DEEP LEARNING HAS been highly successful in recent years and has led to dramatic improvements in multiple domains. Deep-learning algorithms often generalize quite well in practice, namely, given access to labeled training data, they return neural networks that correctly label unobserved test data. However, despite much research, our theoretical understanding of generalization in deep learning is still limited.

Neural networks used in practice often have far more learnable parameters than training examples. In such overparameterized settings, one might expect overfitting to occur, that is, the learned network might perform well on the training dataset and perform poorly on test data. Indeed, in overparameterized settings, there are many solutions that perform well on the training data, but most of them do not generalize well. Surprisingly, it seems that gradient-based deep-learning algorithms prefer the solutions that generalize well.⁴⁰

Decades of research in learning theory suggest that in order to avoid overfitting one should use a model which is “not more expressive than necessary.” Namely, the model should be able to perform well on the training data but should be as “simple” as possible. This idea goes back to the Occam’s Razor philosophical principle, which argues that we should

---

⁴⁰ Neural networks are trained using gradient-based algorithms, where the network’s parameters are randomly initialized, and then adjusted in many stages in order to fit the training dataset by using information based on the gradient of a loss function w.r.t. the network parameters.
prefer simple explanations over complicated ones. For example, in Figure 1 the data points are labeled according to a degree-3 polynomial plus a small random noise, and we fit it with a degree-3 polynomial (green) and with a degree-8 polynomial (red). The degree-8 polynomial achieves better accuracy on the training data, but it overfits and will not generalize well.

Simplicity in neural networks may stem from having a small number of parameters (which is often not the case in modern deep learning), but may also be achieved by adding a regularization term during the training, which encourages networks that minimize a certain measure of complexity. For example, we may add a regularizer that penalizes models where the Euclidean norm of the parameters (viewed as a vector) is large. However, in practice, neural networks seem to generalize well even when trained without such an explicit regularization, and hence the success of deep learning cannot be attributed to explicit regularization. Therefore, it is believed that gradient-based algorithms induce an implicit bias (or implicit regularization) which prefers solutions that generalize well, and characterizing this bias has been a subject of extensive research.

In this review article, we discuss the implicit bias in training neural networks using gradient-based methods. The literature on this subject has rapidly expanded in recent years, and we aim to provide a high-level overview of some fundamental results. This article is not a comprehensive survey, and there are certainly important results which are not discussed here.

The Double-Descent Phenomenon
An important implication of the implicit bias in deep learning is the

- Deep-learning algorithms exhibit remarkable performance in practice, but it is not well-understood why they are able to generalize despite having more parameters than training examples.
- Exploring generalization in deep learning is an intriguing puzzle with both theoretical and practical implications.
- It is believed that implicit bias is a key factor in the ability of deep-learning algorithms to generalize.

key insights
If we continue to increase the number of parameters in the model after the training set is already perfectly labeled, then there might be a second descent in the test risk (hence the name “double descent”). As can be seen in Figure 2B, by increasing the number of parameters beyond what is required to perfectly fit the training set, the generalization improves. Hence, in contrast to the classical approach which seeks a sweet spot, we may achieve generalization by using overparameterized neural networks. Modern deep learning indeed relies on using highly overparameterized networks.

The double-descent phenomenon is believed to be a consequence of the implicit bias in deep-learning algorithms. When using overparameterized models, gradient methods converge to networks that generalize well by implicitly minimizing a certain measure of the network’s complexity. As we will see next, characterizing this complexity measure in different settings is a challenging puzzle.

Implicit Bias in Classification

We start with the implicit bias in classification tasks, namely, where the labels are in \((-1, 1\). Neural networks are trained in practice using the gradient-descent algorithm and its variants, such as stochastic gradient descent (SGD). In gradient descent, we start from some random initialization \(\theta\), of the network’s parameters, and then in each iteration \(t \geq 1\), we update \(\theta_t = \theta_{t-1} - \eta \nabla \mathcal{L}(\theta_{t-1})\), where \(\eta > 0\) is the step size and \(\mathcal{L}\) is some empirical loss that we wish to minimize. Here, we will focus on gradient flow, which is a continuous version of gradient descent. That is, we train the parameters \(\theta\) of the considered neural network, such that for all time \(t \geq 0\) we have \(\frac{d\theta}{dt} = -\nabla \mathcal{L}(\theta(t))\), where \(\mathcal{L}\) is the empirical loss, and \(\theta(0)\) is some random initialization. \(^\text{b}\) We focus on the logistic loss function (a.k.a. binary cross entropy): For a training dataset \(\{(x_i, y_i)\}_{i=1}^{n} \subseteq \mathbb{R}^d \times \{-1, 1\}\) and a neural network \(\mathcal{N}_\theta : \mathbb{R}^d \to \mathbb{R}\) parameterized by \(\theta\), the empirical loss is defined by \(\mathcal{L}(\theta) = \frac{1}{n} \sum_{i=1}^{n} \log(1 + e^{-y_i \langle w, x_i \rangle})\). We note that gradient flow corresponds to gradient descent with an infinitesimally small step size, and many implicit-bias results are shown for gradient flow since it is often easier to analyze.

