Fast Margin Maximization via Dual Acceleration

Ziwei Ji [1] **Nathan Srebro** [2] **Matus Telgarsky** [1]

Abstract

We present and analyze a momentum-based gradient method for training linear classifiers with an exponentially-tailed loss (e.g., the exponential or logistic loss), which maximizes the classification margin on separable data at a rate of $\widetilde{\mathcal{O}}(1/t^2)$. This contrasts with a rate of $\mathcal{O}(1/\log(t))$ for standard gradient descent, and $\mathcal{O}(1/t)$ for normalized gradient descent. This momentum-based method is derived via the convex dual of the maximum-margin problem, and specifically by applying Nesterov acceleration to this dual, which manages to result in a simple and intuitive method in the primal. This dual view can also be used to derive a stochastic variant, which performs adaptive non-uniform sampling via the dual variables.

1. Introduction

First-order optimization methods, such as stochastic gradient descent (SGD) and variants thereof, form the optimization backbone of deep learning, where they can find solutions with both low training error and low test error (Neyshabur et al., 2014; Zhang et al., 2016). Motivated by this observation of low test error, there has been extensive work on the *implicit bias* of these methods: amongst those predictors with low training error, which predictors do these methods implicitly prefer?

For linear classifiers and linearly separable data, Soudry et al. (2017) prove that gradient descent can not only minimize the training error, but also maximize the *margin*. This could help explain the good generalization of gradient descent, since a larger margin could lead to better generalization (Bartlett et al., 2017). However, gradient descent can only maximize the margin at a slow $\mathcal{O}(1/\log(t))$ rate.

It turns out that the margin can be maximized much faster by simply normalizing the gradient: letting θ_t denote the step size and $\mathcal{R}$ the empirical risk with the exponential loss, consider the normalized gradient step

$$w_{t+1} := w_t - \theta_t \frac{\nabla \mathcal{R}(w_t)}{\mathcal{R}(w_t)}.$$

Using this normalized update, margins are maximized at a $\widetilde{\mathcal{O}}(1/\sqrt{t})$ rate with $\theta_t = 1/\sqrt{t}$ (Nacson et al., 2018), and at a $\mathcal{O}(1/t)$ rate with $\theta_t = 1$ (Ji & Telgarsky, 2019). A key observation in proving such rates is that normalized gradient descent is equivalent to an entropy-regularized mirror descent on a certain margin dual problem (cf. Section 3.1).

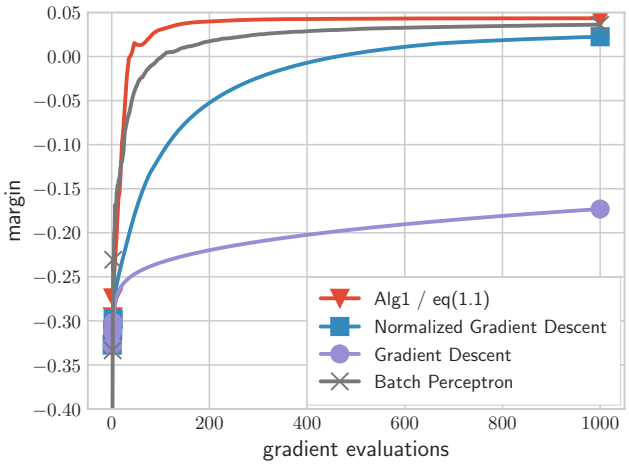


Figure 1. Margin-maximization performance of the new momentum-based method (cf. Algorithm 1 and eq. (1.1)), which has a rate $\widetilde{\mathcal{O}}(1/t^2)$, compared with prior work discussed below. All methods are first-order methods, and all but batch perceptron use an exponentially-tailed smooth loss, whereas batch perceptron applies gradient descent to the hard-margin problem directly. The data here is linearly separable, specifically `mnist` digits 0 and 1.

Contributions. In this work, we further exploit this duality relationship from prior work, and design a momentum-based algorithm with iterates given by

$$\begin{aligned} g_t &:= \beta_t \left(g_{t-1} + \frac{\nabla \mathcal{R}(w_t)}{\mathcal{R}(w_t)} \right), \\ w_{t+1} &:= w_t - \theta_t \left(g_t + \frac{\nabla \mathcal{R}(w_t)}{\mathcal{R}(w_t)} \right). \end{aligned} \tag{1.1}$$

[1]Department of Computer Science, University of Illinois at Urbana-Champaign, Urbana, Illinois, USA [2]Toyota Technical Institute of Chicago, Chicago, Illinois, USA. Correspondence to: Ziwei Ji <ziweiji2@illinois.edu>.

Proceedings of the 38^{th} International Conference on Machine Learning, PMLR 139, 2021.

Our main result is that these iterates, with a proper choice of θ_t and β_t, can maximize the margin at a rate of $\widetilde{\mathcal{O}}(1/t^2)$, whereas prior work had a rate of $\mathcal{O}(1/t)$ at best. The key idea is to reverse the primal-dual relationship mentioned above: those works focus on primal normalized gradient descent, and show that it is equivalent to dual mirror descent, but here we start from the dual, and apply Nesterov acceleration to make dual optimization faster, and then translate the dual iterates into the momentum form in eq. (1.1). Note that if our goal is just to accelerate dual optimization, then it is natural to apply Nesterov's method; however, here our goal is to accelerate (primal) margin maximization – it was unclear whether the momentum method changes the implicit bias, and our margin analysis is very different from the standard analysis of Nesterov's method. The connection between momentum in the primal and acceleration in the dual also appears to be new, and we provide it as an auxiliary contribution. We state the method in full in Algorithm 1, and its analysis in Section 3.

Since our momentum-based iterates (cf. eq. (1.1)) are designed via a primal-dual framework, they can be written purely with dual variables, in which case they can be applied in a kernel setting. However, calculating the full-batch gradient would require n^2 calls to the kernel, where n denotes the number of training examples. To reduce this computational burden, by further leveraging the dual perspective, we give an *adaptive sampling* procedure which avoids the earlier use of batch gradients and only needs n kernel calls per iteration. We prove a $\mathcal{O}(1/\sqrt{t})$ margin rate for a momentum-free version of this adaptive sampling method, but also provide empirical support for an aggressive variant which uses our batch momentum formulation verbatim with these efficient stochastic updates. These results are presented in Section 4.

For sake of presentation, the preceding analyses and algorithm definitions use the exponential loss, however they can be extended to both binary and multiclass losses with exponential tails. The multiclass extension is in fact a straightforward reduction to the binary case, and is used in most figures throughout this work. We discuss these extensions in Section 5.

