Implicit Regularization in Deep Learning: The Kernel and Rich Regimes

Henry Smith
Advised by Professor Harrison Zhou
Yale University

Motivation

Over the past couple of decades, neural networks have risen through the ranks to replace random forests and SVMs as the premier models for prediction. In large part, this proliferation in deep learning models has been due to their good generalization properties. From a theoretical perspective, though, little is known as to why neural networks generalize so well. In our research, we examine two limits present in neural network training, the "kernel" and "rich" limits. Principally, we are interested in what these limits tell us about the generalization properties of neural networks.

Introduction to the Kernel and Rich Limits

Problem Setup

Consider a differentiable model $h : \boldsymbol{w} \mapsto f(\boldsymbol{w}, \boldsymbol{x})$ which maps each weight vector $\boldsymbol{w} \in \mathbb{R}^p$ to a neural network function $f(\boldsymbol{w}, \cdot) : \mathbb{R}^n \to \mathbb{R}$. Assume that the model h is D-positive homogeneous, meaning $h(\alpha \boldsymbol{w}) = \alpha^D h(\boldsymbol{w})$ for each $\alpha > 0$. Also, suppose that we have some differentiable loss function L which measures the misfit of each neural network function on our training data $\{(\boldsymbol{x}_i, y_i)\}_{i=1}^N$.

We would like to understand the **gradient flow** $(\boldsymbol{w}_\alpha(t))_{t \geq 0}$ on the objective $\frac{1}{\alpha^2} L(\alpha h(\boldsymbol{w}))$:

$$\boldsymbol{w}_\alpha'(t) = -\nabla \left(\frac{1}{\alpha^2} L(h(\boldsymbol{w}_\alpha(t))) \right) = -\frac{1}{\alpha^2} Dh(\boldsymbol{w}_\alpha(t))^T \nabla L(h(\boldsymbol{w}_\alpha(t))), \quad \boldsymbol{w}_\alpha(0) = \alpha^{1/D} \boldsymbol{w}_0.$$

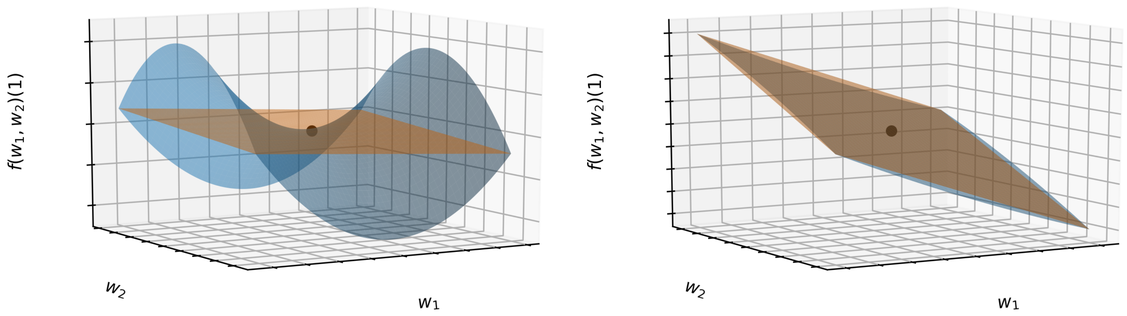
From the practitioner's perspective, gradient flow is no more than a continuous time analog of gradient descent.

Scaling Up

In particular, we are interested in what happens to the gradient flow solution $\boldsymbol{w}_\alpha^\star = \lim_{t \to \infty} \boldsymbol{w}_\alpha(t)$ when the scale α of the initialization $\boldsymbol{w}_\alpha(0) = \alpha^{1/D} \boldsymbol{w}_0$ grows arbitrarily large. Equivalently, what happens to the gradient flow when we initialize the weights our neural network to have larger and larger norm?