Logistic regression. Logistic regression is perhaps the simplest classification setting where the implicit bias of neural networks can be considered. It corresponds to a neural network with a single neuron that does not have an activation function. That is, the model here is a linear predictor \(x \mapsto w^T x\), where \(w \in \mathbb{R}^d\) are the learned parameters.

We assume that the data is linearly separable, that is, there exists some \(\hat{w} \in \mathbb{R}^d\) such that for all \(i \in \{1, \ldots, n\}\), we have \(y_i \cdot \hat{w}^T x_i > 0\). Note that \(\mathcal{L}(w)\) is strictly positive for all \(w\), but it may tend to zero as \(\|w\|_2\) tends to infinity. For example, for \(\alpha \in \mathbb{R}\) and \(w = \alpha \hat{w}\) (where \(\hat{w}\) is the vector defined earlier) we have \(\lim_{\alpha \to \infty} \mathcal{L}(w) = 0\). However, there might be infinitely many directions \(\hat{w}\) that satisfy this condition. Interestingly, Soudry et al.\(^\text{13}\) showed that gradient

\(^\text{b}\) If \(\mathcal{L}\) is non-differentiable, for example, in ReLU networks, then we use the Clarke subdifferential, which is a generalization of the derivative for non-differentiable functions.

---

Figure 1. Fitting training data with a degree-3 polynomial (green) and degree-8 polynomial (red). The latter achieves better accuracy on the training data but it overfits.

Figure 2. The double-descent phenomenon. Dashed lines denote the training risk and solid lines denote the test risk. (A) The classical U-shaped curve; and (B) The double-descent curve, extends the U-shaped curve with a second descent. Beyond the “interpolation threshold” the model fits the training data perfectly, but the test risk decreases. The figure is from Belkin et al.\(^\text{6}\).
flow converges in the direction to the $\ell_2$ maximum-margin predictor. That is, $\mathbf{w} = \lim_{t \to \infty} \mathbf{w}(t)$ exists, and $\mathbf{w}$ points in the direction of the maximum-margin predictor, defined by

$$\text{arg min} ||\mathbf{w}||_2 \; \text{s.t.} \; \forall i \in \{1, \ldots, n\} \; y_i \cdot \mathbf{w}^T \mathbf{x}_i \geq 1.$$  

The characterization of the implicit bias in logistic regression can be used to explain generalization. Consider the case where $d > n$, thus the input dimension is larger than the size of the dataset, and hence there are infinitely many vectors $\mathbf{w}$ with $||\mathbf{w}||_2 = 1$ such that for all $i \in \{1, \ldots, n\}$, we have $y_i \cdot \mathbf{w}^T \mathbf{x}_i > 0$, that is, there are infinitely many directions that correctly label the training data. Some of the directions which fit the training data generalize well, and others do not. The result guarantees that gradient flow converges to the maximum-margin direction. Consequently, we can explain generalization by using standard margin-based generalization bounds (Shalev-Shwartz and Ben-David), which imply that predictors achieving large margins generalize well.

**Linear networks.** We turn to deep neural networks where the activation is the identity function. Such neural networks are called linear networks. These networks compute linear functions, but the network architectures induce significantly different dynamics of gradient flow compared to the case of linear predictors $\mathbf{x} \mapsto \mathbf{w}^T \mathbf{x}$ that we already discussed. Hence, the implicit bias in linear networks has been extensively studied, as a first step toward understanding implicit bias in deep nonlinear networks. It turns out that understanding implicit bias in linear networks is highly non-trivial, and its study reveals valuable insights.

