A Literature Survey on the Behavior of Overparameterized Neural Networks

Kamaljeet Singh
Statistics & Data Science
University of Arizona

Abstract

The success of neural networks has gained significant attention in recent years, yet their theoretical guarantees are only partially understood. Deeper neural networks have become the preferred choice for many applications. In most deep learning models, the number of parameters often exceeds the number of training observations, leading to the phenomenon of overparameterization. This regime gives rise to intriguing behaviors, including double descent Belkin et al. [2019], bengin overfitting, etc. which suggests that sufficiently overparameterized neural networks can perfectly fit the training data while still generalizing well to unseen data, seemingly contradicting the classical bias-variance trade-off.

Several research efforts have attempted to explain this behavior. Some attribute it to implicit regularization, which arises naturally from the optimization process rather than explicit constraints. Recent work suggests that the optimization algorithm itself, particularly gradient-based methods, plays a crucial role in this implicit regularization Barrett and Dherin [2022], Smith et al. [2021], Neyshabur [2017], Wang et al. [2021]. Belkin et al. [2018] Belkin [2021] examines the role of interpolation and over-parameterization in modern machine learning, highlighting their impact on generalization and optimization. In this project, we conducted a literature survey on recent observations in the overparameterized regime and provided a brief overview and explanation of the underlying reasons behind them.

1 Introduction

In supervised learning, our goal is to learn a function $f : R^d \to R^K$ from a hypothesis class $\mathcal{H}$, using a dataset $\{(x_i, y_i)\}_{i=1}^N$, where $x_i \in R^d$ and $y_i \in R^K$. Here, $K = 1$ for regression and binary classification, and $K \geq 2$ for multiclass classification. We assume the data points (x_i, y_i) are independent and identically distributed (i.i.d.) samples drawn from a distribution P. For simplicity, this report focuses on regression and binary classification tasks ($K = 1$).

The most common approach to train such models is Empirical Risk Minimization (ERM), which minimizes a loss function defined as:

$$L = \frac{1}{N} \sum_{i=1}^{N} \ell(f(x_i), y_i),$$

where ℓ is typically the squared error loss or cross-entropy loss. However, the ultimate aim of machine learning is not just optimization but *learning*—that is, finding a function f_N that minimizes the true risk:

$$E_{(x,y)\sim P} \left[\ell(f_N(x), y)\right].$$

In other words, we want a function that performs well on unseen data, achieving good generalization.

Classical statistical theory suggests that the capacity of the hypothesis class $\mathcal{H}$ determines whether a model underfits or overfits, a concept captured by the bias-variance trade-off, as shown in Figure 1(a). High-capacity models, like deep neural networks, have become popular due to their flexibility and ability to handle diverse data types. However, these models often defy classical theory. Belkin et al. [2019] introduced the double descent phenomenon (see Figure 1(b)), which shows that as model capacity increases beyond a certain point, test error can decrease again, challenging traditional views on overfitting.

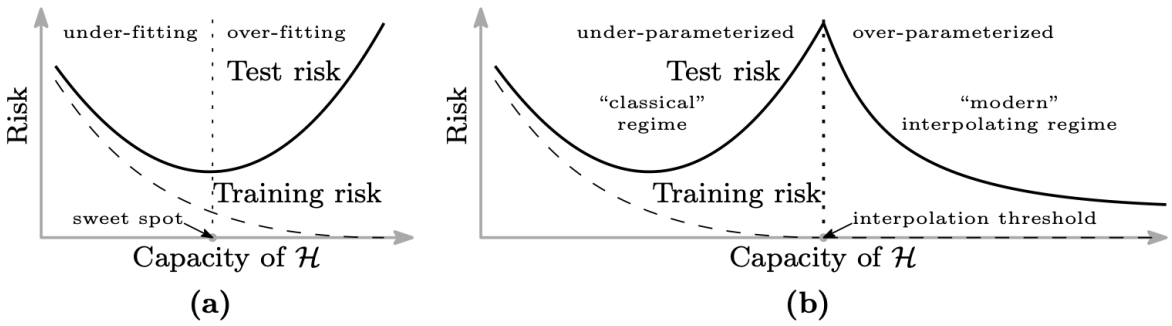

Figure 1: (a) Classical U-curve from the bias-variance trade-off. (b) Double descent curve arising from overparameterization. Belkin et al. [2019]

Surprisingly, over-parameterized models like neural networks, which can fit the training data perfectly (achieving zero training error), often generalize well to unseen data. This contradicts the classical belief that zero training error leads to poor generalization. Recent studies show that in over-parameterized regimes, where models interpolate the data, neural networks and kernel methods can achieve near-optimal test performance even when fitting noisy training data. One explanation credits this to the maximum margin: the large capacity of $\mathcal{H}$ allows the model to learn a function with a large margin, which often generalizes better.