As an illustrative application of these fast margin maximization methods, we use them to study the evolution of the kernel given by various stages of deep network training. The main point of interest is that while these kernels do seem to generally improve during training (in terms of both margins and test errors), we provide an example where simply changing the random seed switches between preferring the final kernel and the initial kernel. These empirical results appear in Section 6.

We conclude with open problems in Section 7. Full proofs and further experimental details are deferred to the appendices.

1.1. Related Work

This work is closely related to others on the implicit bias, most notably the original analysis for gradient descent on linearly separable data (Soudry et al., 2017). The idea of using normalized steps to achieve faster margin maximization rates was first applied in the case of *coordinate* descent (Telgarsky, 2013), where this normalization is closely associated with the usual step sizes in boosting methods (Freund & Schapire, 1997). Many other works have used these normalized iterates, associated potential functions, and duality concepts, both in the linear case (Gunasekar et al., 2018a; Ji & Telgarsky, 2018), and in the nonlinear case (Gunasekar et al., 2018b; Lyu & Li, 2019; Chizat & Bach, 2020; Ji & Telgarsky, 2020).

There appear to be few analyses of momentum methods; one example is the work of Ghadimi et al. (2015), which shows a $\mathcal{O}(1/t)$ convergence rate for general convex problems over bounded domains, but can not be applied to the exponentially-tailed loss setting here since the domain is unbounded and the solutions are off at infinity. Connections between momentum in the primal and Nesterov acceleration in the dual seem to not have been made before, and relatedly our use of momentum coefficient $\beta_t = t/(t+1)$ is non-standard.

Further on the topic of acceleration, Tseng (2008) gave an application to a smoothed version of the nonsmooth hard-margin objective, with a rate of $\mathcal{O}(1/t)$ to a fixed suboptimal margin. This analysis requires accelerated methods for general geometries, which were analyzed by Tseng (2008) and Allen-Zhu & Orecchia (2014). The original accelerated method for Euclidean geometry is due to Nesterov (1983). A simultaneous analysis of mirror descent and Nesterov acceleration is given here in Appendix B.

The methods here, specifically Proposition 3.7, can ensure a margin of $\bar{\gamma}/4$ in $4\sqrt{\ln(n)}/\bar{\gamma}$ steps, where $\bar{\gamma}$ denotes the optimal margin and will be defined formally in Section 2. Another primal-dual method for fast *linear feasibility* was given by Hanashiro & Abernethy (2020); the method terminates in $\mathcal{O}\left(\ln(n)/\bar{\gamma}\right)$ steps with a positive margin, however the analysis does not reveal how large this margin is.

Various figures throughout this work include experiments with the *batch perceptron*, which simply applies (super)gradient ascent to the explicit hard-margin maximization problem (Cotter et al., 2012). Despite this simplicity, the method is hard to beat, and surpasses prior implicit margin maximizers in experiments (cf. Figure 1). Interestingly, another standard method with strong guarantees fared less well in experiments (Clarkson et al., 2012), and is thus omitted from the figures.

2. Notation

The dataset is denoted by $\{(x_i, y_i)\}_{i=1}^n$, where $x_i \in \mathbb{R}^d$ and $y_i \in \{-1, +1\}$. Without loss of generality, we assume $\|x_i\|_2 \leq 1$. Moreover, let $z_i := -y_i x_i$, and collect these vectors into a matrix $Z \in \mathbb{R}^{n \times d}$, whose i-th row is $z_i^\top$.

We consider linear classifiers. The margin of a nonzero linear classifier $w \in \mathbb{R}^d$ is defined as

$$\gamma(w) := \frac{\min_{1 \leq i \leq n} y_i \langle w, x_i \rangle}{\|w\|_2} = \frac{-\max_{1 \leq i \leq n} \langle w, z_i \rangle}{\|w\|_2},$$

with $\gamma(0) := 0$. The *maximum margin* is

$$\bar{\gamma} := \max_{\|w\|_2 \leq 1} \gamma(w).$$

If $\bar{\gamma} > 0$, then the dataset is *linearly separable*; in this case, the *maximum-margin classifier* is defined as

$$\bar{u} := \arg\max_{\|w\|_2 \leq 1} \gamma(w) = \arg\max_{\|w\|_2 = 1} \gamma(w).$$

If $\bar{\gamma} = 0$, the dataset is linearly nonseparable.

Our algorithms are based on the empirical risk, defined as

$$\mathcal{R}(w) := \frac{1}{n} \sum_{i=1}^n \ell\left(\langle w, z_i \rangle\right).$$

For presentation, we mostly focus on the exponential loss $\ell(z) := e^z$, but our analysis can be extended to other exponentially-tailed losses such as the logistic loss $\ell(z) := \ln(1 + e^z)$ and various multiclass losses; these extensions are discussed in Section 5.

The following potential function $\psi : \mathbb{R}^n \to \mathbb{R}$ will be central to our analysis: given a strictly increasing loss $\ell : \mathbb{R} \to \mathbb{R}$ with $\lim_{z \to -\infty} \ell(z) = 0$ and $\lim_{z \to \infty} \ell(z) = \infty$, for $\xi \in \mathbb{R}^n$, let

$$\psi(\xi) := \ell^{-1}\left(\sum_{i=1}^n \ell(\xi_i)\right), \tag{2.1}$$

thus $\psi(Zw) = \ell^{-1}\left(n\mathcal{R}(w)\right)$. For the exponential loss, ψ is the ln-sum-exp function, meaning $\psi(Zw) = \ln\left(\sum_{i=1}^n \exp(\langle w, z_i \rangle)\right)$. This ψ is crucial in our analysis since (i) it induces the dual variable, which motivates our algorithms (cf. Section 3.1); (ii) it gives a smoothed approximation of margin, which helps in the margin analysis (cf. Section 3.3). Here we note another useful property of ψ: the gradient of $\psi(Zw)$ with respect to w is $Z^\top \nabla \psi(Zw)$, which is a normalized version of $\nabla \mathcal{R}(w)$:

$$Z^\top \nabla \psi(Zw) = \frac{\sum_{i=1}^n \ell'\left(\langle w, z_i \rangle\right) z_i}{\ell'\left(\psi(Zw)\right)} = \frac{\nabla \mathcal{R}(w)}{\ell'\left(\psi(Zw)\right)/n}. \tag{2.2}$$

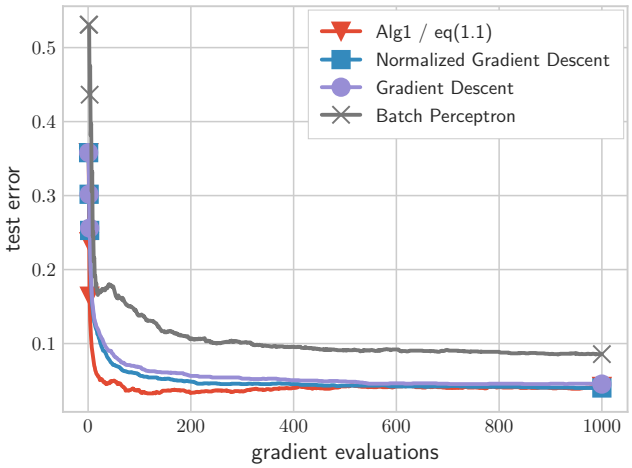


Figure 2. Here the various margin-maximization methods from Figure 1 are run on *non-separable* data, specifically `mnist` digits 3 and 5; as such, test error and not margin are reported. The methods based on exponential loss still perform well; by contrast, the batch perceptron suffers, and perhaps requires additional effort to tune a regularization parameter.