The work of Chizat and colleagues demonstrates that in this limit $\alpha \to \infty$, training the model h is equivalent to training the linearization of h around its initialization $\boldsymbol{w}_\alpha(0)$ [1]. That is, training h, which may be highly nonconvex as a function of its weights $\boldsymbol{w} \in \mathbb{R}^p$, is simplified to training an affine model. This $\alpha \to \infty$ limit of network training is called the **kernel limit**.

Conversely, when the scale of the initialization grows very small $\alpha \to 0$, training the model h is very different from training the linearization of h around its initialization. This is called the **rich limit** of network training.



(a) $\alpha = 10^{-1}$ (b) $\alpha = 10$

Figure 1. We plot a linear model $f(w_1, w_2, x) = (w_1^2 - w_2^2)x$ (blue) as a function of its weights $(w_1, w_2) \in \mathbb{R}^2$ at the fixed input $x = 1$. In addition, we visualize the linearization of the model $f(w_1, w_2, 1)$ around its initialization $\boldsymbol{w}_\alpha(0) = \alpha \mathbb{1}$, $\bar{f}(w_1, w_2, 1) = f(\boldsymbol{w}_\alpha(0)) + D_{(w_1,w_2)} f(\boldsymbol{w}_\alpha(0), \boldsymbol{x})$ $\cdot((w_1, w_2) - \boldsymbol{w}_\alpha(0))$ (orange). Notice that for (a), $f(w_1, w_2, 1)$ is very different from $\bar{f}$ about $\boldsymbol{w}_\alpha(0)$, whereas for (b), $f(w_1, w_2, 1)$ is very close to $\bar{f}$.

Connection with Implicit Regularization

In [3], Woodworth and colleagues consider the linear regression model $f(\boldsymbol{w}, \boldsymbol{x}) = \langle \boldsymbol{\beta}_{\boldsymbol{w}}, \boldsymbol{x} \rangle$ with coefficient vector $\boldsymbol{\beta}_{\boldsymbol{w}} = \boldsymbol{w}_+^2 - \boldsymbol{w}_-^2$. Here, the weights of our neural network are $\boldsymbol{w} = [\boldsymbol{w}_+, \boldsymbol{w}_-]^T \in \mathbb{R}^{2n}$, and $\boldsymbol{v}^2$ denotes element-wise squaring of $\boldsymbol{v} \in \mathbb{R}^n$. They suppose that the loss function is the empirical risk $L(f(\boldsymbol{w}, \cdot)) = \sum_{i=1}^N [y_i - f(\boldsymbol{w}, \boldsymbol{x}_i)]^2$, where the training data $\{(\boldsymbol{x}_i, y_i)\}_{i=1}^N$ satisfies $N \ll n$. More precisely, they assume that there are many solutions to the system $\boldsymbol{X}\boldsymbol{\beta} = \boldsymbol{y}$.

For this particular linear regression problem, the gradient flow solution in the kernel limit $\alpha \to \infty$ corresponds to the minimum ℓ^2 solution of the system $\boldsymbol{X}\boldsymbol{\beta} = \boldsymbol{y}$, whereas the solution in the rich limit $\alpha \to 0$ corresponds to the minimum ℓ^1 solution [3]. This result suggests the immense benefit of training near the rich limit for problems in which the input data $\boldsymbol{x} \in \mathbb{R}^n$ is very high-dimensional but is suspected to have implicit sparsity.