A linear fully-connected network $N : \mathbb{R}^d \to \mathbb{R}$ of depth $k$ computes a function $N(x) = W_k \ldots W_1 x$, where $W_1, \ldots, W_k$ are weight matrices (where $W_1$ is a row vector). The trainable parameters are a collection $\theta = [W_1, \ldots, W_k]$ of the weight matrices. By Ji and Telgarsky, if gradient flow converges to zero loss, namely, $\lim_{t \to \infty} L(\theta(t)) = 0$, then the vector $W_k^T \ldots W_1^T \mathbf{w}$ converges in direction to the $\ell_2$ maximum-margin predictor. That is, although the dynamics of gradient flow in the case of a deep fully-connected linear network is significantly different compared to the case of a linear predictor, in both cases gradient flow is biased toward the $\ell_2$ maximum-margin predictor. We also note that by Ji and Telgarsky, the weight matrices $W_1, \ldots, W_k$ converge to matrices of rank 1. Thus, gradient flow on fully-connected linear networks is also biased toward rank minimization of the weight matrices.

Once we consider linear networks which are not fully connected, gradient flow no longer maximizes the $\ell_2$ margin. For example, Gunasekar et al. showed that in linear diagonal networks (that is, where the weight matrices $W_1, \ldots, W_k$ are diagonal) gradient flow encourages predictors that maximize the margin w.r.t. the $\ell_2$ quasinorm. As a result, diagonal weights are biased toward sparse linear predictors. In linear convolutional networks, they proved bias toward the sparsity of the linear predictors in the frequency domain (see also Yun et al.).

**Homogeneous neural networks.**

Neural networks of practical interest have nonlinear activations and compute nonlinear predictors. The notion of margin maximization in nonlinear predictors is generally not well-defined. Nevertheless, it has been established that for certain neural networks, gradient flow maximizes the margin in a global/local minimum. Intuitively, convergence of the gradient flow in the case of a fully-connected linear network is not sufficient to guarantee convergence to a maximum-margin solution. As we already discussed, in linear classifiers and linear neural networks, margin maximization in the parameter space can explain generalization using well-known margin-based generalization bounds. Can margin maximization in parameter space explain generalization in nonlinear neural networks? In recent years several works showed margin-based generalization bounds for neural networks (for example, Bartlett et al. and Golowich et al.).

---

c Similar results under weaker assumptions were previously established in Gunasekar et al., Ji and Telgarsky.

d A similar result under stronger assumptions was previously established in Nacson et al.
generalization in neural networks might be established by combining these results with results on the implicit bias toward margin maximization in parameter space. On the flip side, we note that it is still unclear how tight the known margin-based generalization bounds for neural networks are, and whether the sample complexity implied by such results (that is, the required size of the training dataset) may be small enough to capture the situation in practice.

Several results on implicit bias were shown for some specific cases of non-linear homogeneous networks. For example: Chizat and Bach\(^4\) showed bias toward margin maximization w.r.t. a certain function norm (known as the variation norm) in infinite-width two-layer homogeneous networks; Lyu et al.\(^2\) proved margin maximization in two-layer Leaky-ReLU networks trained on linearly separable and symmetric data, and Sarassi et al.\(^2\) proved convergence to a linear classifier in two-layer homogeneous networks. Lyu et al.\(^2\) showed bias toward minimizing the number of iterations. We discuss all these important extensions in an online appendix.

Additional aspects of the implicit bias, which apply both to classification and regression, are discussed later.

**Implicit Bias in Regression**

We turn to consider the implicit bias of gradient methods in regression tasks, namely, where the labels are in \(\mathbb{R}\). We focus on gradient flow and on the square-loss function. Thus, we have

\[ \frac{d\theta(t)}{dt} = -\nabla L(\theta(t)), \]

where the empirical loss \( L(\theta) \) is defined such that given a training dataset \( \{(x_i, y_i)\}_{i=1}^{n} \subset \mathbb{R}^d \times \mathbb{R} \) and a neural network \( \mathcal{N}_\theta : \mathbb{R}^d \to \mathbb{R} \), we have \( L(\theta) = \frac{1}{n} \sum_{i=1}^{n} (\mathcal{N}_\theta(x_i) - y_i)^2 \). In an overparameterized setting, there might be many possible choices of \( \theta \) such that \( L(\theta) = 0 \), thus there are many global minima. Ideally, we want to find some implicit regularization function \( \mathcal{R}(\theta) \) such that gradient flow prefers global minima which minimize \( \mathcal{R} \). That is, if gradient flow converges to some \( \theta^\ast \) with \( L(\theta^\ast) = 0 \), then we have \( \theta^\ast \in \text{argmin}_\theta \mathcal{R}(\theta) \) s.t. \( L(\theta) = 0 \).