Algorithm 1

Input: data matrix $Z \in \mathbb{R}^{n \times d}$, step size $(\theta_t)_{t=0}^\infty$, momentum factor $(\beta_t)_{t=0}^\infty$.
Initialize: $w_0 = g_{-1} = (0, \ldots, 0) \in \mathbb{R}^d$, $q_0 = (\frac{1}{n}, \ldots, \frac{1}{n}) \in \Delta_n$.
for $t = 0, 1, 2, \ldots$ **do**
  $g_t \leftarrow \beta_t(g_{t-1} + Z^\top q_t)$.
  $w_{t+1} \leftarrow w_t - \theta_t\left(g_t + Z^\top q_t\right)$.
  $q_{t+1} \propto \exp(Z w_{t+1})$, and $q_{t+1} \in \Delta_n$.
end for

For the exponential loss, $\nabla \psi(Zw) \in \Delta_n$ is just the softmax mapping over Zw, where Δ_n denotes the probability simplex. Moreover,

$$Z^\top \nabla \psi(Zw) = \frac{\nabla \mathcal{R}(w)}{\mathcal{R}(w)}. \tag{2.3}$$

3. Analysis of Algorithm 1

A formal version of our batch momentum method is presented in Algorithm 1. It uses the exponential loss, and is equivalent to eq. (1.1) since by eq. (2.3),

$$Z^\top q_t = Z^\top \nabla \psi(Z w_t) = \frac{\nabla \mathcal{R}(w_t)}{\mathcal{R}(w_t)}.$$

Here are our main convergence results.

Theorem 3.1. *Let w_t and g_t be given by Algorithm 1 with $\theta_t = 1$ and $\beta_t = t/(t+1)$.*

1. *If the dataset is separable, then for all $t \geq 1$,*

$$\gamma(w_t) \geq \bar{\gamma} - \frac{4\left(1+\ln(n)\right)\left(1+2\ln(t+1)\right)}{\bar{\gamma}(t+1)^2}.$$

2. *For any dataset, separable or nonseparable, it holds for all $t \geq 1$ that*

$$\frac{4\|g_t\|_2^2}{t^2} - \frac{8\ln(n)}{(t+1)^2} \leq \bar{\gamma}^2 \leq \frac{4\|g_t\|_2^2}{t^2}.$$

Our main result is in the separable case, where Algorithm 1 can maximize the margin at a $\widetilde{\mathcal{O}}(1/t^2)$ rate; by contrast, as mentioned in the introduction, all prior methods have a $\mathcal{O}(1/t)$ rate at best. On the other hand, for any dataset, our algorithm can find an interval of length $\mathcal{O}(1/t^2)$ which includes $\bar{\gamma}^2$, in particular certifying non-existence of predictors with margin larger than any value in this interval. Moreover, as shown in Figure 2, Algorithm 1 can also achieve good test accuracy even in the nonseparable case; it is an interesting open problem to build a theory for this phenomenon.

The rest of this section sketches the proof of Theorem 3.1, with full details deferred to the appendices. In Section 3.1, we first consider gradient descent without momentum (i.e., $\beta_t = 0$), which motivates consideration of a dual problem. Then in Section 3.2, we apply Nesterov acceleration (Nesterov, 2004; Tseng, 2008; Allen-Zhu & Orecchia, 2014) to this dual problem, and further derive the corresponding primal method in Algorithm 1, and also prove the second part of Theorem 3.1. Finally, we give a proof sketch of the margin rate in Section 3.3.

3.1. Motivation from Gradient Descent

We start by giving an alternate presentation and discussion of certain observations from the prior work of Ji & Telgarsky (2019), which in turn motivates Algorithm 1.

Consider gradient descent $w_{t+1} := w_t - \eta_t \nabla\mathcal{R}(w_t)$. Define the dual variable by $q_t := \nabla\psi(Zw_t)$; for the exponential loss, it is given by $q_t \propto \exp(Zw_t)$, $q_t \in \Delta_n$. Note that

$$\begin{aligned} w_{t+1} &= w_t - \eta_t \nabla\mathcal{R}(w_t) \\ &= w_t - \eta_t \mathcal{R}(w_t)\frac{\nabla\mathcal{R}(w_t)}{\mathcal{R}(w_t)} \\ &= w_t - \theta_t Z^\top q_t, \end{aligned}$$

where we let $\theta_t = \eta_t\mathcal{R}(w_t)$. Moreover,

$$\begin{aligned} q_{t+1} \propto \exp\left(Zw_{t+1}\right) &= \exp\left(Zw_t - \theta_t ZZ^\top q_t\right) \\ &\propto q_t \odot \exp\left(-\theta_t ZZ^\top q_t\right) \\ &= q_t \odot \exp\left(-\theta_t \nabla\phi(q_t)\right), \end{aligned}$$

where $\phi(q) := \left\|Z^\top q\right\|_2^2/2$ and $\odot$ denotes coordinate-wise product. In other words, the update from q_t to q_{t+1} is a mirror descent / dual averaging update with the entropy regularizer on the dual objective ϕ.

