The Dynamics of Learning: A Random Matrix Approach

Anonymous Authors[1]

Abstract

Understanding the learning dynamics of neural networks is one of the key issues for the improvement of optimization algorithms as well as for the theoretical comprehension of why deep neural nets work so well today. In this paper, we introduce a random matrix-based framework to analyze the learning dynamics of a single-layer linear network on a binary classification problem, for data of simultaneously large dimension and size, trained by gradient descent. Our results provide rich insights into common questions in neural nets, such as overfitting, early stopping and the initialization of training, thereby opening the door for future studies of more elaborate structures and models appearing in today's neural networks.

1. Introduction

Deep neural networks trained with back-propagation have commonly attained superhuman performance in applications of computer vision (Krizhevsky et al., 2012) and many others (Schmidhuber, 2015) and are thus receiving an unprecedented research interest. Despite the rapid growth of the list of successful applications with these gradient-based methods, our theoretical understanding, however, is progressing at a more modest pace.

One of the salient features of deep networks today is that they often have far more model parameters than the number of training samples that they are trained on, but meanwhile some of the models still exhibit remarkably good generalization performance when applied to unseen data of similar nature, while others generalize poorly in exactly the same setting. A satisfying explanation of this phenomenon would be the key to more powerful and reliable network structures.

To answer such a question, statistical learning theory has proposed interpretations from the viewpoint of system complexity (Vapnik, 2013; Bartlett & Mendelson, 2002; Poggio et al., 2004). In the case of large numbers of parameters, it is suggested to apply some form of regularization to ensure good generalization performance. Regularizations can be explicit, such as the dropout technique (Srivastava et al., 2014) or the l_2-penalization (weight decay) as reported in (Krizhevsky et al., 2012); or implicit, as in the case of the early stopping strategy (Yao et al., 2007) or the stochastic gradient descent algorithm itself (Zhang et al., 2016).

Inspired by the recent line of works (Saxe et al., 2013; Advani & Saxe, 2017), in this article we introduce a random matrix framework to the analysis of training and, more importantly, generalization performance of large neural networks, trained by gradient descent. Preliminary results established from a toy model of two-class classification on a single-layer linear network are presented, which, despite their simplicity, shed new light on the understanding of many important aspects in training neural nets. In particular, we demonstrate how early stopping can naturally protect the network against overfitting, which becomes more severe as the number of training sample approaches the dimension of the data. We also provide a strict lower bound on the training sample size for a given classification task in this simple setting. A byproduct of our analysis implies that random initialization, although commonly used in practice in training deep networks (Glorot & Bengio, 2010; Krizhevsky et al., 2012), may lead to a degradation of the network performance.

From a more theoretical point of view, our analyses allow one to evaluate any functional of the eigenvalues of the sample covariance matrix of the data (or of the data representation learned from previous layers in a deep model), which is at the core of understanding many experimental observations in today's deep networks (Glorot & Bengio, 2010; Ioffe & Szegedy, 2015). Our results are envisioned to generalize to more elaborate settings, notably to deeper models that are trained with the stochastic gradient descent algorithm, which is of more practical interest today due to the tremendous size of the data.

Notations: Boldface lowercase (uppercase) characters stand for vectors (matrices), and non-boldface for scalars respectively. $\mathbf{0}_p$ is the column vector of zeros of size p, and $\mathbf{I}_p$ the $p \times p$ identity matrix. The notation $(\cdot)^\mathsf{T}$ denotes the

[1]Anonymous Institution, Anonymous City, Anonymous Region, Anonymous Country. Correspondence to: Anonymous Author <anon.email@domain.com>.

Preliminary work. Under review by the International Conference on Machine Learning (ICML). Do not distribute.

transpose operator. The norm $\|\cdot\|$ is the Euclidean norm for vectors and the operator norm for matrices. $\Im(\cdot)$ denotes the imaginary part of a complex number. For $x \in \mathbb{R}$, we denote for simplicity $(x)^+ \equiv \max(x, 0)$.

In the remainder of the article, we introduce the problem of interest and recall the results of (Saxe et al., 2013) in Section 2. After a brief overview of basic concepts and methods to be used throughout the article in Section 3, our main results on the training and generalization performance of the network are presented in Section 4, followed by a thorough discussion in Section 5 and experiments on the popular MNIST database (LeCun et al., 1998) in Section 6. Section 7 concludes the article by summarizing the main results and outlining future research directions.

2. Problem Statement

Let the training data $\mathbf{x}_1, \ldots, \mathbf{x}_n \in \mathbb{R}^p$ be independent vectors drawn from two distribution classes $\mathcal{C}_1$ and $\mathcal{C}_2$ of cardinality n_1 and n_2 (thus $n_1 + n_2 = n$), respectively. We assume that the data vector $\mathbf{x}_i$ of class $\mathcal{C}_a$ can be written as

$$\begin{cases} \mathbf{x}_i = -\boldsymbol{\mu} + \mathbf{z}_i, & a = 1 \\ \mathbf{x}_i = \boldsymbol{\mu} + \mathbf{z}_i, & a = 2 \end{cases}$$

with $\boldsymbol{\mu} \in \mathbb{R}^p$ and $\mathbf{z}_i$ a Gaussian random vector $\mathbf{z}_i \sim \mathcal{N}(\mathbf{0}_p, \mathbf{I}_p)$. In the context of a binary classification problem, one takes the label $y_i = -1$ for $\mathbf{x}_i \in \mathcal{C}_1$ and $y_j = 1$ for $\mathbf{x}_j \in \mathcal{C}_2$ to distinguish the two classes.

We denote the training data matrix $\mathbf{X} = \left[\mathbf{x}_1, \ldots, \mathbf{x}_n\right] \in \mathbb{R}^{p \times n}$ by cascading all $\mathbf{x}_i$'s as column vectors and associated label vector $\mathbf{y} \in \mathbb{R}^n$. With the pair $\{\mathbf{X}, \mathbf{y}\}$, a classifier is trained using "full-batch" gradient descent to minimize the loss function $L(\mathbf{w})$ given by

$$L(\mathbf{w}) = \frac{1}{2n}\|\mathbf{y}^\mathsf{T} - \mathbf{w}^\mathsf{T}\mathbf{X}\|^2$$

such that for a new datum $\hat{\mathbf{x}}$, the output of the classifier is $\hat{y} = \mathbf{w}^\mathsf{T}\hat{\mathbf{x}}$, where we use the sign of $\hat{y}$ as a predictor of the class of $\hat{\mathbf{x}}$. The derivative of $L(\mathbf{w})$ with respective to $\mathbf{w}$ is given by

$$\frac{\partial L(\mathbf{w})}{\partial \mathbf{w}} = -\frac{1}{n}\mathbf{X}(\mathbf{y} - \mathbf{X}^\mathsf{T}\mathbf{w}).$$

The gradient descent algorithm (Boyd & Vandenberghe, 2004) takes steps of size α (often considered to be a small constant) to the *negative* of the gradient of the loss function, i.e.,

$$\mathbf{w}_{t+1} = \mathbf{w}_t - \alpha \left.\frac{\partial L(\mathbf{w})}{\partial \mathbf{w}}\right|_{\mathbf{w}=\mathbf{w}_t}.$$