This raises an intriguing question: in the modern over-parameterized regime, where models can perfectly fit the training data, there exist many solutions (interpolators) that achieve zero training error. What makes some of these solutions generalize better than others, and how does the training algorithm select the better ones? The literature suggests that this is due to *inductive bias*, which guides algorithms like SGD to favor certain solutions. Controlling this bias can be explicit, such as choosing a specific neural network architecture, or implicit, through regularization techniques like early stopping. Understanding these mechanisms is key to explaining why over-parameterized models succeed where classical theory predicts they should fail. In the following sections, we discuss possible reasons for the good generalization performance of overparameterized networks, along with some other related observations.

2 Implicit Regularization

In one of the early explanations of regularization or inductive bias in the overparameterized regime, Barrett and Dherin [2022] introduce the term Implicit Gradient Regularization (IGR). The idea is that gradient descent implicitly biases the optimization trajectory toward regions of the loss landscape with flatter minima—regions where models tend to generalize better and are more robust to small perturbations in the parameters. The authors use backward error analysis to derive the form of this implicit regularization. The core argument revolves around the discrepancy introduced when using discrete-time optimization methods like gradient descent to approximate the continuous-time gradient flow. However, since this kind of regularization is observed primarily in the overparameterized regime, we can't claim that it's solely due to the discretization of gradient flow. If discretization were the only reason, we would expect similar effects in underparameterized models as well—which we generally don't see. So, implicit gradient regularization is likely one important piece of a broader set of mechanisms driving generalization in overparameterized models. Gradient flow describes the continuous-time limit of gradient descent, where parameters evolve smoothly over time to minimize a loss function. It models optimization as a differential equation, capturing the trajectory of parameters under the influence of the gradient:

$$\dot{\theta} = -\nabla L(\theta) \tag{1}$$

Here, $\dot{\theta}$ denotes the time derivative of the parameter vector θ, and $\nabla L(\theta)$ is the gradient of the loss function L.

Gradient descent updates parameters to minimize a loss function $L(\theta)$ using the update rule:

$$\theta_{t+1} = \theta_t - \epsilon \, \nabla_\theta L(\theta_t),$$

where ϵ is the learning rate.

However, gradient descent does not exactly follow the continuous gradient flow path. Instead, it approximately follows (or stay close to) a modified gradient flow:

$$\dot{\theta} = -\nabla_\theta \widetilde{L}(\theta) \tag{2}$$

where $\widetilde{L}(\theta)$ is a modified loss defined as:

$$\widetilde{L}(\theta) = L(\theta) + \lambda R_{IG}(\theta) \tag{3}$$

with regularization rate

$$\lambda = \frac{\epsilon * p}{4},$$

where p is the number of parameters and implicit gradient regularization term

$$R_{IG}(\theta) = \frac{1}{p} \sum_{i=1}^{p} \left(\nabla_{\theta_i} L(\theta)\right)^2 .$$

This regularization $R_{IG}(\theta)$ penalizes regions with large gradients, encouraging flatter minima and thus implicitly improving generalization, even though it is not explicitly added to the loss function.

We directly quote the following results and Theorem 1 from Barrett and Dherin [2022]:

Implicit Gradient Regularization (IGR) has the following effects:

- "IGR encourages smaller values of $R_{\mathrm{IG}}(\theta)$ relative to the loss $L(\theta)$".
- "IGR encourages the discovery of flatter optima".
- "IGR encourages higher test accuracy".
- "IGR encourages the discovery of optima that are more robust to parameter perturbations".

Theorem 2.1. *Let L be a sufficiently differentiable function on a parameter space $\theta \in R^p$. The modified equation for gradient flow (Equation 1) is of the form*

$$\dot{\theta} = -\nabla \tilde{L}(\theta) + \mathcal{O}(h^2), \tag{7}$$

where $\tilde{L} = L + \mathcal{R}_{IG}$ is the modified loss introduced in Equation 3. Consider gradient flow with the modified loss $\dot{\theta} = -\nabla \tilde{L}(\theta)$ and its solution $\tilde{\theta}(t)$ starting at θ_{n-1}. Now the local error $\|\theta_n - \tilde{\theta}(h)\|$ between $\tilde{\theta}(h)$ and one step of gradient descent $\theta_n = \theta_{n-1} - h\nabla L(\theta_{n-1})$ is of order $\mathcal{O}(h^3)$, while it is of order $\mathcal{O}(h^2)$ for gradient flow with the original loss.

