top\boldsymbol{\Sigma}})\bar{\boldsymbol{w}}. \tag{9}$$

The key observation is that $\boldsymbol{A}\boldsymbol{A}^\top$ is an $M \times M$ matrix of rank N. This means that only N eigenvalues can be learned. One can use random matrix theory to precisely calculate the average case behavior of these dynamics and show that this is also the typical case. Recall the Nth eigenvalue of $\boldsymbol{\Sigma}$ goes as $N^{-\alpha}$, and this is the cutoff past which the model cannot resolve more of the spectrum. So for $t < 1/\lambda_N \sim N^\alpha$, we would observe $\mathcal{L} \sim t^{-\beta}$ as before. However, once $t > N^\alpha$, we observe a plateau to a value of

$$\sum_{\ell > N} \lambda_k \bar{w}_k^2 \sim N^{-\alpha\beta}. \tag{10}$$

The Chinchilla law for the loss would then be:

$$\mathcal{L} \sim t^{-\beta} + N^{-\alpha\beta}. \tag{11}$$

Maximizing $\mathcal{L}$ at constant compute $C = Nt$ yields the law:

$$\mathcal{L} \sim C^{\frac{\alpha\beta}{\alpha+1}}, \quad t \sim C^{\frac{\alpha}{\alpha+1}}, N \sim C^{\frac{1}{\alpha+1}}. \tag{12}$$

2.3 Finite Dataset Size

One can extend this model to also include only a finite training set of size P which is then repeated. Instead of having the population covariance $\boldsymbol{\Sigma}$ in the gradient flow equations, it is replaced by the empirically estimated covariance:

$$\hat{\boldsymbol{\Sigma}} \equiv \frac{1}{P}\sum_{\boldsymbol{x}\in\mathcal{D}} \boldsymbol{\psi}(\boldsymbol{x})\boldsymbol{\psi}(\boldsymbol{x})^\top. \tag{13}$$

The gradient flow equation is then modified to:

$$\frac{d}{dt}\hat{\boldsymbol{w}} = -\boldsymbol{A}\boldsymbol{A}^\top\hat{\boldsymbol{\Sigma}}(\bar{\boldsymbol{w}} - \hat{\boldsymbol{w}}) \Rightarrow \hat{\boldsymbol{w}} = (1 - e^{-t\boldsymbol{A}\boldsymbol{A}^\top\hat{\boldsymbol{\Sigma}}})\bar{\boldsymbol{w}}. \tag{14}$$

Now $\boldsymbol{A}\boldsymbol{A}^\top$ is rank N while $\hat{\boldsymbol{\Sigma}}$ is rank P. Whichever of N, P is smaller will determine the bottleneck on the spectrum. We get the following extension to the Chinchilla law:

$$\mathcal{L} \sim t^{-\beta} + N^{-\alpha\beta} + P^{-\alpha\beta}. \tag{15}$$

2.4 Including SGD Effects

Several works also extend this model to SGD rather than gradient flow [1, 4, 8]. For hard tasks, namely when $\beta < 1$, this doesn't affect the scaling laws.

3 Going Beyond Linear Models

In [4], we study a simple toy model of a feature learning network and obtain improved scaling law exponents when $\beta < 1$ (i.e. for "hard" tasks). The Chinchilla laws are then modified by replacing $\beta \to 2\beta/(1+\beta)$, yielding an almost doubling in the exponent at small β. It is not clear how universal these predictions are, and future work remains to be done studying all the effects that feature learning can have on improving neural scaling laws.

References

[1] A. Atanasov, B. Bordelon, J. A. Zavatone-Veth, C. Paquette, and C. Pehlevan, "Two-point deterministic equivalence for stochastic gradient dynamics in linear models", arXiv preprint arXiv:2502.05074 (2025).

[2] A. Atanasov, J. A. Zavatone-Veth, and C. Pehlevan, "Scaling and renormalization in high-dimensional regression", arXiv preprint arXiv:2405.00592 (2024).

[3] B. Bordelon, A. Atanasov, and C. Pehlevan, "A dynamical model of neural scaling laws", arXiv preprint arXiv:2402.01092 (2024).

[4] B. Bordelon, A. Atanasov, and C. Pehlevan, "How feature learning can improve neural scaling laws", arXiv preprint arXiv:2409.17858 (2024).

[5] B. Bordelon and C. Pehlevan, "Learning curves for sgd on structured features", arXiv preprint arXiv:2106.02713 (2021).

[6] A. Caponnetto and E. De Vito, "Optimal rates for the regularized least-squares algorithm", Foundations of Computational Mathematics **7**, 331–368 (2007).

[7] L. Defilippis, B. Loureiro, and T. Misiakiewicz, "Dimension-free deterministic equivalents and scaling laws for random feature regression", arXiv preprint arXiv:2405.15699 (2024).

[8] E. Paquette, C. Paquette, L. Xiao, and J. Pennington, "4+ 3 phases of compute-optimal neural scaling laws", arXiv preprint arXiv:2405.15074 (2024).

[9] L. Pillaud-Vivien, A. Rudi, and F. Bach, "Statistical optimality of stochastic gradient descent on hard learning problems through multiple passes", Advances in Neural Information Processing Systems **31** (2018).

[10] A. Wei, W. Hu, and J. Steinhardt, "More than a toy: random matrix models predict how real-world neural representations generalize", in International conference on machine learning (PMLR, 2022), pp. 23549–23588.