Following the idea in (Saxe et al., 2013; Advani & Saxe, 2017), when the learning rate α is small, the two points $\mathbf{w}_{t+1}$ and $\mathbf{w}_t$ are close to each other so that by performing a continuous-time approximation, one obtains the following differential equation

$$\frac{\partial \mathbf{w}(t)}{\partial t} = -\alpha\frac{\partial L(\mathbf{w})}{\partial \mathbf{w}} = \frac{\alpha}{n}\mathbf{X}\left(\mathbf{y} - \mathbf{X}^\mathsf{T}\mathbf{w}(t)\right)$$

the solution of which is given explicitly by

$$\mathbf{w}(t) = e^{-\frac{\alpha t}{n}\mathbf{X}\mathbf{X}^\mathsf{T}}\mathbf{w}_0 + \left(\mathbf{I}_p - e^{-\frac{\alpha t}{n}\mathbf{X}\mathbf{X}^\mathsf{T}}\right)(\mathbf{X}\mathbf{X}^\mathsf{T})^{-1}\mathbf{X}\mathbf{y} \tag{1}$$

if one assumes that $\mathbf{X}\mathbf{X}^\mathsf{T}$ is invertible (only possible in the case $p < n$), with $\mathbf{w}_0 \equiv \mathbf{w}(t = 0)$ the initialization of the weight vector; we recall the definition of the exponential of a matrix $\frac{1}{n}\mathbf{X}\mathbf{X}^\mathsf{T}$ given by the power series

$$e^{\frac{1}{n}\mathbf{X}\mathbf{X}^\mathsf{T}} = \sum_{k=0}^{\infty}\frac{1}{k!}(\frac{1}{n}\mathbf{X}\mathbf{X}^\mathsf{T})^k = \mathbf{V}e^{\boldsymbol{\Lambda}}\mathbf{V}^\mathsf{T}$$

with the eigendecomposition of $\frac{1}{n}\mathbf{X}\mathbf{X}^\mathsf{T} = \mathbf{V}\boldsymbol{\Lambda}\mathbf{V}^\mathsf{T}$ and $e^{\boldsymbol{\Lambda}}$ is a diagonal matrix with elements equal to the exponential of the elements of $\boldsymbol{\Lambda}$. As $t \to \infty$ the network "forgets" the initialization $\mathbf{w}_0$ and results in the least-square solution $\mathbf{w}_{LS} \equiv (\mathbf{X}\mathbf{X}^\mathsf{T})^{-1}\mathbf{X}\mathbf{y}$.

When $p > n$, the matrix $\mathbf{X}\mathbf{X}^\mathsf{T}$is no longer invertible. Assuming $\mathbf{X}^\mathsf{T}\mathbf{X}$ is invertible and writing $\mathbf{X}\mathbf{y} = \left(\mathbf{X}\mathbf{X}^\mathsf{T}\right)\mathbf{X}\left(\mathbf{X}^\mathsf{T}\mathbf{X}\right)^{-1}\mathbf{y}$, the solution is similarly given by

$$\mathbf{w}(t) = e^{-\frac{\alpha t}{n}\mathbf{X}\mathbf{X}^\mathsf{T}}\mathbf{w}_0 + \mathbf{X}\left(\mathbf{I}_n - e^{-\frac{\alpha t}{n}\mathbf{X}^\mathsf{T}\mathbf{X}}\right)(\mathbf{X}^\mathsf{T}\mathbf{X})^{-1}\mathbf{y}$$

with the least-square solution $\mathbf{w}_{LS} \equiv \mathbf{X}(\mathbf{X}^\mathsf{T}\mathbf{X})^{-1}\mathbf{y}$.

In the work of (Advani & Saxe, 2017) it has been assumed that $\mathbf{X}$ has i.i.d. entries and the target (label) $\mathbf{y}$ is independent of $\mathbf{X}$ so as to simplify the analysis. Here, we position ourselves in a more realistic setting where $\mathbf{X}$ and $\mathbf{y}$ are statistically correlated, and therefore our results would be of more guiding significance for practical interests.

From (1) note that both $e^{-\frac{\alpha t}{n}\mathbf{X}\mathbf{X}^\mathsf{T}}$ and $\mathbf{I}_p - e^{-\frac{\alpha t}{n}\mathbf{X}\mathbf{X}^\mathsf{T}}$ share the same eigenvectors with the *sample covariance matrix* $\frac{1}{n}\mathbf{X}\mathbf{X}^\mathsf{T}$, which thus plays a pivotal role in the network learning dynamics. More concretely, the projections of $\mathbf{w}_0$ and $\mathbf{w}_{LS}$ onto the eigenspace of $\frac{1}{n}\mathbf{X}\mathbf{X}^\mathsf{T}$, weighted by functions ($\exp(-\alpha t\lambda_i)$ or $1 - \exp(-\alpha t\lambda_i)$) of the associated eigenvalue λ_i, give the temporal evolution of $\mathbf{w}(t)$. The core of our study therefore consists in deeply understanding of the eigenpairs of the sample covariance matrix, which has been largely investigated in the random matrix literature (Bai & Silverstein, 2010).

3. Preliminaries

Throughout this paper, we will be relying on some basic yet powerful concepts and methods from random matrix theory, which shall be briefly highlighted in this section.

3.1. Resolvent and deterministic equivalents

Consider an $n \times n$ Hermitian random matrix $\mathbf{M}$. We define its *resolvent* $\mathbf{Q}_{\mathbf{M}}(z)$, for all $z \in \mathbb{C}$ not an eigenvalue of $\mathbf{M}$, as

$$\mathbf{Q}_{\mathbf{M}}(z) = (\mathbf{M} - z\mathbf{I}_n)^{-1}.$$

Through the Cauchy integral formula discussed in the following subsection, as well as its central importance in random matrix theory, $\mathbf{Q}_{\frac{1}{n}\mathbf{X}\mathbf{X}^\mathsf{T}}(z)$ is the key object investigated in this article.

For certain simple distributions of $\mathbf{M}$, one may define a so-called *deterministic equivalent* $\bar{\mathbf{Q}}_{\mathbf{M}}$ for $\mathbf{Q}_{\mathbf{M}}$, which is a deterministic matrix such that for all $\mathbf{A} \in \mathbb{R}^{n\times n}$ and all $\mathbf{a}, \mathbf{b} \in \mathbb{R}^n$ of bounded (spectral and Euclidean, respectively) norms, $\frac{1}{n}\operatorname{tr}(\mathbf{A}\mathbf{Q}_{\mathbf{M}}) - \frac{1}{n}\operatorname{tr}\left(\mathbf{A}\bar{\mathbf{Q}}_{\mathbf{M}}\right) \to 0$ and $\mathbf{a}^\mathsf{T}\left(\mathbf{Q}_{\mathbf{M}} - \bar{\mathbf{Q}}_{\mathbf{M}}\right)\mathbf{b} \to 0$ almost surely as $n \to \infty$. As such, deterministic equivalents allow to transfer random spectral properties of $\mathbf{M}$ in the form of deterministic limiting quantities and thus allows for a more detailed investigation.

3.2. Cauchy's integral formula

First note that the resolvent $\mathbf{Q}_{\mathbf{M}}$ has the same eigenspace as $\mathbf{M}$, with associated eigenvalue λ_i replaced by $\frac{1}{\lambda_i - z}$. Our objective is to evaluate functions of these eigenvalues, which reminds us of the fundamental Cauchy's integral formula, stating that for any function f homomorphic on an open subset U of the complex plane, one can compute $f(\lambda)$ by contour integration. More concretely, for a closed positively (counter-clockwise) oriented path γ in U with winding number one (i.e., describing a 360° rotation), one has, for λ contained in the surface described by γ,

$$\frac{1}{2\pi i}\oint_\gamma \frac{f(z)}{z-\lambda}dz = f(\lambda)$$

and $\frac{1}{2\pi i}\oint_\gamma \frac{f(z)}{z-\lambda}dz = 0$ if λ is outside the contour of γ.