This dual objective $\|Z^\intercal q\|_2^2/2$ is related to the usual hard-margin dual objective, and is evocative of the SVM dual problem; this connection is made explicit in Appendix A. Even without deriving this duality formally, it makes sense that q_t tries to minimize ϕ, since ϕ encodes extensive structural information of the problem: for instance, if the dataset is not separable, then $\min_{q\in\Delta_n}\phi(q) = 0$ (cf. Lemma A.1). With a proper step size, we can ensure

$$\phi(q_t) = \frac{\left\|\nabla\mathcal{R}(w_t)\right\|_2^2}{2\mathcal{R}(w_t)^2} \to 0, \quad \mathcal{R}(w_t) \text{ is nonincreasing},$$

and it follows that $\left\|\nabla\mathcal{R}(w_t)\right\|_2 \to 0$. If the dataset is separable, then $\min_{q\in\Delta_n}\phi(q) = \bar{\gamma}^2/2$ (cf. Lemma A.1), and

$$Z^\top\bar{q} = \bar{\gamma}\bar{u}, \quad \text{for } \bar{q} \in \arg\min_{q\in\Delta_n}\phi(q),$$

where $\bar{u}$ is the unique maximum-margin predictor, as defined in Section 2. As q_t minimizes ϕ, the vector $Z^\top q_t$ becomes biased towards $\bar{u}$, by which we can also show $w_t/\|w_t\|_2 \to \bar{u}$. Ji & Telgarsky (2019) use this idea to show a $\mathcal{O}(1/t)$ margin maximization rate for primal gradient descent.

The idea in this work is to reverse the above process: we can start from the dual and aim to minimize ϕ more efficiently, and then take the dual iterates $(q_t)_{t=0}^\infty$ from this more efficient minimization and use them to construct primal iterates $(w_t)_{t=0}^\infty$ satisfying $\nabla\psi(Zw_t) = q_t$. It is reasonable to expect such w_t to maximize the margin faster, and indeed we show this is true in the following, by applying Nesterov acceleration to the dual, thanks to the ℓ_1 smoothness of ϕ (Ji & Telgarsky, 2019, Lemma 2.5).

3.2. Primal and Dual Updates

To optimize the dual objective ϕ, we apply Nesterov's method with the ℓ_1 geometry (Tseng, 2008; Allen-Zhu & Orecchia, 2014). The following update uses the entropy regularizer; more general updates are given in Appendix B.

Let $\mu_0 = q_0 := (\frac{1}{n}, \ldots, \frac{1}{n})$. For $t \geq 0$, let $\lambda_t, \theta_t \in (0, 1]$, and

$$\begin{aligned} \nu_t &:= (1-\lambda_t)\mu_t + \lambda_t q_t, \\ q_{t+1} &\propto q_t \odot \exp\left(-\frac{\theta_t}{\lambda_t} ZZ^\top \nu_t\right), \quad q_{t+1} \in \Delta_n, \\ \mu_{t+1} &:= (1-\lambda_t)\mu_t + \lambda_t q_{t+1}. \end{aligned}$$

If we just apply the usual mirror descent / dual averaging to ϕ, then ϕ can be minimized at a $\mathcal{O}(1/t)$ rate (Ji & Telgarsky, 2019, Theorem 2.2). However, using the above accelerated process, we can minimize ϕ at a $\mathcal{O}(1/t^2)$ rate.

Lemma 3.2. *For all $t \geq 0$, let $\theta_t = 1$ and $\lambda_t = 2/(t+2)$. Then for all $t \geq 1$ and $\bar{q} \in \arg\min_{q \in \Delta_n} \phi(q)$,*

$$\phi(\mu_t) - \phi(\bar{q}) \leq \frac{4\ln(n)}{(t+1)^2}.$$

Next we construct corresponding primal variables $(w_t)_{t=0}^{\infty}$ such that $\nabla\psi(Zw_t) = q_t$. (We do not try to make $\nabla\psi(Zw_t) = \nu_t$ or μ_t, since only q_t is constructed using a mirror descent / dual averaging update.) Let $w_0 := 0$, and for $t \geq 0$, let

$$w_{t+1} := w_t - \frac{\theta_t}{\lambda_t} Z^\top \nu_t. \tag{3.3}$$

We can verify that q_t is indeed the dual variable to w_t, in the sense that $\nabla\psi(Zw_t) = q_t$: this is true by definition at $t = 0$, since $\nabla\psi(Zw_0) = \nabla\psi(0) = q_0$. For $t \geq 0$, we have

$$\begin{aligned} q_{t+1} &\propto q_t \odot \exp\left(-\frac{\theta_t}{\lambda_t} Z Z^\top \nu_t\right) \\ &\propto \exp(Zw_t) \odot \exp\left(-\frac{\theta_t}{\lambda_t} Z Z^\top \nu_t\right) \\ &= \exp\left(Z\left(w_t - \frac{\theta_t}{\lambda_t} Z^\top \nu_t\right)\right) = \exp(Zw_{t+1}). \end{aligned}$$

In addition, we have the following characterization of w_t based on a momentum term, giving rise to the earlier eq. (1.1).

Lemma 3.4. *For all $\lambda_t, \theta_t \in (0, 1]$, if $\lambda_0 = 1$, then for all $t \geq 0$,*

$$w_{t+1} = w_t - \theta_t\left(g_t + Z^\top q_t\right),$$

where $g_0 := 0$, and for $t \geq 1$,

$$g_t := \frac{\lambda_{t-1}(1-\lambda_t)}{\lambda_t}\left(g_{t-1} + Z^\top q_t\right).$$

Specifically, for $\lambda_t = 2/(t+2)$, it holds that

$$\frac{\lambda_{t-1}(1-\lambda_t)}{\lambda_t} = \frac{t}{t+1}, \quad \text{and} \quad g_t = \sum_{j=1}^{t} \frac{j}{t+1} Z^\top q_j,$$

and $Z^\top \mu_t = 2g_t/t$.

Consequently, with $\lambda_t = 2/(t+2)$, the primal iterate defined by eq. (3.3) coincides with the iterate given by Algorithm 1 with $\beta_t = t/(t+1)$.

Additionally, Lemmas 3.2 and 3.4 already prove the second part of Theorem 3.1, since $\phi(\mu_t) = 4\|g_t\|_2^2/(2t^2)$ by Lemma 3.4, while $\phi(\bar{q}) = \bar{\gamma}^2/2$ by Lemma A.1.

3.3. Margin Analysis

Now we consider the margin maximization result of Theorem 3.1. The function ψ will be important here, since it gives a smoothed approximation of the margin: recall that $\psi(Zw)$ is defined as

$$\psi(Zw) = \ell^{-1}\left(\sum_{i=1}^{n} \ell\left(\langle z_i, w\rangle\right)\right).$$

Since ℓ is increasing, we have

$$\begin{aligned} -\psi(Zw) &\leq -\ell^{-1}\left(\max_{1\leq i\leq n} \ell\left(\langle z_i, w\rangle\right)\right) \\ &= -\ell^{-1}\left(\ell\left(\max_{1\leq i\leq n} \langle z_i, w\rangle\right)\right) \\ &= -\max_{1\leq i\leq n} \langle z_i, w\rangle. \end{aligned}$$

As a result, to prove a lower bound on $\gamma(w_t)$, we only need to prove a lower bound on $-\psi(Zw_t)/\|w_t\|_2$, and it would be enough if we have a lower bound on $-\psi(Zw_t)$ and an upper bound on $\|w_t\|_2$.