This theorem reinterprets gradient descent as a numerical method solving the ODE $\dot{\theta} = -\nabla \tilde{L}(\theta)$ (Equation 2). The modified equation $\dot{\theta} = -\nabla \tilde{L}(\theta) + \mathcal{O}(h^2)$ introduces a modified loss $\tilde{L} = L + \mathcal{R}_{IG}$, where $\mathcal{R}_{IG}$ is an implicit regularization term. The solution $\tilde{\theta}(t)$ of this modified flow is compared to a gradient descent step $\theta_n = \theta_{n-1} - h\nabla L(\theta_{n-1})$. The local error $\|\theta_n - \tilde{\theta}(h)\|$ is $\mathcal{O}(h^3)$, meaning gradient descent closely follows the modified flow, improving accuracy over the $\mathcal{O}(h^2)$ error of the original flow. As previously mentioned, IGR is just one component contributing to the generalization ability of overparameterized neural networks. It also depends on factors such as model architecture and parameter initialization.

A clear observation is that the IGR rate λ is chosen to be proportional to the number of parameters m, and experiments show that it controls test accuracy. Moreover, IGR and the original loss function share the same global minima because the regularization term vanishes when the gradient of the loss is zero. Therefore, IGR influences the learning trajectory, potentially leading to different final solutions in overparameterized models, though it does not assist in escaping local minima.

This analysis is based on full-batch gradient descent. However, in stochastic gradient descent (SGD), an additional source of randomness arises from mini-batch selection. We explore the role of implicit regularization under SGD in the next section.

3 Implicit Regularization in Stochastic Gradient Descent

In practice, when training a neural network, we rarely use the full dataset to compute the gradient at each step. Instead, a common approach is to use a randomly selected subset of the data, known as a mini-batch, to update the parameters. This variant of gradient descent is called *Stochastic Gradient Descent* (SGD). The update rule for SGD is given by:

$$\theta_{t+1} = \theta_t - \epsilon \nabla_\theta L_{\mathcal{B}_t}(\theta_t), \tag{4}$$

where ϵ is the learning rate, and $\nabla_\theta L_{\mathcal{B}_t}(\theta_t)$ is the gradient of the loss computed over the mini-batch $\mathcal{B}_t$ sampled from the training data at iteration t.

In SGD, an additional source of randomness arises from two factors: (1) the specific data selected in each mini-batch, and (2) the order in which the batches are processed during training (i.e., within each epoch). To simplify the analysis, the first factor is held constant, and the stochasticity is assumed to arise only from the order in which batches are observed during training.

Using similiar backward error analyisis apporach as in Barrett and Dherin [2022] Smith et al. [2021] explore implicit regularization in Stochastic Gradient Descent (SGD), particularly when using finite learning rates and random shuffling and demonstrate that SGD follows the gradient flow of a modified loss, which combines the original loss with an implicit regularizer penalizing the norms of minibatch gradients. When the batch size is small, the regularization strength scales with the ratio of the learning rate to the batch size. This implicit regularization biases SGD towards flatter minima, enhancing generalization by reducing minibatch gradient variance. Smith et al. [2021] further analyze implicit regularization in SGD by studying its mean iterate over random shuffles. For a full-batch loss $L(\theta) = \frac{1}{N}\sum_{j=1}^{N} L_j(\theta)$, SGD with batch size B, $m = N/B$ minibatches, and learning rate ϵ follows the gradient flow of a modified loss after one epoch:

$$\tilde{L}_{\text{SGD}}(\theta) = L(\theta) + \frac{\epsilon}{4m}\sum_{k=0}^{m-1} \|\nabla \hat{L}_k(\theta)\|^2,$$

$$\tilde{L}_{\text{SGD}}(\theta) = L(\theta) + \frac{\epsilon}{4}\|\nabla L(\theta)\|^2 + \frac{\epsilon}{4m}\sum_{i=0}^{m-1} \|\nabla \hat{L}_i(\theta) - \nabla L(\theta)\|^2. \tag{5}$$

where $\hat{L}_k(\theta) = \frac{1}{B}\sum_{j=kB+1}^{(k+1)B} L_j(\theta)$ is the minibatch loss. The regularizer penalizes the mean squared norm of minibatch gradients. In contrast, GD's modified loss is (the first two terms in 5):

$$\tilde{L}_{\text{GD}}(\theta) = L(\theta) + \frac{\epsilon}{4}\|\nabla L(\theta)\|^2.$$

SGD's regularizer additionally penalizes non-uniformity in minibatch gradients, biasing it towards flatter minima with better generalization.