**Linear regression.** We start with linear regression, which is perhaps the simplest setting where the implicit bias of neural networks can be considered. Here, the model is a linear predictor \( x \mapsto w^\top x \), where \( w \in \mathbb{R}^d \) are the learned parameters. In this case, it is not hard to show that gradient flow converges to the global minimum of \( L(w) \), which is closest to the initialization \( w(0) \) in the \( l_2 \) distance.\(^1\)\(^,\)\(^2\)\(^,\)\(^4\)\(^,\)\(^9\) Thus, the implicit regularization function is \( \mathcal{R}(w)(w) = \|w - w(0)\|_2^2 \). Minimizing this \( l_2 \) norm allows us to explain generalization using standard norm-based generalization bounds.\(^1\)\(^,\)\(^2\)\(^,\)\(^3\)\(^,\)\(^4\)\(^,\)\(^9\)

We note that the results on linear regression can be extended to optimization methods other than gradient flow, such as gradient descent, SGD, and mirror descent.\(^5\)\(^,\)\(^8\)\(^,\)\(^9\)

**Linear networks.** A linear neural network \( \mathcal{N}_\theta : \mathbb{R}^d \to \mathbb{R} \) has the identity activation function and computes a linear predictor \( x \mapsto w \cdot x \), where \( w \) is the vector obtained by multiplying the weight matrices in \( \mathcal{N}_\theta \). Hence, instead of seeking an implicit regularization function \( \mathcal{R}(\theta) \), we may aim to find \( \mathcal{R}(w) \).

Similiar to our discussion in the context of classification, the network architectures in deep linear networks induce significantly different dynamics of gradient flow compared to the case of linear regression. Hence, the implicit bias in such networks has been studied as a first step toward understanding implicit bias in deep nonlinear networks.

Exact expressions for \( \mathcal{R}(w) \) have been obtained for linear diagonal networks and linear fully-connected networks.\(^3\)\(^,\)\(^6\)\(^,\)\(^9\) The expressions for \( \mathcal{R}(w) \) are rather complicated, and depend on the initialization scale, that is, the norm of \( w(0) \), and the initialization “shape,” namely, the relative magnitudes of different weights and layers in the network. Here, we discuss a few special cases.

Recall that diagonal networks are neural networks where the weight matrices are diagonal. The regularization function \( \mathcal{R} \) has been obtained for a few variants of diagonal networks. In Woodworth et al.\(^6\) the authors considered networks of the form \( x \mapsto (u \odot (u \cdot x - u \cdot u))^\top x \), where \( u \odot u \in \mathbb{R}^d \) and the operator \( \odot \) denotes the Hadamard (entrywise) product. Thus, in this network, which can be viewed as a variant of a diagonal network, the weights in the two layers are shared. For an initialization \( u(0) = u \odot u \), the scale \( \alpha > 0 \) of the initialization does not affect \( w(0) = u(0) \odot u(0) - u(0) \odot u(0) = 0 \). However, the initialization scale does affect the implicit bias: if \( \alpha \to 0 \) then \( \mathcal{R}(w) = ||w||_0 \), and if \( \alpha \to \infty \) (which corresponds to the so-called kernel regime) then \( \mathcal{R}(w) = ||w||_2 \). Thus, the implicit bias transitions between the \( l_0 \) norm and the \( l_2 \) norm as we increase the initialization scale. A similar result holds also for deeper networks. In twolayer “diagonal networks” with a similar structure, but where the weights are not shared, Azulay et al.\(^3\) showed that the implicit bias is affected by both the initialization scale and “shape.” In Yun et al.\(^9\) the authors showed a transition between the \( l_0 \) and \( l_2 \) norms both in diagonal networks and in certain convolutional networks (in the frequency domain).

In two-layer fully-connected linear networks with infinitesimally small initialization, gradient flow minimizes the \( l_2 \) norm, namely, \( \mathcal{R}(w) = ||w||_2 \).\(^3\)\(^,\)\(^9\) This result was shown also for...
deep fully-connected linear networks, under certain assumptions on the initialization.\textsuperscript{19}

Some additional aspects of the implicit bias in linear networks are discussed later.