With Cauchy's integral formula, one is able to evaluate more sophisticated functionals of the random matrix $\mathbf{M}$. For example, for $f(\mathbf{M}) \equiv \mathbf{a}^\mathsf{T} e^{\mathbf{M}}\mathbf{b}$ one has

$$f(\mathbf{M}) = -\frac{1}{2\pi i}\oint_\gamma \exp(z)\mathbf{a}^\mathsf{T}\mathbf{Q}_{\mathbf{M}}(z)\mathbf{b}\, dz$$

with γ a positively oriented path circling around *all* the eigenvalues of $\mathbf{M}$. Moreover, from the previous subsection one knows that the bilinear form $\mathbf{a}^\mathsf{T}\mathbf{Q}_{\mathbf{M}}(z)\mathbf{b}$ is asymptotically close to a non-random quantity $\mathbf{a}^\mathsf{T}\bar{\mathbf{Q}}_{\mathbf{M}}(z)\mathbf{b}$. One thus deduces that the functional $\mathbf{a}^\mathsf{T}e^{\mathbf{M}}\mathbf{b}$ has an asymptotically deterministic behavior that can be expressed as $-\frac{1}{2\pi i}\oint_\gamma \exp(z)\mathbf{a}^\mathsf{T}\bar{\mathbf{Q}}_{\mathbf{M}}(z)\mathbf{b}\, dz$.

This observation serves in the present article as the foundation for the performance analysis of the gradient-based classifier, as described in the following section.

4. Temporal Evolution of Training and Generalization Performance

With the explicit expression of $\mathbf{w}(t)$ in (1), we now turn our attention to the training and generalization performances of the classifier as a function of the training time t. To this end, we shall be working under the following assumptions.

Assumption 1 (Growth rate)**.** *As $n \to \infty$,*

1. $\frac{p}{n} \to c \in (0, \infty)$
2. *for* $a = 1, 2$, $\frac{n_a}{n} \to c_a \in (0, 1)$
3. $\|\boldsymbol{\mu}\| = O(1)$.

The above assumption ensures that the matrix $\frac{1}{n}\mathbf{X}\mathbf{X}^\mathsf{T}$ is of bounded operator norm for all large n, p with probability one (Bai & Silverstein, 1998).

Assumption 2 (Random initialization)**.** *We let $\mathbf{w}_0 \equiv \mathbf{w}(0)$ be a random vector with i.i.d. entries of zero mean, variance σ^2/p for some $\sigma > 0$ and finite fourth moment.*

We first focus on the generalization performance, i.e., the average performance of the trained classifier taking as input an unseen new datum $\hat{\mathbf{x}}$ drawn from class $\mathcal{C}_1$ or $\mathcal{C}_2$.

4.1. Generalization Performance

To evaluate the generalization performance of the classifier, we are interested in two types of misclassification rates, for a new datum $\hat{\mathbf{x}}$ given by

$$\mathrm{P}(\mathbf{w}(t)^\mathsf{T}\hat{\mathbf{x}} > 0 \mid \hat{\mathbf{x}} \in \mathcal{C}_1), \quad \mathrm{P}(\mathbf{w}(t)^\mathsf{T}\hat{\mathbf{x}} < 0 \mid \hat{\mathbf{x}} \in \mathcal{C}_2).$$

Since $\hat{\mathbf{x}}$ is independent of $\mathbf{w}(t)$, the scalar $\mathbf{w}(t)^\mathsf{T}\hat{\mathbf{x}}$ is a Gaussian random variable of mean $\pm\mathbf{w}(t)^\mathsf{T}\boldsymbol{\mu}$ and of variance $\|\mathbf{w}(t)\|^2$ and the above probabilities can be expressed through the Q-function: $Q(x) \equiv \frac{1}{\sqrt{2\pi}}\int_x^\infty \exp\left(-\frac{u^2}{2}\right)du$. We thus resort to the computation of $\mathbf{w}(t)^\mathsf{T}\boldsymbol{\mu}$ as well as $\mathbf{w}(t)^\mathsf{T}\mathbf{w}(t)$ to evaluate the classification error.

For $\boldsymbol{\mu}^\mathsf{T}\mathbf{w}(t)$, with Cauchy's integral formula we have

$$\begin{aligned}\boldsymbol{\mu}^\mathsf{T}\mathbf{w}(t) &= \boldsymbol{\mu}^\mathsf{T}e^{-\frac{\alpha t}{n}\mathbf{X}\mathbf{X}^\mathsf{T}}\mathbf{w}_0 + \boldsymbol{\mu}^\mathsf{T}\left(\mathbf{I}_p - e^{-\frac{\alpha t}{n}\mathbf{X}\mathbf{X}^\mathsf{T}}\right)\mathbf{w}_{LS} \\ &= -\frac{1}{2\pi i}\oint_\gamma f_t(z)\boldsymbol{\mu}^\mathsf{T}\left(\frac{1}{n}\mathbf{X}\mathbf{X}^\mathsf{T} - z\mathbf{I}_p\right)^{-1}\mathbf{w}_0\, dz \\ &\quad - \frac{1}{2\pi i}\oint_\gamma \frac{1-f_t(z)}{z}\boldsymbol{\mu}^\mathsf{T}\left(\frac{1}{n}\mathbf{X}\mathbf{X}^\mathsf{T} - z\mathbf{I}_p\right)^{-1}\frac{1}{n}\mathbf{X}\mathbf{y}\, dz\end{aligned}$$

with $f_t(z) \equiv \exp(-\alpha t z)$, for a positive closed path γ circling around all eigenvalues of $\frac{1}{n}\mathbf{X}\mathbf{X}^\mathsf{T}$. Note that the data matrix $\mathbf{X}$ can be written as

$$\mathbf{X} = -\boldsymbol{\mu}\mathbf{j}_1^\mathsf{T} + \boldsymbol{\mu}\mathbf{j}_2^\mathsf{T} + \mathbf{Z} = \boldsymbol{\mu}\mathbf{y}^\mathsf{T} + \mathbf{Z}$$

with $\mathbf{Z} \equiv \left[\mathbf{z}_1, \ldots, \mathbf{z}_n\right] \in \mathbb{R}^{p\times n}$ of i.i.d. $\mathcal{N}(0,1)$ entries and $\mathbf{j}_a \in \mathbb{R}^n$ the canonical vectors of class $\mathcal{C}_a$ defined as $(\mathbf{j}_a)_i = \delta_{\mathbf{x}_i \in \mathcal{C}_a}$. To isolate the deterministic vectors $\boldsymbol{\mu}$ and $\mathbf{j}_a$'s from the random $\mathbf{Z}$ in $\boldsymbol{\mu}^\mathsf{T}\mathbf{w}(t)$, we exploit Woodbury's identity to obtain

$$\left(\frac{1}{n}\mathbf{X}\mathbf{X}^\mathsf{T} - z\mathbf{I}_p\right)^{-1} = \mathbf{Q}(z) - \mathbf{Q}(z)\left[\boldsymbol{\mu} \quad \tfrac{1}{n}\mathbf{Z}\mathbf{y}\right]$$
$$\begin{bmatrix} \boldsymbol{\mu}^\mathsf{T}\mathbf{Q}(z)\boldsymbol{\mu} & 1 + \frac{1}{n}\boldsymbol{\mu}^\mathsf{T}\mathbf{Q}(z)\mathbf{Z}\mathbf{y} \\ * & -1 + \frac{1}{n}\mathbf{y}^\mathsf{T}\mathbf{Z}^\mathsf{T}\mathbf{Q}(z)\frac{1}{n}\mathbf{Z}\mathbf{y} \end{bmatrix}^{-1} \begin{bmatrix} \boldsymbol{\mu}^\mathsf{T} \\ \frac{1}{n}\mathbf{y}^\mathsf{T}\mathbf{Z}^\mathsf{T} \end{bmatrix} \mathbf{Q}(z)$$