A New Example: The Logistic Regression Problem

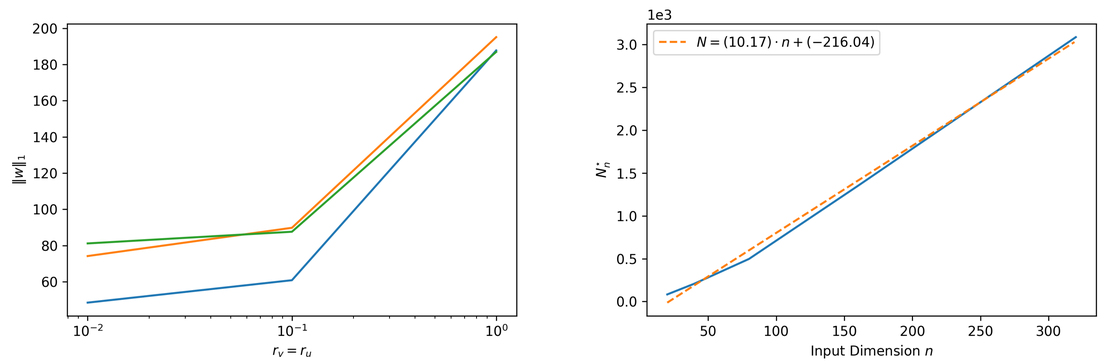


Figure 2. (left) The ℓ^1 norm of the gradient descent solution vector, where the gradient descent path is initialized with scale $r_v = r_{\boldsymbol{u}}$. Since the starting point of gradient descent is random, we average over $K = 20$ trials. (right) The smallest number of training samples necessary to attain test error < 0.4 for input dimension n. For this experiment, the initialization scale is $r_v = r_{\boldsymbol{u}} = 10^{-1}$, and so training is performed close to the rich limit.

To investigate whether or not training near the rich limit results in implicit ℓ^1 regularization for more advanced problems, we consider the sparse logistic regression problem posed by Wei and colleagues [2]. Specifically, we suppose that the data $(\boldsymbol{x}, y) \sim \rho$ is generated such that only the first two coordinates of $\boldsymbol{x} \in \{+1, 0, -1\}^n$ determine $y \in \{+1, -1\}$. For our model, we choose a ReLU neural network with a single hidden layer containing m hidden units: $f(\boldsymbol{w}, \boldsymbol{x}) = \sum_{i=1}^m v_i \cdot \max\{0, \boldsymbol{u}_i^T \boldsymbol{x}\}$. The weights $\{(v_i, \boldsymbol{u}_i)\}_{i=1}^m$ of the network are initialized according to $v_i \sim \mathcal{N}(0, r_v^2)$, $\boldsymbol{u}_i \sim \mathcal{N}(\boldsymbol{0}, r_{\boldsymbol{u}}^2 \mathbb{I})$.

Our experiments suggest that the rich limit $r_v, r_{\boldsymbol{u}} \to 0$, does, in fact, impose implicit ℓ^1 regularization on the gradient descent solution. Even more remarkable, our empirical results provide evidence that the generalization error of the ReLU network trained in the rich limit is bounded by $\mathbb{P}_{(\boldsymbol{x},y)\sim\rho}(f(\boldsymbol{w}, \boldsymbol{x})y \leq 0) \lesssim \sqrt{\frac{n}{N}}$. This is the same [asymptotic] bound on the generalization error that Wei and colleagues achieved by including an explicit ℓ^2 regularizer in the loss function.

That is, by simply decreasing the scale with which we initialize the network weights, we have achieved commensurate performance to when we use an explicit ℓ^2 regularizer. This result points to the immense role of implicit regularization in neural network generalization, and it suggests the necessity of further research into the kernel and rich limits.

References

[1] Lenaic Chizat, Edouard Oyallon, and Francis Bach. On lazy training in differentiable programming. *arXiv preprint arXiv:1812.07956*, 2018.

[2] Colin Wei, Jason Lee, Qiang Liu, and Tengyu Ma. Regularization matters: Generalization and optimization of neural nets vs their induced kernel. 2019.

[3] Blake Woodworth, Suriya Gunasekar, Jason D Lee, Edward Moroshko, Pedro Savarese, Itay Golan, Daniel Soudry, and Nathan Srebro. Kernel and rich regimes in overparametrized models. In *Conference on Learning Theory*, pages 3635–3673. PMLR, 2020.

Department of Statistics & Data Science henry.smith@yale.edu