**Matrix factorization as a test-bed for neural networks.** Toward the goal of understanding implicit bias in neural networks, much effort was directed at the matrix factorization (and more specifically, matrix sensing) problem. In this problem, we are given a set of observations about an unknown matrix $W^* \in \mathbb{R}^{d \times d}$ and we find a matrix $W = W_1 W_2 \ldots W_{k-1} W_k$ with $W_k \in \mathbb{R}^{d \times d}$ that is compatible with the given observations by running gradient flow. More precisely, consider the observations $\{(X_i, y_i)\}_{i=1}^{d}$, where in the inner product, we view $W^*$ and $X_i$ as vectors (namely, $\langle W^*, X_i \rangle = \text{tr}(W^* X_i)$). Then, we find $W = W_1 W_2 \ldots W_k$ using gradient flow on the objective $L(W_1, W_2) = \frac{1}{2} \sum_{i=1}^d \langle W_1^* W_2^* X_i, X_i \rangle - y_i^2$. Since $W_k, W_k \in \mathbb{R}^{d \times d}$ then the rank of $W$ is not limited by the parameterization. However, the parameterization has a crucial effect on the implicit bias. Assuming that there are many matrices $W$ that are compatible with the observations, the implicit bias controls which matrix gradient flow finds. A notable special case of matrix sensing is matrix completion. Here, every observation has the form $X_i = e_i^T e_i$, where $\{e_1, \ldots, e_d\}$ is the standard basis. Thus, each observation $(X_i, y_i)$ reveals a single entry from the matrix $W^*$.

The matrix factorization problem is analogous to training a two-layer linear network. Furthermore, we may consider deep matrix factorization where we have $W = W_1 W_2 \ldots W_k$, which is analogous to training a deep linear network. Therefore, this problem is considered a natural test-bed to investigate implicit bias in neural networks, and has been studied extensively in recent years (for example, Arora et al.,\textsuperscript{14} Gunasekar et al.,\textsuperscript{19,20} Razin and Cohen\textsuperscript{29}).

Gunasekar et al.\textsuperscript{14} conjectured that in matrix factorization, the implicit bias of gradient flow starting from small initialization is given by the nuclear norm of $W$ (a.k.a. the trace norm). That is, $\mathcal{R}(W) = \|W\|_\text{Tr}$. They also proved this conjecture in the restricted case where the matrices $\{X_j\}_{j=1}^d$ commute. The conjecture was further studied in a line of works (Arora et al.,\textsuperscript{1} Li et al.,\textsuperscript{19} Razin and Cohen\textsuperscript{29}) providing positive and negative evidence, and was formally refuted by Li et al.\textsuperscript{28} In Razin and Cohen\textsuperscript{29} the authors showed that gradient flow in matrix completion might approach a global minimum at infinity rather than converging to a global minimum with a finite norm. This result suggests that the implicit bias in matrix factorization may not be expressible by any norm or quasi-norm. They conjectured that the implicit bias can be explained by rank minimization rather than norm minimization. Li et al.\textsuperscrap{29} provide theoretical and empirical evidence that gradient flow with infinitesimally small initialization in the matrix-factorization problem is mathematically equivalent to a simple heuristic rank-minimization algorithm called Greedy Low-Rank Learning.

Overall, although an explicit expression for a function $\mathcal{R}$ is not known in the case of matrix factorization, we can view the implicit bias as a heuristic for rank minimization. It is still unclear what the implications of these results are for more practical nonlinear neural networks.

**Nonlinear networks.** Once we consider nonlinear networks the situation is even less clear. For a single-neuron network, namely, $X = \sigma(W^* X)$, where $\sigma: \mathbb{R} \rightarrow \mathbb{R}$ is strictly monotonic, gradient flow converges to the closest global minimum in $\ell_2$ distance, similarly to the case of linear regression. Thus, we have $\mathcal{R}_{\mathcal{Q}}(W^*) = \|W - W^*\|_{\mathcal{Q}}$. However, if $\sigma$ is the ReLU activation then the implicit bias is not expressible by any non-trivial function of $w$. A bit more precisely, suppose that $\mathcal{R}: \mathbb{R}^d \rightarrow \mathbb{R}$ is such that if gradient flow starting from $w(0) = 0$ converges to a zero-loss solution $w^*$, then $w^*$ is a zero-loss solution that minimizes $\mathcal{R}$. Then, such a function $\mathcal{R}$ must be constant in $\mathbb{R}^d \setminus \{0\}$. Hence, the approach of precisely specifying the implicit bias of gradient flow via a regularization function is not feasible in single-neuron ReLU networks.\textsuperscript{34} It suggests that this approach may not be useful for understanding implicit bias in the general case of ReLU networks.