Below is our lower bound on $-\psi$ for Algorithm 1. Its proof is based on a much finer analysis of dual Nesterov, and uses both primal and dual smoothness.

Lemma 3.5. *Let $\theta_t = 1$ for all $t \geq 0$, and $\lambda_0 = 1$, then for all $t \geq 1$,*

$$\begin{aligned} -\psi(Zw_t) \geq\ & -\psi(Zw_0) + \frac{1}{2\lambda_{t-1}^2}\left\|Z^\top \mu_t\right\|_2^2 \\ &+ \sum_{j=1}^{t-1} \frac{1}{2}\left(\frac{1}{\lambda_{j-1}^2} - \frac{1-\lambda_j}{\lambda_j^2}\right)\left\|Z^\top \mu_j\right\|_2^2 \\ &+ \sum_{j=0}^{t-1} \frac{1}{2\lambda_j}\left\|Z^\top \nu_j\right\|_2^2. \end{aligned}$$

Additionally, here are our bounds on $\|w_t\|_2$.

Lemma 3.6. *Let $\theta_t = 1$ for all $t \geq 0$, then*

$$\sum_{j=0}^{t-1} \frac{\bar{\gamma}}{\lambda_j} \leq \|w_t\|_2 \leq \sum_{j=0}^{t-1} \frac{1}{\lambda_j}\left\|Z^\top \nu_j\right\|_2.$$

With Lemmas 3.5 and 3.6, we can prove Theorem 3.1. Here we show a weaker result which gives $1/t^2$ convergence to $\bar{\gamma}/2$; its proof is also part of the full proof of Theorem 3.1, but much simpler. The remaining proof of Theorem 3.1 is deferred to Appendix C.

Proposition 3.7 (weaker version of Theorem 3.1)**.** *With $\theta_t = 1$ and $\lambda_t = 2/(t+2)$, we have*

$$\gamma(w_t) \geq \frac{\bar{\gamma}}{2} - \frac{4\ln(n)}{\bar{\gamma}(t+1)^2}.$$

Proof. With $\lambda_t = 2/(t+2)$, it holds that

$$\frac{1}{\lambda_{j-1}^2} - \frac{1-\lambda_j}{\lambda_j^2} \geq 0,$$

therefore

$$-\psi(Zw_t) \geq -\psi(Zw_0) + \sum_{j=0}^{t-1} \frac{1}{2\lambda_j} \left\| Z^\top \nu_j \right\|_2^2. \tag{3.8}$$

Then eq. (3.8) and Lemma 3.6 imply

$$\frac{\psi(Zw_0) - \psi(Zw_t)}{\|w_t\|_2} \geq \frac{\sum_{j=0}^{t-1} \frac{1}{2\lambda_j} \left\| Z^\top \nu_j \right\|_2^2}{\sum_{j=0}^{t-1} \frac{1}{\lambda_j} \left\| Z^\top \nu_j \right\|_2} \geq \frac{\bar{\gamma}}{2}, \tag{3.9}$$

since $\left\| Z^\top \nu_j \right\|_2 \geq \bar{\gamma}$ (cf. Lemma A.1). On the other hand, Lemma 3.6 and $\lambda_t = 2/(t+2)$ imply

$$\|w_t\|_2 \geq \sum_{j=0}^{t-1} \frac{\bar{\gamma}}{\lambda_j} \geq \frac{\bar{\gamma}(t+1)^2}{4},$$

and thus

$$\frac{\psi(Zw_0)}{\|w_t\|_2} = \frac{\ln(n)}{\|w_t\|_2} \leq \frac{4\ln(n)}{\bar{\gamma}(t+1)^2}. \tag{3.10}$$

It then follows from eqs. (3.9) and (3.10) that

$$\gamma(w_t) \geq \frac{-\psi(Zw_t)}{\|w_t\|_2} \geq \frac{\bar{\gamma}}{2} - \frac{4\ln(n)}{\bar{\gamma}(t+1)^2}.$$

□

4. Analysis of Algorithm 2

Since Algorithm 1 is derived from dual Nesterov, it can also be run completely in the dual, meaning primal iterates and in particular the primal dimensionality never play a role. However, this dual version would require calculating $ZZ^\top q_t$, which in the kernel setting requires n^2 kernel calls. In Algorithm 2, we replace $Z^\top q_t$ with a single column z_{i_t} of $Z^\top$, where i_t is sampled from $q_t \in \Delta_n$. This sampling allows us to make only n kernel calls per iteration, rather than n^2 as in Algorithm 1.

Unfortunately, we do not have a general theory for Algorithm 2. Instead, as follows, we provide here an analysis with momentum disabled, meaning $\beta_t = 0$, and a small constant step size θ_t.

Theorem 4.1. *Given $\epsilon > 0$ and $\delta \in (0,1)$, let*

$$t = \max\left(\left\lceil \frac{32\ln(n) + 64\ln(2/\delta)}{\bar{\gamma}^2\epsilon^2} \right\rceil, \left\lceil \frac{32}{\delta\epsilon^2} \right\rceil \right),$$

and $\theta_j = \sqrt{\ln(n)/t}$ for $0 \leq j < t$, then with probability $1-\delta$,

$$\gamma(w_t) \geq \bar{\gamma} - \epsilon.$$

Algorithm 2

Input: data matrix $Z \in \mathbb{R}^{n\times d}$, step size $(\theta_t)_{t=0}^\infty$, momentum factor $(\beta_t)_{t=0}^\infty$.
Initialize: $w_0 = g_{-1} = (0,\ldots,0) \in \mathbb{R}^d$, $q_0 = (\frac{1}{n},\ldots,\frac{1}{n}) \in \Delta_n$.
for $t = 0,1,2,\ldots$ **do**
  Sample $i_t \sim q_t$.
  $g_t \leftarrow \beta_t\,(g_{t-1} + z_{i_t})$.
  $w_{t+1} \leftarrow w_t - \theta_t\,(g_t + z_{i_t})$.
  $q_{t+1} \propto \exp(Zw_{t+1})$, and $q_{t+1} \in \Delta_n$.
end for

The proof of Theorem 4.1 is similar to the proof of Theorem 3.1, but must additionally produce high-probability bounds on $-\psi(Zw_t)$ and $\|w_t\|_2$; details are deferred to Appendix D.