4 Batch Size Saturation in Over-Parameterized Models

While there has been intense studies about models with just the right number of parameters (under-parameterized models), new research shows that models with extra parameters (over-parameterized models) have some big advantages. Batch Saturation is one of them. The claim is that in overeparam-eterzied models, we can enjoy the benefits (upto a multiplicative constant) of full bacth gradient by only using a small batch. In smaller models, using a batch of size m (a group of m data points) takes about the same effort as doing m updates with just one data point at a time. But in bigger models, this changes. There's a special batch size, m^*, where this pattern stops. After m^*, making the batch bigger doesn't help much—it gives less and less benefit. So, using a medium-sized batch with SGD can work almost as well as using the whole dataset at once, like in full Gradient Descent. Ma et al. [2018]. This batch size saturation phenomenon is expressed in 2.

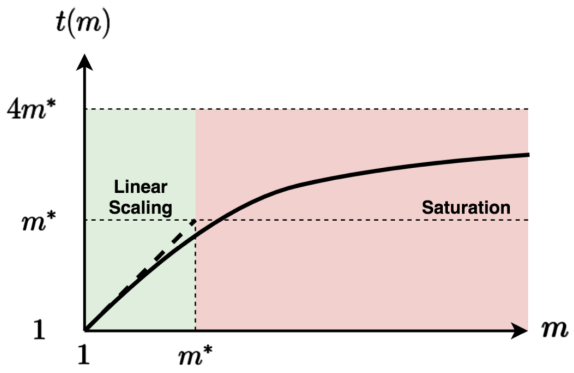

Figure 1: $t(m)$ iterations with batch size 1 (the y axis) equivalent to one iteration with batch size m (the x axis) for convergence.

Figure 2:

5 Conclusion

In this literature survey, we identified several theoretical gaps that have emerged in the modern machine learning. Despite these gaps, the interpolation regime offers some notable advantages, which we have discussed throughout the paper. We summarize them below:

1. SGD converges faster in the overparameterized regime and becomes nearly equivalent to full gradient descent when the batch size exceeds a certain threshold.

2. The use of gradient-based optimization introduces implicit regularization, which becomes even more pronounced in the case of SGD. This implicit bias guides the model toward solutions that generalize better, even when the training data is overfit.

However, the use of large models also introduces challenges, particularly in terms of computational cost during training. These issues, nevertheless, can be addressed using techniques discussed in the literature, such as in Livni et al. [2014].

References

David G. T. Barrett and Benoit Dherin. Implicit gradient regularization, 2022. URL https://arxiv.org/abs/2009.11162.

Mikhail Belkin. Fit without fear: remarkable mathematical phenomena of deep learning through the prism of interpolation, 2021. URL https://arxiv.org/abs/2105.14368.

Mikhail Belkin, Daniel Hsu, and Partha Mitra. Overfitting or perfect fitting? risk bounds for classification and regression rules that interpolate, 2018. URL https://arxiv.org/abs/1806.05161.

Mikhail Belkin, Daniel Hsu, Siyuan Ma, and Soumik Mandal. Reconciling modern machine-learning practice and the classical bias–variance trade-off. *Proceedings of the National Academy of Sciences*, 116(32):15849–15854, July 2019. ISSN 1091-6490. doi: 10.1073/pnas.1903070116. URL http://dx.doi.org/10.1073/pnas.1903070116.

Roi Livni, Shai Shalev-Shwartz, and Ohad Shamir. On the computational efficiency of training neural networks, 2014. URL https://arxiv.org/abs/1410.1141.

Siyuan Ma, Raef Bassily, and Mikhail Belkin. The power of interpolation: Understanding the effectiveness of sgd in modern over-parametrized learning, 2018. URL https://arxiv.org/abs/1712.06559.

Behnam Neyshabur. Implicit regularization in deep learning, 2017. URL https://arxiv.org/abs/1709.01953.

Samuel L. Smith, Benoit Dherin, David G. T. Barrett, and Soham De. On the origin of implicit regularization in stochastic gradient descent, 2021. URL https://arxiv.org/abs/2101.12176.

Bohan Wang, Qi Meng, Wei Chen, and Tie-Yan Liu. The implicit bias for adaptive optimization algorithms on homogeneous neural networks, 2021. URL https://arxiv.org/abs/2012.06244.