On the positive side, in single-neuron ReLU networks, the implicit bias can be expressed approximately, within a factor of 2, by the $\ell_2$ norm. Namely, let $w^*$
be a zero-loss solution with a minimal $\ell_1$ norm, and assume that gradient flow converges to some zero-loss solution $w(\infty)$, then $\|w(\infty)\|_1 \leq 2\|w^*\|_1$.\textsuperscript{34}

For two-layer Leaky-ReLU networks with a single hidden neuron, an explicit expression for $R$ was obtained in Azulay et al.\textsuperscript{4} However, if we replace the Leaky-ReLU activation with ReLU, then the implicit bias is not expressible by any non-trivial function, similar to the case of a single-neuron ReLU network.\textsuperscript{34}

Overall, our understanding of implicit bias in nonlinear networks with regression losses is very limited. While in classification we have a useful characterization of the implicit bias in nonlinear homogeneous models, here we do not understand even extremely simple models. Improving our understanding of this question is a major challenge for the upcoming years.

In what follows, we will discuss some additional aspects of the implicit bias, which apply both to regression and classification.

**Dynamical Stability**

In the previous sections, we mostly focused on implicit bias in gradient flow, and also discussed cases where the results on gradient flow can be extended to other gradient methods. However, when considering gradient descent or SGD with a finite step size (rather than infinitesimal), the discrete nature of the algorithms and the stochasticity induce additional implicit biases that do not exist in gradient flow. These biases are crucial for understanding the behavior of gradient methods in practice, as empirical results suggest that increasing the step size and decreasing the batch size may improve generalization (Jastrzębski et al.,\textsuperscript{15} Keskar et al.\textsuperscript{18}).

It is well known that gradient descent and SGD cannot stably converge to minima that are too sharp relative to the step size (see, for example, Mulayoff and Michaeli,\textsuperscript{24} Mulayoff et al.,\textsuperscript{33} Nacson et al.,\textsuperscript{27} Wu et al.\textsuperscript{37}). Namely, in a stable minimum, the maximal eigenvalue of the Hessian is at most $2/\eta$, where $\eta$ is the step size. As a result, when using an appropriate step size we can rule out stable convergence to certain (sharp) minima, and thus encourage convergence to flat minima, which are believed to generalize better.\textsuperscript{18, 37} Similarly, small batch sizes in SGD also encourage flat minima.\textsuperscript{18, 23, 37, 38}

We note that a function might be represented with many different networks, that is, there are many sets of parameters that correspond to the same function. However, it is possible that some of these representations are sharp (in parameter space) and others are flat (Dinh et al.). As a result, understanding the implications in the function space of the bias toward flat minima (in parameter space) is often a challenging question.

Implications of the implicit bias toward flat minima were studied in several works. For example, Mulayoff et al.\textsuperscript{25} studied flat minima in univariate ReLU networks, and showed that SGD with large step size is biased toward functions whose second derivative (w.r.t. the input) has a bounded weighted $\ell_1$ norm, which encourages convergence to smooth functions; Nacson et al.\textsuperscript{27} showed that gradient descent with large step size on a diagonal linear network with non-centered data minimizes the $\ell_1$ norm of the corresponding linear predictor, even in settings where gradient descent with small step size minimizes the $\ell_2$ norm; Ma and Ying\textsuperscript{23} studied minima stability in SGD, and showed that flat minima induce regularization of the function’s gradient (w.r.t. the input data), and more generally that SGD regularizes Sobolev seminorms of the function; Mulayoff and Michaeli\textsuperscript{24} studied flat minima in linear neural networks and showed bias toward solutions where the magnitudes of the layers are balanced, as well as other properties of flat solutions. We note that all these papers consider regression with the square loss.

An intriguing related phenomenon, observed by Cohen et al.,\textsuperscript{9} is the tendency of gradient descent to operate in a regime called the Edge of Stability (EoS). They showed empirically for several architectures and tasks, and for both the cross-entropy and the square loss, that the behavior of gradient descent with step size $\eta$ consists of two stages. First, the loss curvature grows until the sharpness touches the bound of $2/\eta$. Then, gradient descent enters the EoS regime,
where curvature hovers around this bound, and the train loss behaves non-monotonically, yet consistently decreases over long timescales. This phenomenon is not well understood, as traditional convergence analyses of gradient descent do not apply when the sharpness is above $2/\eta$. The study of the EoS regime may contribute to our understanding of the dynamics and implicit bias in gradient descent and SGD. It was investigated in several recent theoretical works. For example, Arora et al. showed that training with modified gradient descent or modified loss (for a large family of losses) provably enters the EoS regime, and that the loss decreases in a non-monotone manner, and Chen and Bruna studied EoS in training a two-layer single-neuron ReLU network with the square loss.