Although we do not have a convergence analysis for Algorithm 2 with a nonzero momentum, it works well in practice, as verified on the full `mnist` data, shown in Figure 3. Still with $\beta_t = t/(t+1)$, Algorithm 2 can slightly beat the batch perceptron method, which is the fastest prior algorithm in the hard-margin kernel SVM setting. (Other classical methods, such as stochastic dual coordinate ascent (Shalev-Shwartz & Zhang, 2013), are focused on the nonseparable soft-margin SVM setting.)

5. Other Exponentially-Tailed Losses

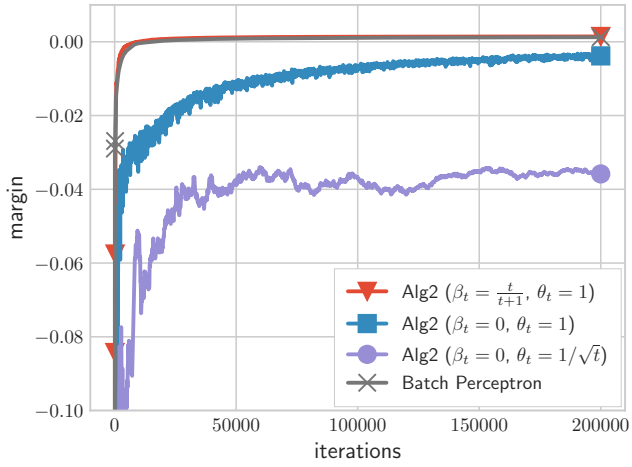
Here we discuss the extension to other exponentially-tailed losses, such as the logistic loss in the case of binary classification, and to multiclass losses.

5.1. Binary Classification

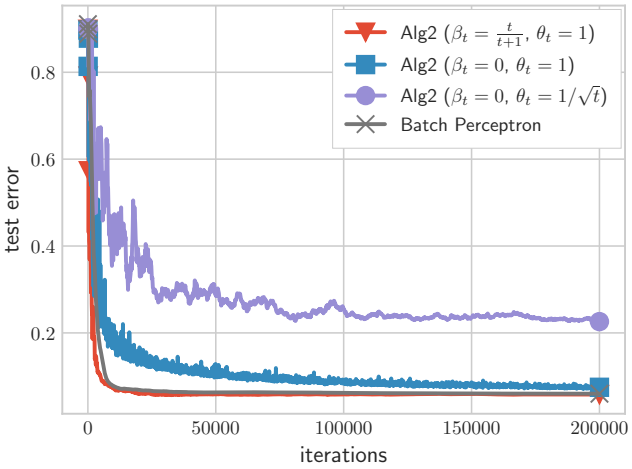
In previous sections, we focused on the exponential loss. Our methods can also be applied to other strictly decreasing losses, such as the logistic loss $\ell(z) := \ln(1+e^z)$, simply by replacing $Z^\top q_t$ in Algorithm 1 with $Z^\top \nabla\psi(Zw_t)$, where ψ is still defined by eq. (2.1).

In the proof of Theorem 3.1, we only use two properties of ψ: (i) ψ is ρ-smooth with respect to the ℓ_∞ norm, and (ii) $\|\nabla\psi\|_1 \geq 1$. These two properties hold with $\rho = 1$ for the exponential loss, and with $\rho = n$ for the logistic loss (Ji & Telgarsky, 2019, Lemma 5.3, Lemma D.1). Therefore we can use the same analysis to prove a $\widetilde{\mathcal{O}}(1/t^2)$ margin maximization rate for the logistic loss; details are given in Appendix C.

However, the margin rate would additionally depend on ρ, which is n for the logistic loss. Such a bad dependency on n is probably due to the aggressive initial step size: from eq. (2.2), we know that $\nabla\psi(Zw_t)$ is just $\nabla\mathcal{R}(w_t)$ normalized by $\ell'\left(\psi(Zw)\right)/n$. However, this quantity is at most



(a) Margins.



(b) Test error.

Figure 3. Margin maximization performance of various methods requiring $\mathcal{O}(n)$ kernel evaluations per iteration. The batch perceptron is slightly beaten by Algorithm 2 using the momentum and step size parameters from Algorithm 1, which is only provided here as a heuristic. By contrast, the theoretically-justified parameters, as analyzed in Theorem 4.1, are slower than batch perceptron. The data here is the full `mnist` data, with features given by the initial kernel of a 2-homogeneous network of width 128 (cf. Appendix F).

$1/n$ for the logistic loss, even at initialization. It is an interesting open problem to find a better initial step size.

5.2. Multiclass Classification

Suppose now that inputs $(x_i)_{i=1}^N$ have multiclass labels $(c_i)_{i=1}^N$, meaning $c_i \in \{1, \dots, k\}$. The standard approach to multiclass linear prediction associates a linear predictor u_j for each class $j \in \{1, \dots, k\}$; collecting these as columns of a matrix $U \in \mathbb{R}^{d \times k}$, the multiclass prediction is

$$x \mapsto \operatorname*{arg\,max}_{c \in \{1,\dots,k\}} x^\top U \boldsymbol{e}_c,$$

and letting $\|U\|_{\mathrm{F}}$ denote the Frobenius norm, the margin of U and maximum margin are respectively

$$\gamma_{\mathrm{m}}(U) := \frac{\min_i \min_{c \neq c_i} \left(x^\top U \boldsymbol{e}_{c_i} - x^\top U \boldsymbol{e}_c\right)}{\|U\|_{\mathrm{F}}},$$
$$\bar{\gamma}_{\mathrm{m}} := \max_{\|U\|_{\mathrm{F}} \leq 1} \gamma_{\mathrm{m}}(U),$$

with edge case $\gamma_{\mathrm{m}}(0) = 0$ as before.