where we denote the resolvent $\mathbf{Q}(z) \equiv \left(\frac{1}{n}\mathbf{Z}\mathbf{Z}^\mathsf{T} - z\mathbf{I}_p\right)^{-1}$, a deterministic equivalent of which is given by

$$\mathbf{Q}(z) \leftrightarrow \bar{\mathbf{Q}}(z) \equiv m(z)\mathbf{I}_p$$

with $m(z)$ determined by the popular Marčenko–Pastur equation (Marčenko & Pastur, 1967)

$$m(z) = \frac{1-c-z}{2cz} + \frac{\sqrt{(1-c-z)^2 - 4cz}}{2cz} \tag{2}$$

where the branch of the square root is selected in such a way that $\Im(z)\cdot\Im m(z) > 0$.

Substituting $\mathbf{Q}(z)$ by the simple form deterministic equivalent $m(z)\mathbf{I}_p$ in the expression of $\boldsymbol{\mu}^\mathsf{T}\mathbf{w}(t)$, we are thus able to estimate the random variable $\boldsymbol{\mu}^\mathsf{T}\mathbf{w}(t)$ with a contour integral of some deterministic quantities as $n, p \to \infty$. Similar arguments also hold for $\mathbf{w}(t)^\mathsf{T}\mathbf{w}(t)$, together leading to the following theorem.

Theorem 1 (Generalization performance)**.** *Let Assumptions 1 and 2 hold. As $n \to \infty$, with probability one*

$$\mathrm{P}(\mathbf{w}(t)^\mathsf{T}\hat{\mathbf{x}} > 0 \mid \hat{\mathbf{x}} \in \mathcal{C}_1) - Q\left(\frac{-E}{\sqrt{V}}\right) \to 0$$
$$\mathrm{P}(\mathbf{w}(t)^\mathsf{T}\hat{\mathbf{x}} < 0 \mid \hat{\mathbf{x}} \in \mathcal{C}_2) - Q\left(\frac{E}{\sqrt{V}}\right) \to 0$$

where

$$E \equiv -\frac{1}{2\pi i}\oint_\gamma \frac{1-f_t(z)}{z}\frac{\|\boldsymbol{\mu}\|^2 m(z)\, dz}{(\|\boldsymbol{\mu}\|^2 + c)\, m(z) + 1}$$
$$V \equiv \frac{1}{2\pi i}\oint_\gamma \left(\frac{\frac{1}{z^2}\left(1-f_t(z)\right)^2}{(\|\boldsymbol{\mu}\|^2 + c)\, m(z) + 1} - \sigma^2 f_t^2(z) m(z)\right) dz$$

with γ a closed positively oriented path that contains all eigenvalues of $\frac{1}{n}\mathbf{X}\mathbf{X}^\mathsf{T}$ and the origin, $f_t(z) \equiv \exp(-\alpha t z)$ and $m(z)$ given by Equation (2).

Although derived from the case $p < n$, Theorem 1 also applies when $p > n$. Note that in this case, once Cauchy's integral formula is applied, for $z \neq 0$ not an eigenvalue of $\frac{1}{n}\mathbf{X}\mathbf{X}^\mathsf{T}$ (thus not of $\frac{1}{n}\mathbf{X}^\mathsf{T}\mathbf{X}$), one has

$$\mathbf{X}\left(\frac{1}{n}\mathbf{X}^\mathsf{T}\mathbf{X} - z\mathbf{I}_n\right)^{-1}\mathbf{y} = \left(\frac{1}{n}\mathbf{X}\mathbf{X}^\mathsf{T} - z\mathbf{I}_p\right)^{-1}\mathbf{X}\mathbf{y}$$

which leads to the same expressions as in Theorem 1. Since $\frac{1}{n}\mathbf{X}\mathbf{X}^\mathsf{T}$ and $\frac{1}{n}\mathbf{X}^\mathsf{T}\mathbf{X}$ have the same eigenvalues except for additional zero eigenvalues for the larger matrix, the path γ remains unchanged and hence Theorem 1 holds true for both $p < n$ and $p > n$. The case $p = n$ can be obtained by continuity arguments.

4.2. Training performance

To compare generalization versus training performance, we are now interested in the behavior of the classifier when applied to the training set $\mathbf{X}$. To this end, we shall consider the random vector $\mathbf{X}^\mathsf{T}\mathbf{w}(t)$ given by

$$\mathbf{X}^\mathsf{T}\mathbf{w}(t) = \mathbf{X}^\mathsf{T}e^{-\frac{\alpha t}{n}\mathbf{X}\mathbf{X}^\mathsf{T}}\mathbf{w}_0 + \mathbf{X}^\mathsf{T}\left(\mathbf{I}_p - e^{-\frac{\alpha t}{n}\mathbf{X}\mathbf{X}^\mathsf{T}}\right)\mathbf{w}_{LS}.$$

Note that the i-th entry of $\mathbf{X}^\mathsf{T}\mathbf{w}(t)$ is given by the bilinear form $\mathbf{e}_i^\mathsf{T}\mathbf{X}^\mathsf{T}\mathbf{w}(t)$, with $\mathbf{e}_i$ the canonical vector with unique non-zero entry $[\mathbf{e}_i]_i = 1$. With previous notations we have

$$\begin{aligned}&\mathbf{e}_i^\mathsf{T}\mathbf{X}^\mathsf{T}\mathbf{w}(t) \\ &= -\frac{1}{2\pi i}\oint_\gamma f_t(z,t)\mathbf{e}_i^\mathsf{T}\mathbf{X}^\mathsf{T}\left(\frac{1}{n}\mathbf{X}\mathbf{X}^\mathsf{T} - z\mathbf{I}_p\right)^{-1}\mathbf{w}_0\, dz \\ &- \frac{1}{2\pi i}\oint_\gamma \frac{1-f_t(z)}{z}\mathbf{e}_i^\mathsf{T}\frac{1}{n}\mathbf{X}^\mathsf{T}\left(\frac{1}{n}\mathbf{X}\mathbf{X}^\mathsf{T} - z\mathbf{I}_p\right)^{-1}\mathbf{X}\mathbf{y}\, dz\end{aligned}$$

which yields the following results on the training performance.