**Additional Implicit Biases and Further Implications**

Additional aspects of the implicit bias were studied in the literature. These aspects include the effect of label noise on the implicit bias in SGD; transforming gradient descent and SGD into gradient flow on a modified loss; balancedness of weights’ magnitudes between different layers; implications of norm minimization in parameter space on the predictor in function space; as well as implicit bias in the NTK regime. We briefly discuss these aspects as well as further implications of the implicit bias other than generalization, namely, implications on adversarial robustness and on privacy in an online appendix (https://dl.acm.org/doi/10.1145/3571070).

**Conclusion**

Deep-learning algorithms exhibit remarkable performance, but it is not well-understood why they are able to generalize despite having much more parameters than training examples. Exploring generalization in deep learning is an intriguing puzzle with both theoretical and practical implications. It is believed that implicit bias is a main ingredient in the ability of deep-learning algorithms to generalize, and hence it has been widely studied.

Our understanding of implicit bias improved dramatically in recent years but is still far from satisfactory. We believe that further progress in the study of implicit bias will eventually shed light on the mystery of generalization in deep learning.

**Acknowledgments**

I thank Nadav Cohen, Noam Razin, Ohad Shamir, and Daniel Soudry for valuable comments and discussions.

**References**

1. Arora, S., Cohen, N., Hu, W., Luo, Y. Implicit regularization in deep matrix factorization. In Advances in Neural Information Processing Systems (2018), Curran Associates, Inc, Red Hook, NY, 7413–7424.
2. Arora, S., Li, Z., Panigrahi, A. Understanding gradient descent on edge of stability in deep learning. arXiv preprint arXiv:2205.09745 (2022).
3. Azizan, N., Hassibi, B. Stochastic gradient/mirror descent: Minimax optimality and implicit regularization. In Conference on Learning Representations (ICLR) (2019), OpenReview.net.
4. Azizy, S., Moroshko, E., Nacson, M.S., Woodworth, B., Srebro, N. A dual view on the implicit bias of initialization shape: Beyond infinitesimal mirror descent. In International Conference on Machine Learning (2021), PMLR, 468–477.
5. Bartlett, P.L., Foster, D.J., Telgarsky, M.J. Spectrally-normalized margin bounds for neural networks. Adv Neural Inf. Process. Syst. 30 (2017), 6240–6248.
6. Belkin, M., Hsu, D., Ma, S., Mandal, S. Reconciling modern machine-learning practice and the classical bias–variance trade-off. Proc. Natl. Acad. Sci. 116, 32 (2019), 15849–15894.
7. Chen, L., Bruna, J. On gradient descent convergence beyond the edge of stability. arXiv preprint arXiv:2208.04170 (2022).
8. Chizat, L., Bach, F. Implicit bias of gradient descent for wide two-layer neural networks trained with the logit loss. In Conference on Learning Theory (2020), PMLR, 1305–1338.
9. Cohen, J.M., Kaur, S., Li, Y., Zico Kolter, J. Taliwalker, A. Gradient descent on neural networks typically occurs at the edge of stability. In International Conference on Learning Representations (ICLR) (2021), OpenReview.net.
10. Drich, L., Pascanu, R., Bengio, S., Bengio, Y. Sharp minima can generalize for deep nets. In International Conference on Machine Learning (2017), PMLR, 1019–1028.
11. Golowich, N., Rakhlin, A., Shamir, O. Size-independent sample complexity of neural networks. In Conference On Learning Theory (2018), PMLR, 297–298.
12. Gunasekar, S., Lee, J., Soudry, D., Srebro, N. Characterizing implicit bias in terms of optimization geometry. In International Conference on Machine Learning (2018), PMLR, 8341–8349.
13. Gunasekar, S., Lee, J.D., Soudry, D., Srebro, N. Implicit bias of gradient descent on linear convolutional networks. In Advances in Neural Information Processing Systems (2018), Curran Associates, Inc, Red Hook, NY, 9461–9471.
14. Gunasekar, S., Woodworth, E.B., Bhosanjapalli, S., Neyshabur, B., Srebro, N. Implicit regularization in matrix factorization. Adv. Neural Inf. Process. Syst. 30 (2017), 6151–6159.
15. Jastrzębski, S., Kulicke, T., Arpit, D., Ballas, N., Fischer, A., Bengio, Y. et al. Three factors influencing minima in SGD. arXiv preprint arXiv:1711.04093 (2017).