We now show how to reduce this case to the binary case and allow the application of Algorithm 1 and its analysis in Theorem 3.1. The standard construction of multiclass losses uses exactly the differences of labels as in the preceding definition of γ_{m} (Zhang, 2005; Tewari & Bartlett, 2007); that is, define a multiclass risk as

$$\mathcal{R}_{\mathrm{m}}(U) = \frac{1}{N} \sum_{i=1}^N \sum_{j \neq c_i} \ell\left(x_i^\top U \boldsymbol{e}_j - x_i^\top U \boldsymbol{e}_{c_i}\right).$$

To rewrite this in our notation as a prediction problem defined by a single matrix Z, define $n := N(k-1)$, let $F : \mathbb{R}^{d \times k} \to \mathbb{R}^{dk}$ be any fixed flattening of a $d \times k$ matrix into a vector of length dk, and let $\pi : \{1, \dots, N\} \times \{1, \dots, k-1\} \to \{1, \dots, n\}$ be any bijection between the N original examples and their n new fake counterparts defined as follows: for each example i and incorrect label $j \neq c_i$, define $z_{\pi(i,j)} := x_i(\boldsymbol{e}_{c_i} - \boldsymbol{e}_j)^\top / \sqrt{2}$, and let $Z \in \mathbb{R}^{n \times dk}$ be the matrix where row $\pi(i,j)$ is the flattening $F(z_{\pi(i,j)})^\top$; then, equivalently,

$$\frac{1}{k-1} \mathcal{R}_{\mathrm{m}}(U) = \frac{1}{n} \sum_{i=1}^n \ell\left(F(U)^\top F(z_{\pi(i,j)})\right).$$

In particular, it suffices to consider a flattened weight vector $w = F(U) \in \mathbb{R}^{dk}$, and invoke the algorithm and analysis from Section 3, with the preceding matrix Z.

Theorem 5.1. *Let a multiclass problem $\{(x_i, c_i)\}_{i=1}^N$ be given with maximum multiclass margin $\bar{\gamma}_{\mathrm{m}} > 0$. Then the corresponding matrix Z as defined above has binary margin $\bar{\gamma} := \bar{\gamma}_{\mathrm{m}}/\sqrt{2} > 0$. Moreover, letting w_t denote the output of Algorithm 1 when run on this Z as in Theorem 3.1, meaning exponential loss ℓ and $\beta_t := t/(t+1)$ and $\theta_t := 1$, for every $t \geq 1$ the un-flattened output $U_t := F^{-1}(w_t)$ satisfies*

$$\gamma_{\mathrm{m}}(U_t) \geq \bar{\gamma}_{\mathrm{m}} - \frac{4\left(1 + \ln(n)\right)\left(1 + 2\ln(t+1)\right)}{\bar{\gamma}_{\mathrm{m}}(t+1)^2}.$$

Due to proceeding by reduction, the guarantees of Section 4 also hold for an analogous multiclass version of Algorithm 2.

Indeed, Algorithm 2, with the aggressive (heuristic) parameters $\beta_t = t/(t+1)$ and $\theta_t = 1$ proved effective in practice, and was used in the experiments of Figure 3, as well as the upcoming Figure 4.

One issue that arises in these reduction-based implementations is avoiding explicitly writing down Z or even individual rows of Z, which have dk elements. Instead, note that sampling from q as in Algorithm 2 now returns both an example index i, as well as an incorrect label $j \neq c_i$. From here, updates to just the two columns of U corresponding to j and c_i can be constructed.

6. Application: Deep Network Kernel Evolution

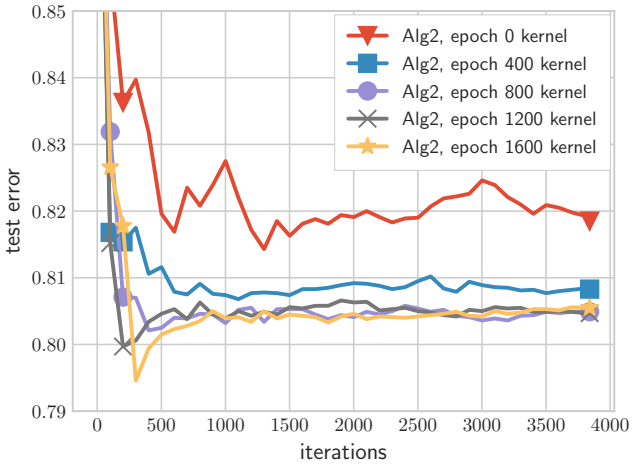
As an application of these fast margin-maximization methods, we study the evolution of kernels encountered during deep network training. Specifically, consider the `cifar10` dataset, which has 50,000 input images in 10 classes; a standard deep network architecture for this problem is the AlexNet (Krizhevsky et al., 2012), which has both convolutional, dense linear, and various nonlinear layers.

Let v_t denote the AlexNet parameters encountered at epoch t of training on `cifar10` with a standard stochastic gradient method, and let $A(x; v_t)$ denote the prediction of AlexNet on input x with these parameters v_t. From here, we can obtain a feature vector $\nabla_v A(x; v_t)$, and use it to construct a matrix Z to plug in to our methods; when $t = 0$, this corresponds to the Neural Tangent Kernel (NTK) (Jacot et al., 2018; Li & Liang, 2018; Du et al., 2018), but here we are also interested in later kernels, meaning $t > 0$. (To handle multiclass output, we simply flatten the Jacobian; as another technical point, we ℓ_2-normalize the features to further simplify training and the selection of step sizes.)

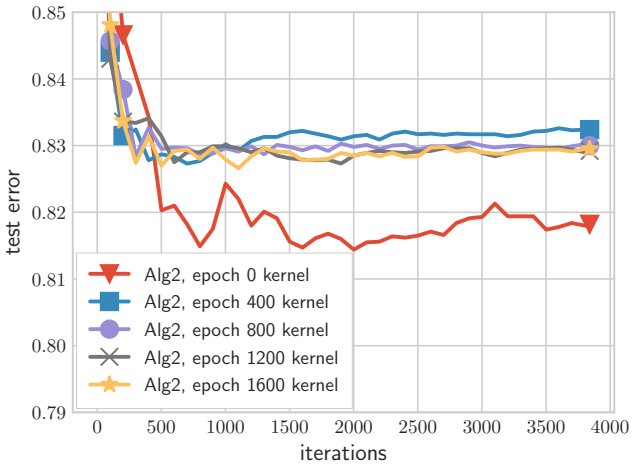
For any fixed t, we thus obtain a linear prediction problem with rows of matrix Z given by the features $\nabla_v A(x; v_t)$ (with additional care for class labels, as in the reductions defined in Section 5.2), and can use Algorithm 2 to quickly determine the maximum margin. Figure 4(a) presents an experiment that is consistent with standard beliefs: as t increases, the test error of the corresponding maximum-margin (kernel) predictor decreases. In these experiments, the AlexNet training is run until the features converge, and the test error of the final maximum-margin kernel predictor is identical to that of the final deep network.

A more interesting example is given in Figure 4(b): a case where feature learning does not help. All that differs between Figure 4(a) and Figure 4(b) is the choice of random seed.

A key point is that the AlexNet in both experiments was trained with only 128 training points (the testing set had the usual 10,000 images, but test error is unsurprisingly large). The idea is that the feature learning implicit in deep network training can overfit with such small amounts of data.



(a) Random seed 100.



(b) Random seed 13579.