Theorem 2 (Training performance)**.** *Under the assumptions and notations of Theorem 1, as $n \to \infty$,*

$$\mathrm{P}(\mathbf{w}(t)^\mathsf{T}\mathbf{x}_i > 0 \mid \mathbf{x}_i \in \mathcal{C}_1) - Q\left(\frac{-E_*}{\sqrt{V_* - E_*^2}}\right) \to 0$$
$$\mathrm{P}(\mathbf{w}(t)^\mathsf{T}\mathbf{x}_i < 0 \mid \mathbf{x}_i \in \mathcal{C}_2) - Q\left(\frac{E_*}{\sqrt{V_* - E_*^2}}\right) \to 0$$

almost surely, with

$$E_* \equiv \frac{1}{2\pi i}\oint_\gamma \frac{1-f_t(z)}{z}\frac{dz}{(\|\boldsymbol{\mu}\|^2 + c)\, m(z) + 1}$$
$$V_* \equiv \frac{1}{2\pi i}\oint_\gamma \left(\frac{\frac{1}{z}\left(1-f_t(z)\right)^2}{(\|\boldsymbol{\mu}\|^2 + c)\, m(z) + 1} - \sigma^2 f_t^2(z) z m(z)\right) dz$$

In Figure 1 we compare finite dimensional simulations with theoretical Simulation results obtained from Theorem 1 and 2 and observe a very close match, already for not too large n, p. As t grows large, the generalization error first drops rapidly with the training error, then goes up, although slightly, while the training error continues to decrease to zero. This is because the classifier starts to over-fit the training data $\mathbf{X}$ and thus performs badly for unseen ones. To

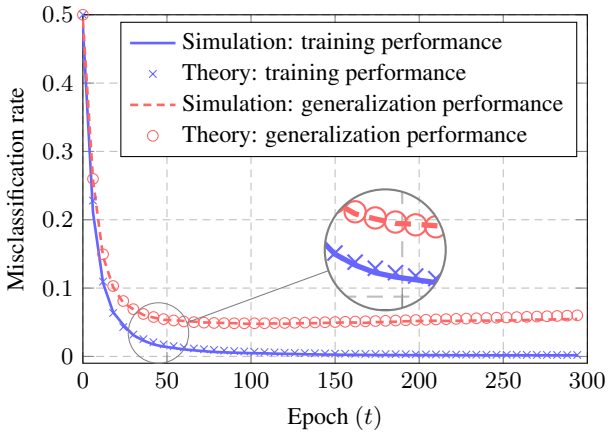

Figure 1. Training and generalization performance for $\boldsymbol{\mu} = [2; \mathbf{0}_{p-1}]$, $p = 256$, $n = 512$, $\sigma^2 = 0.1$, $\alpha = 0.01$ and $c_1 = c_2 = 1/2$. Simulation results obtained by averaging over 50 runs.

avoid over-fitting, one effectual approach is to apply regularization strategies (Bishop, 2007), for example, to "early stop" (at $t = 100$ for instance in the setting of Figure 1) in the training process. However, this introduces new hyper-parameters such as the optimal stopping time t_{opt} that is of crucial importance for the network performance and is often tuned through cross-validation in practice. Theorem 1 and 2 tell us that the training and generalization performances, although being random themselves, have asymptotically deterministic behaviors given by (E_*, V_*) and (E, V), respectively, which allows for a deeper comprehension into the choice of t_{opt}, since E, V are in fact functions of t via $f_t(z) \equiv \exp(-\alpha t z)$.

Nonetheless, the expressions in Theorem 1 and 2 of contour integrations are not easily analyzable nor interpretable. To gain more insight, we shall rewrite (E, V) and (E_*, V_*) in a more readable way. First, note from Figure 2 that the matrix $\frac{1}{n}\mathbf{X}\mathbf{X}^\mathsf{T}$ has (possibly) two types of eigenvalues: those inside the *main bulk* (between $\lambda_- \equiv (1-\sqrt{c})^2$ and $\lambda_+ \equiv (1+\sqrt{c})^2$) of the Marčenko–Pastur distribution

$$\nu(dx) = \frac{\sqrt{(x-\lambda_-)^+(\lambda_+ - x)^+}}{2\pi c x} dx + \left(1 - \frac{1}{c}\right)^+ \delta(x) \tag{3}$$

and a (possibly) isolated one that lies away from $[\lambda_-, \lambda_+]$, that shall be treated separately. We rewrite the path γ (that contains all eigenvalues of $\frac{1}{n}\mathbf{X}\mathbf{X}^\mathsf{T}$) as the sum of two paths γ_b and γ_s, that circle around the main bulk and the isolated eigenvalue (if any), respectively. To handle the first integral of γ_b, we use the fact that for any nonzero $\lambda \in \mathbb{R}$, the limit

$$\lim_{z \in \mathbb{Z} \to \lambda} m(z) \equiv \check{m}(\lambda)$$

exists (Silverstein & Choi, 1995) and follow the idea in (Bai & Silverstein, 2008) by choosing the contour γ_b to be a rectangle with sides parallel to the axes, intersecting the real axis at 0 and λ_+ and the horizontal sides being a distance $\varepsilon \to 0$ away from the real axis, to split the contour integral into four single ones of $\check{m}(x)$. The second integral circling around γ_s can be computed with the residue theorem. This together leads to the expressions of (E, V) and (E_*, V_*) as follows

$$E = \int \frac{1 - f_t(x)}{x} \mu(dx) \tag{4}$$

$$V = \frac{\|\boldsymbol{\mu}\|^2 + c}{\|\boldsymbol{\mu}\|^2} \int \frac{(1 - f_t(x))^2 \mu(dx)}{x^2} + \sigma^2 \int f_t^2(x) \nu(dx) \tag{5}$$

$$E_* = \frac{\|\boldsymbol{\mu}\|^2 + c}{\|\boldsymbol{\mu}\|^2} \int \frac{1 - f_t(x)}{x} \mu(dx) \tag{6}$$

$$V_* = \frac{\|\boldsymbol{\mu}\|^2 + c}{\|\boldsymbol{\mu}\|^2} \int \frac{(1 - f_t(x))^2 \mu(dx)}{x} + \sigma^2 \int x f_t^2(x) \nu(dx) \tag{7}$$

where we recall $f_t(x) = \exp(-\alpha t x)$, $\nu(x)$ given by (3) and denote the measure

$$\mu(dx) \equiv \frac{\sqrt{(x-\lambda_-)^+(\lambda_+ - x)^+}}{2\pi(\lambda_s - x)} dx + \frac{(\|\boldsymbol{\mu}\|^4 - c)^+}{\|\boldsymbol{\mu}\|^2} \delta_{\lambda_s}(x) \tag{8}$$

as well as

$$\lambda_s = c + 1 + \|\boldsymbol{\mu}\|^2 + \frac{c}{\|\boldsymbol{\mu}\|^2} \geq (\sqrt{c} + 1)^2$$

with equality if and only if $\|\boldsymbol{\mu}\|^2 = \sqrt{c}$.

A first remark on the expressions of (4)-(7) is that E_* differs from E only by a factor of $\frac{\|\boldsymbol{\mu}\|^2 + c}{\|\boldsymbol{\mu}\|^2}$. Also, both V and V_* are the sum of two parts: the first part that strongly depends on $\boldsymbol{\mu}$ and the second one that is independent of $\boldsymbol{\mu}$. One thus deduces for $\|\boldsymbol{\mu}\| \to 0$ that $E \to 0$ and

$$V \to \int \frac{(1 - f_t(x))^2}{x^2} \rho(dx) + \sigma^2 \int f_t^2(x) \nu(dx) > 0$$

with $\rho(dx) \equiv \frac{\sqrt{(x-\lambda_-)^+(\lambda_+ - x)^+}}{2\pi(c+1)} dx$ and therefore the generalization performance goes to $Q(0) = 0.5$. On the other hand, for $\|\boldsymbol{\mu}\| \to \infty$, one has $\frac{E}{\sqrt{V}} \to \infty$ and hence the classifier makes perfect predictions.

In a more general context (i.e., for Gaussian mixture models with generic means and covariances, and obviously for practical datasets), there may be more than one eigenvalue of $\frac{1}{n}\mathbf{X}\mathbf{X}^\mathsf{T}$ lying outside the main bulk, which may not be limited to the interval $[\lambda_-, \lambda_+]$. The expression of $m(z)$, instead of being explicitly given by (2), may be determined through more elaborate (often implicit) formulations. Our analysis scheme can be easily extended to handle these more intricate cases.