16. Ji, Z., Telgarsky, M. Gradient descent aligns the layers of deep linear networks. In International Conference on Learning Representations (2018), OpenReview.net.
17. Ji, Z., Telgarsky, M. Directional convergence and alignment in deep learning. In Advances in Neural Information Processing Systems (2020), Curran Associates, Inc, Red Hook, NY.
18. Keskar, N.S., Mudigere, D., Nocedal, J., Smelyanskiy, M., Tang, P.T.P. On large-batch training for deep learning: Generalization gap and sharp minima. In International Conference on Learning Representations (ICLR) (2017), OpenReview.net.
19. Li, Y., Ma, T., Zhang, H. Algorithmic regularization in over-parameterized matrix sensing and neural networks with quadratic activations. In Conference on Learning Theory (2018), PMLR, 2–47.
20. Li, Z., Luo, Y., Lyu, K. Towards Resolving the Implicit Bias of Gradient Descent for Matrix Factorization: Greedy Low-Rank Learning. In International Conference on Learning Representations (2020), OpenReview.net.
21. Lyu, K., Li, J. Gradient descent maximizes the margin of homogeneous neural networks. In International Conference on Learning Representations (2020), PMLR, 123–131.
22. Lyu, K., Li, Z., Wang, R., Arora, S. Gradient descent on two-layer nets: Margin maximization and simplicity bias. Adv. Neural Inf. Process. Syst. 34 (2021), 17749–17761.
23. Ma, C., Ying, L. On linear stability of SGD and input-smoothness of neural networks. Adv. Neural Inf. Process. Syst. 34 (2021), 18805–18817.
24. Malyutin, R., Michaeli, T. Unique properties of flat minima in deep networks. In International Conference on Machine Learning (2020), PMLR, 7108–7118.
25. Malyutin, R., Michaeli, T., Soudry, D. The implicit bias of minima stability: A view from function space. Adv. Neural Inf. Process. Syst. 34 (2021), 17749–17761.
26. Nacson, M.S., Gunasekar, S., Lee, J., Srebro, N., Soudry, D. Lexicographic and depth-sensitive margins in homogeneous and non-homogeneous deep models. In International Conference on Machine Learning (2019), PMLR, 4683–4692.
27. Nacson, M.S., Ravichandran, K., Srebro, N., D. Implicit Bias of the Linear Diagonal Neural Networks. In International Conference on Machine Learning (2022), PMLR, 16270–16295.
28. Neyshabur, B., Bhogajapalli, S., McAi1essler, D., Srebro, N. Exploring generalization in deep learning. In Advances in Neural Information Processing Systems (2017), Curran Associates, Inc, Red Hook, NY, 5947–5956.
29. Razin, N., Cohen, N. Implicit regularization in deep learning may not be explainable by norms. Adv. Neural Inf. Process. Syst. 34 (2021), 1019–1028.
30. Safran, I., Vardi, G., Lee, J.D. On the effective number of linear regions in shallow univariate ReLU networks: Convergence guarantees and implicit bias. arXiv preprint arXiv:2205.08972 (2022).
31. Sarusi, R., Brutzkus, A., Globerson, A. Towards understanding learning in neural networks with linear teachers. In International Conference on Machine Learning (2021), PMLR, 9313–9322.
32. Shalev-Shwartz, S., Ben-David, S. Understanding Machine Learning: From Theory to Algorithms. Cambridge University Press, 2014.
33. Soudry, D., Hofer, E., Nacson, M.S., Gunasekar, S., Srebro, N. The implicit bias of gradient descent on separate data. J Mach. Learn. Res. 19, 1 (2018), 2822–2878.
34. Vardi, G., Shamir, O. Implicit regularization in relu networks with the square loss. In Conference on Learning Theory (2021), PMLR, 4224–4268.
35. Vardi, G., Shamir, O., Srebro, N. On margin maximization in linear and ReLU networks. arXiv preprint arXiv:2110.00802 (2021).
36. Woodworth, B., Gunasekar, S., Lee, J.D., Moroshko, E., Savarese, P., Golan, I., et al. Kernel and rich regimes in overparameterized models. In Conference on Learning Theory (2020), PMLR, 3635–3673.
37. Wu, L., Ma, C., et al. How sgld selects the global minima in overparameterized learning: A dynamical stability perspective. Adv. Neural Inf. Process. Syst. 31 (2018).
38. Wu, L., Wang, M., Su, W. When does SGD favor flat minima? A quantitative characterization of linear dependence and SGD. In International Conference on Learning Representations (ICLR) (2021), OpenReview.net.
39. Zhang, C., Bengio, S., Hardt, M., Recht, B., Vinyals, O. Understanding deep learning (still) requires rethinking generalization. Commun. ACM 64, 3 (2021), 107–115.

Gal Vardi is a postdoctoral scholar at TTI-Chicago and the Hebrew University, Jerusalem, Israel.