Figure 4. Test error curves of kernel predictors trained with Algorithm 2, using kernels from different epochs of standard deep network training. Please see Section 6 and appendix F for details; the short summary is that changing the random seed suffices to change whether kernel features improve or not.

Of course, 128 examples is not a standard deep learning regime; these figures merely illustrate that feature learning may fail, not that it always fails. It is an interesting open question to study this phenomenon in realistic scenarios.

7. Concluding Remarks and Open Problems

In this work, we gave two new algorithms based on a dual perspective of margin maximization and implicit bias: a momentum-based method in Section 3 constructed via translating dual Nesterov acceleration iterates into the primal,

and an adaptive sampling method in Section 4 which aims for greater per-iteration efficiency in the kernel case.

Turning first to Algorithm 1, its derivation exposes a connection between Nesterov acceleration in the dual and momentum in the primal. Does this connection exist more generally, namely in other optimization problems?

A second open problem is to formally analyze Algorithm 2 with momentum. As demonstrated empirically in Figure 3, it can work well, whereas our analysis disables momentum.

On the empirical side, the small-scale experiments of Section 6 scratched the surface of situations where feature learning can fail. Can this phenomenon be exhibited in more realistic scenarios?

Acknowledgements

We thank the reviewers for their comments. ZJ and MT are grateful for support from the NSF under grant IIS-1750051, and from NVIDIA under a GPU grant.

References

Allen-Zhu, Z. and Orecchia, L. Linear coupling: An ultimate unification of gradient and mirror descent. *arXiv preprint arXiv:1407.1537*, 2014.

Bartlett, P. L., Foster, D. J., and Telgarsky, M. J. Spectrally-normalized margin bounds for neural networks. In *Advances in Neural Information Processing Systems*, pp. 6240–6249, 2017.

Borwein, J. and Lewis, A. *Convex Analysis and Nonlinear Optimization*. Springer Publishing Company, Incorporated, 2000.

Chizat, L. and Bach, F. Implicit bias of gradient descent for wide two-layer neural networks trained with the logistic loss. *arXiv preprint arXiv:2002.04486*, 2020.

Clarkson, K. L., Hazan, E., and Woodruff, D. P. Sublinear optimization for machine learning. *Journal of the ACM (JACM)*, 59(5):1–49, 2012.

Cotter, A., Shalev-Shwartz, S., and Srebro, N. The kernelized stochastic batch perceptron. In *ICML*, 2012.

Du, S. S., Zhai, X., Poczos, B., and Singh, A. Gradient descent provably optimizes over-parameterized neural networks. *arXiv preprint arXiv:1810.02054*, 2018.

Freund, Y. and Schapire, R. E. A decision-theoretic generalization of on-line learning and an application to boosting. *J. Comput. Syst. Sci.*, 55(1):119–139, 1997.

Ghadimi, E., Feyzmahdavian, H. R., and Johansson, M. Global convergence of the heavy-ball method for convex optimization. In *2015 European control conference (ECC)*, pp. 310–315. IEEE, 2015.

Gunasekar, S., Lee, J., Soudry, D., and Srebro, N. Characterizing implicit bias in terms of optimization geometry. *arXiv preprint arXiv:1802.08246*, 2018a.

Gunasekar, S., Lee, J. D., Soudry, D., and Srebro, N. Implicit bias of gradient descent on linear convolutional networks. In *Advances in Neural Information Processing Systems*, pp. 9461–9471, 2018b.

Hanashiro, R. and Abernethy, J. Linear separation via optimism. *arXiv preprint arXiv:2011.08797*, 2020.

Jacot, A., Gabriel, F., and Hongler, C. Neural tangent kernel: Convergence and generalization in neural networks. In *Advances in neural information processing systems*, pp. 8571–8580, 2018.

Ji, Z. and Telgarsky, M. Risk and parameter convergence of logistic regression. *arXiv preprint arXiv:1803.07300v2*, 2018.

Ji, Z. and Telgarsky, M. Characterizing the implicit bias via a primal-dual analysis. *arXiv preprint arXiv:1906.04540*, 2019.

Ji, Z. and Telgarsky, M. Directional convergence and alignment in deep learning. *arXiv preprint arXiv:2006.06657*, 2020.

Krizhevsky, A., Sutskever, I., and Hinton, G. Imagenet classification with deep convolutional neural networks. In *NIPS*, 2012.

Li, Y. and Liang, Y. Learning overparameterized neural networks via stochastic gradient descent on structured data. In *Advances in Neural Information Processing Systems*, pp. 8157–8166, 2018.

Lyu, K. and Li, J. Gradient descent maximizes the margin of homogeneous neural networks. *arXiv preprint arXiv:1906.05890*, 2019.

Nacson, M. S., Lee, J., Gunasekar, S., Srebro, N., and Soudry, D. Convergence of gradient descent on separable data. *arXiv preprint arXiv:1803.01905*, 2018.

Nesterov, Y. A method of solving a convex programming problem with convergence rate $\mathcal{O}(1/k^2)$. *Soviet Mathematics Doklady*, 27(2):372–376, 1983.

Nesterov, Y. *Introductory Lectures on Convex Optimization: A Basic Course*. Kluwer Academic Publishers, 2004.

Neyshabur, B., Tomioka, R., and Srebro, N. In search of the real inductive bias: On the role of implicit regularization in deep learning. arXiv:1412.6614 [cs.LG], 2014.

Shalev-Shwartz, S. and Zhang, T. Stochastic dual coordinate ascent methods for regularized loss minimization. *Journal of Machine Learning Research*, 14(2), 2013.

Shalev-Shwartz, S. et al. Online learning and online convex optimization. *Foundations and trends in Machine Learning*, 4(2):107–194, 2011.

Soudry, D., Hoffer, E., Nacson, M. S., Gunasekar, S., and Srebro, N. The implicit bias of gradient descent on separable data. *arXiv preprint arXiv:1710.10345*, 2017.

Telgarsky, M. Margins, shrinkage, and boosting. In *ICML*, 2013.

Tewari, A. and Bartlett, P. L. On the consistency of multiclass classification methods. *JMLR*, 8:1007–1025, 2007.

Tseng, P. On accelerated proximal gradient methods for convex-concave optimization. http://www.mit.edu/~dimitrib/PTseng/papers/apgm.pdf, 2008.

Zhang, C., Bengio, S., Hardt, M., Recht, B., and Vinyals, O. Understanding deep learning requires rethinking generalization. *arXiv preprint arXiv:1611.03530*, 2016.

Zhang, T. Statistical analysis of some multi-category large margin classification methods. *JMLR*, 5:1225–1251, 2005.