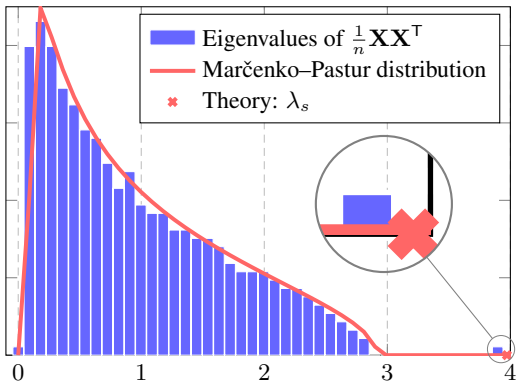

Figure 2. Eigenvalue distribution of $\frac{1}{n}\mathbf{X}\mathbf{X}^\mathsf{T}$ for $\boldsymbol{\mu} = [1.5; \mathbf{0}_{p-1}]$, $p = 512$, $n = 1\,024$ and $c_1 = c_2 = 1/2$.

5. Discussions

In this section, with a careful inspection of (4) and (5), discussions will be made from several different aspects. First of all, recall that the generalization performance is simply given by $Q\left(\frac{\boldsymbol{\mu}^\mathsf{T}\mathbf{w}(t)}{\|\mathbf{w}(t)\|}\right)$, with the term $\frac{\boldsymbol{\mu}^\mathsf{T}\mathbf{w}(t)}{\|\mathbf{w}(t)\|}$ describing the alignment between $\mathbf{w}(t)$ and $\boldsymbol{\mu}$, therefore the best possible generalization performance can be expressed as $Q(\|\boldsymbol{\mu}\|)$.

Although the integrals in (4) and (5) do not have nice closed forms, note that, for t close to 0, with a Taylor expansion of $f_t(x) \equiv \exp(-\alpha t x)$ around $\alpha t x = 0$, one gets more interpretable forms of E and V without integrals, as presented in the following subsection.

5.1. Approximation for t close to 0

Taking $t = 0$, one has $f_t(x) = 1$ and thus

$$E = 0, \quad V = \sigma^2 \int \nu(dx) = \sigma^2$$

with $\nu(x)$ the Marčenko–Pastur distribution given by (3). As a consequence, one deduces that when $t \to 0$, the generalization performance goes to $Q(0) = 0.5$ for $\sigma^2 \neq 0$ and the classifier is simply making random guesses.

For t not equal but close to 0, the Taylor expansion of $f_t(x) \equiv \exp(-\alpha t x)$ around $\alpha t x = 0$ gives

$$f_t(x) \equiv \exp(-\alpha t x) \approx 1 - \alpha t x + O(\alpha^2 t^2 x^2).$$

Making the substitution $x = 1 + c - 2\sqrt{c}\cos\theta$ and with the following relation (Gradshteyn & Ryzhik, 2014)

$$\int_0^\pi \frac{\sin^2\theta}{p + q\cos\theta} d\theta = \frac{p\pi}{q^2}\left(1 - \sqrt{1 - \frac{q^2}{p^2}}\right)$$

one gets $E = \tilde{E} + O(\alpha^2 t^2)$ and $V = \tilde{V} + O(\alpha^2 t^2)$, where

$$\begin{aligned}
\tilde{E} &\equiv \frac{\alpha t}{2} g(\boldsymbol{\mu}, c) + \frac{(\|\boldsymbol{\mu}\|^4 - c)^+}{\|\boldsymbol{\mu}\|^2} \alpha t = \|\boldsymbol{\mu}\|^2 \alpha t \\
\tilde{V} &\equiv \frac{\|\boldsymbol{\mu}\|^2 + c}{\|\boldsymbol{\mu}\|^2} \frac{(\|\boldsymbol{\mu}\|^4 - c)^+}{\|\boldsymbol{\mu}\|^2} \alpha^2 t^2 + \frac{\|\boldsymbol{\mu}\|^2 + c}{\|\boldsymbol{\mu}\|^2} \frac{\alpha^2 t^2}{2} g(\boldsymbol{\mu}, c) \\
&\quad + \sigma^2(1 + c)\alpha^2 t^2 - 2\sigma^2 \alpha t + \left(1 - \frac{1}{c}\right)^+ \sigma^2 \\
&\quad + \frac{\sigma^2}{2c}\left(1 + c - (1 + \sqrt{c})|1 - \sqrt{c}|\right) \\
&= (\|\boldsymbol{\mu}\|^2 + c + c\sigma^2)\alpha^2 t^2 + \sigma^2(\alpha t - 1)^2
\end{aligned}$$

with $g(\boldsymbol{\mu}, c) \equiv \|\boldsymbol{\mu}\|^2 + \frac{c}{\|\boldsymbol{\mu}\|^2} - \left(\|\boldsymbol{\mu}\| + \frac{\sqrt{c}}{\|\boldsymbol{\mu}\|}\right)\left|\|\boldsymbol{\mu}\| - \frac{\sqrt{c}}{\|\boldsymbol{\mu}\|}\right|$ and consequently $\frac{1}{2}g(\boldsymbol{\mu}, c) + \frac{(\|\boldsymbol{\mu}\|^4 - c)^+}{\|\boldsymbol{\mu}\|^2} = \|\boldsymbol{\mu}\|^2$. It is interesting to note from the above calculation that, although E and V seem to have different behaviors[1] for $\|\boldsymbol{\mu}\|^2 > \sqrt{c}$ or $c > 1$, it is in fact not the case and the extra part of $\|\boldsymbol{\mu}\|^2 > \sqrt{c}$ (or $c > 1$) compensates for the singularity of the integral, so that the performance is a smooth function of both $\|\boldsymbol{\mu}\|^2$ and c.

Taking the derivative of $\frac{\tilde{E}}{\sqrt{\tilde{V}}}$ with respect to t, one has

$$\frac{\partial}{\partial t}\frac{\tilde{E}}{\sqrt{\tilde{V}}} = \frac{\alpha(1 - \alpha t)\sigma^2}{\tilde{V}^{3/2}}$$

which implies that the maximum of $\frac{\tilde{E}}{\sqrt{\tilde{V}}}$ is $\frac{\|\boldsymbol{\mu}\|^2}{\sqrt{\|\boldsymbol{\mu}\|^2 + c + c\sigma^2}}$ and can be attained with $t = \frac{1}{\alpha}$. Moreover, taking $t = 0$ in the above equation one gets $\frac{\partial}{\partial t}\frac{\tilde{E}}{\sqrt{\tilde{V}}}\Big|_{t=0} = \frac{\alpha}{\sigma}$. Therefore, large σ is harmful to the training efficiency.

The approximation error arising from Taylor expansion can be large for t away from 0, for example, at $t = \frac{1}{\alpha}$ the difference between E and $\tilde{E}$ is of order $O(1)$ and cannot be neglected. Nonetheless, some insights from the above calculations contribute to the understanding of the object of interest $\frac{E}{\sqrt{V}}$, as presented in the following remark.

Remark 1 (Optimal generalization performance)**.** *Note from (8) that $\int \mu(dx) = \|\boldsymbol{\mu}\|^2$, one has, with Cauchy–Schwarz inequality*

$$\begin{aligned}
E^2 &\leq \int \left(\frac{1 - f_t(x)}{x}\right)^2 d\mu(x) \cdot \int d\mu(x) \\
&\leq \frac{\|\boldsymbol{\mu}\|^4}{\|\boldsymbol{\mu}\|^2 + c} V
\end{aligned}$$

with the second equality holds if and only if $\sigma^2 = 0$. One thus concludes that $\frac{E}{\sqrt{V}} \leq \frac{\|\boldsymbol{\mu}\|^2}{\sqrt{\|\boldsymbol{\mu}\|^2 + c}}$ and the best generalization performance (lowest misclassification rates for

[1] This phenomenon has been largely observed in random matrix theory and is referred to as "phase transition" (Baik et al., 2005).

unseen data) is given by $Q(\frac{\|\boldsymbol{\mu}\|^2}{\sqrt{\|\boldsymbol{\mu}\|^2+c}})$ *and can be attained only when* $\sigma^2 = 0$.

The above remark is of particular interest because, for a given task (thus $p, \boldsymbol{\mu}$ fixed) it allows one to compute the *minimum* training data number n to fulfill a certain request of classification accuracy.

As a side remark, note that in the expression of $\frac{E}{\sqrt{V}}$, σ^2 only appears in the denominator, meaning that random initializations impair the generalization performance of the network. As such, one should initialize with σ^2 very close, but not equal, to zero, to obtain symmetry breaking between hidden units (Goodfellow et al., 2016) and at the same time to mitigate the drop of performance due to large σ^2.

In Figure 3 we plot the optimal generalization performance and the optimal stopping time as functions of σ^2, showing that small initialization helps training in terms of both accuracy and efficiency.

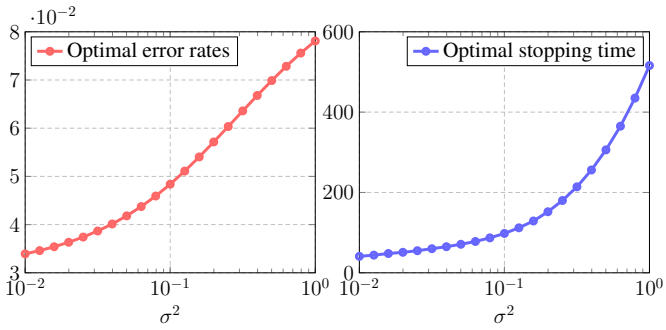

Figure 3. Optimal performance and stopping time as functions of σ^2, with $c = 1/2$, $\|\boldsymbol{\mu}\|^2 = 4$ and $\alpha = 0.01$.

5.2. As $t \to \infty$: least-squares solution

As $t \to \infty$, one has $f_t(x) \to 0$ which results in the least-square solution $\mathbf{w}_{LS} = (\mathbf{X}\mathbf{X}^\mathsf{T})^{-1}\mathbf{X}\mathbf{y}$ or $\mathbf{w}_{LS} = \mathbf{X}(\mathbf{X}^\mathsf{T}\mathbf{X})^{-1}\mathbf{y}$ and

$$\frac{\boldsymbol{\mu}^\mathsf{T}\mathbf{w}_{LS}}{\|\mathbf{w}_{LS}\|} = \frac{\|\boldsymbol{\mu}\|^2}{\sqrt{\|\boldsymbol{\mu}\|^2 + c}}\sqrt{1 - \min\left(c, \frac{1}{c}\right)}. \tag{9}$$

Comparing (9) with the expression in Remark 1, one observes that when $t \to \infty$ the network becomes "over-trained" and the performance drops by a factor of $\sqrt{1 - \min\left(c, \frac{1}{c}\right)}$. This becomes worse when c gets close to 1, as is consistent with the empirical findings in (Advani & Saxe, 2017). However, the point $c = 1$ is a singularity for (9), but not for $\frac{E}{\sqrt{V}}$ given by (4) and (5). One may thus expect to have a smooth and reliable behavior of the well-trained network for c close to 1, which is a noticeable advantage of gradient-based training compared to simple least-square method.

In Figure 4 we plot the generalization performance from simulation (blue line), the theoretical optimal performance in Remark 1 (cyan line), the approximation from Taylor expansion of $f_t(x)$ (red dashed line), together with the performance of $\mathbf{w}_{LS}$ (purple dashed line). One observes a close match between the Taylor expansion and the true performance for t close to 0, with the former being optimal at $t = 100$ and the latter slowly approaching the performance of $\mathbf{w}_{LS}$ as t goes to infinity.

In Figure 5 we underline the case $c = 1$ by taking $p = n = 512$ with all other parameters unchanged from Figure 4. One observes that the simulation curve (blue line) increases much faster compared to Figure 4 and is supposed to end up at 0.5, which is the performance of $\mathbf{w}_{LS}$ (purple dashed line). This confirms a serious degradation of performance for c close to 1 of the classical least-squares solution.

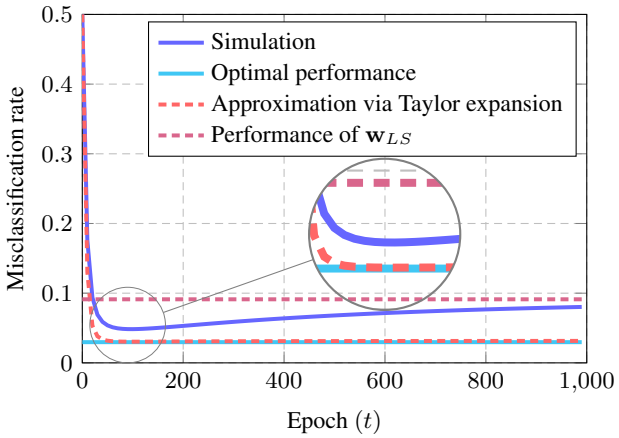

Figure 4. Generalization performance for $\boldsymbol{\mu} = [2; \mathbf{0}_{p-1}]$, $p = 256$, $n = 512$, $c_1 = c_2 = 1/2$, $\sigma^2 = 0.1$ and $\alpha = 0.01$. Simulation results obtained by averaging over 50 runs.

5.3. Special case for $c = 0$

One major interest of the random matrix analysis is that the ratio c appears constantly in the analysis. $c = 0$ indicates the fact that we have far more training data than their dimension. This results in both $\lambda_-, \lambda_+ \to 1$, $\lambda_s \to 1 + \|\boldsymbol{\mu}\|^2$ and

$$E \to \|\boldsymbol{\mu}\|^2 \frac{1 - f_t(1 + \|\boldsymbol{\mu}\|^2)}{1 + \|\boldsymbol{\mu}\|^2}$$
$$V \to \|\boldsymbol{\mu}\|^2 \left(\frac{1 - f_t(1 + \|\boldsymbol{\mu}\|^2)}{1 + \|\boldsymbol{\mu}\|^2}\right)^2 + \sigma^2 f_t^2(1).$$

As a consequence, $\frac{E}{\sqrt{V}} \to \|\boldsymbol{\mu}\|$ if $\sigma^2 = 0$. This can be explained by the fact that with sufficient training data the classifier learns to align perfectly to $\boldsymbol{\mu}$ so that $\frac{\boldsymbol{\mu}^\mathsf{T}\mathbf{w}(t)}{\|\mathbf{w}(t)\|} = \|\boldsymbol{\mu}\|$.

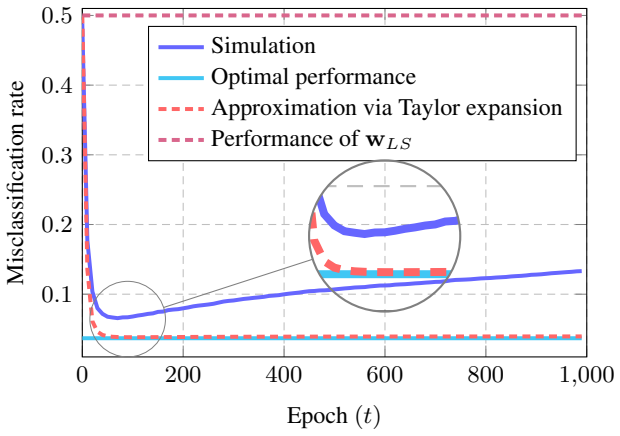

Figure 5. Generalization performance for $\boldsymbol{\mu} = [2; \mathbf{0}_{p-1}]$, $p = 512$, $n = 512$, $c_1 = c_2 = 1/2$, $\sigma^2 = 0.1$ and $\alpha = 0.01$. Simulation results obtained by averaging over 50 runs.

On the other hand, with initialization $\sigma^2 \neq 0$, one always has $\frac{E}{\sqrt{V}} < \|\boldsymbol{\mu}\|$. But still, as t goes large, the network forgets the initialization exponentially fast and converges to the optimal $\mathbf{w}(t)$ that aligns to $\boldsymbol{\mu}$.

For $\sigma^2 \neq 0$, we are interested in the optimal stopping time and shall thus take the derivative with respect to t,

$$\frac{\partial}{\partial t}\frac{E}{\sqrt{V}} = \frac{\alpha\sigma^2\|\boldsymbol{\mu}\|^2}{V^{3/2}} \frac{\|\boldsymbol{\mu}\|^2 f_t(1+\|\boldsymbol{\mu}\|^2)+1}{1+\|\boldsymbol{\mu}\|^2} f_t^2(1) > 0$$

showing that when $c = 0$, the generalization performance continues to increase as t grows and there is no "over-training" in this case.

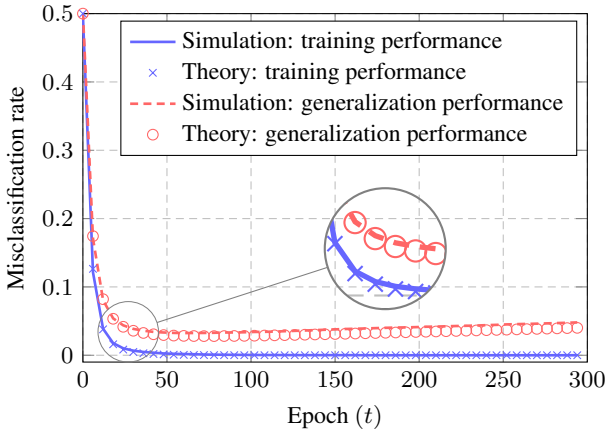

Figure 6. Training and generalization performance for MNIST data (number 1 and 7) with $n = p = 784$, $c_1 = c_2 = 1/2$, $\alpha = 0.01$ and $\sigma^2 = 0.1$. Simulation results obtained by averaging over 100 runs.

6. Numerical Validations

We close this article with experiments on the popular MNIST dataset (LeCun et al., 1998) (number 1 and 7). We randomly select training sets of size $n = 784$ vectorized images of dimension $p = 784$ and add artificially a Gaussian white noise of -10dB in order to be more compliant with our toy model setting. Empirical means and covariances of each class are estimated from the full set of 13 007 MNIST images (6 742 images of number 1 and 6 265 of number 7). The image vectors in each class are whitened by pre-multiplying $\mathbf{C}_a^{-1/2}$ and re-centered to have means of $\pm\boldsymbol{\mu}$, with $\boldsymbol{\mu}$ half of the difference between means from the two classes. We observe an extremely close fit between our results and the empirical simulations, as shown in Figure 6.

7. Conclusion

In this article, we established a random matrix approach to the analysis and understanding of learning dynamics for gradient-based algorithms applied to data of simultaneously large dimension and size. With a simple toy model of two classes of Gaussian data with $\pm\boldsymbol{\mu}$ means and identity covariance, we have shown that the training and generalization performances of a single-layer linear network have asymptotically deterministic behaviors that can be evaluated via the tools of deterministic equivalent and computed with complex contour integrals (and even under the form of simple real integrals in the present setting). This result can be taken as a first step into the analysis of more elaborate network structures or data model.

In this article, the analysis has been performed on the "full-batch" gradient descent system. However, the most popular method used today is in fact its "stochastic" version (Bottou, 2010) where only a fixed-size (n_{batch}) randomly selected subset (called a *mini-batch*) of the training data is used to compute the gradient and descend *one* step along with the opposite direction of this gradient in each iteration. In this scenario, one of major concern in practice lies in determining the optimal size of the mini-batch and its influence on the generalization performance of the network (Keskar et al., 2016), that can be naturally linked to the ratio n_{batch}/p in the random matrix analysis.

Deep networks that are of more practical interests, however, need more efforts. As mentioned in (Saxe et al., 2013; Advani & Saxe, 2017), in the case of multi-layer networks, the learning dynamics depend, instead of each eigenmode itself separately, on the coupling of different eigenmodes from different layers. To handle this difficulty, one may add extra assumptions of independence between layers as in (Choromanska et al., 2015) so that each layer can be studied separately and then reassembled to retrieve the results of the whole network.

References

Advani, Madhu S and Saxe, Andrew M. High-dimensional dynamics of generalization error in neural networks. *arXiv preprint arXiv:1710.03667*, 2017.

Bai, Zhi-Dong and Silverstein, Jack W. No eigenvalues outside the support of the limiting spectral distribution of large-dimensional sample covariance matrices. *Annals of probability*, pp. 316–345, 1998.

Bai, Zhidong and Silverstein, Jack W. *Spectral analysis of large dimensional random matrices*, volume 20. Springer, 2010.

Bai, Zhidong D and Silverstein, Jack W. CLT for linear spectral statistics of large-dimensional sample covariance matrices. In *Advances In Statistics*, pp. 281–333. World Scientific, 2008.

Baik, Jinho, Arous, Gérard Ben, Péché, Sandrine, et al. Phase transition of the largest eigenvalue for nonnull complex sample covariance matrices. *The Annals of Probability*, 33(5):1643–1697, 2005.

Bartlett, Peter L and Mendelson, Shahar. Rademacher and gaussian complexities: Risk bounds and structural results. *Journal of Machine Learning Research*, 3(Nov):463–482, 2002.

Bishop, Christopher M. *Pattern Recognition and Machine Learning*. Springer, 2007.

Bottou, Léon. Large-scale machine learning with stochastic gradient descent. In *Proceedings of COMPSTAT'2010*, pp. 177–186. Springer, 2010.

Boyd, Stephen and Vandenberghe, Lieven. *Convex optimization*. Cambridge university press, 2004.

Choromanska, Anna, Henaff, Mikael, Mathieu, Michael, Arous, Gérard Ben, and LeCun, Yann. The loss surfaces of multilayer networks. In *Artificial Intelligence and Statistics*, pp. 192–204, 2015.

Glorot, Xavier and Bengio, Yoshua. Understanding the difficulty of training deep feedforward neural networks. In *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*, pp. 249–256, 2010.

Goodfellow, Ian, Bengio, Yoshua, and Courville, Aaron. *Deep Learning*. MIT Press, 2016. http://www.deeplearningbook.org.

Gradshteyn, Izrail Solomonovich and Ryzhik, Iosif Moiseevich. *Table of integrals, series, and products*. Academic press, 2014.

Ioffe, Sergey and Szegedy, Christian. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *International conference on machine learning*, pp. 448–456, 2015.

Keskar, Nitish Shirish, Mudigere, Dheevatsa, Nocedal, Jorge, Smelyanskiy, Mikhail, and Tang, Ping Tak Peter. On large-batch training for deep learning: Generalization gap and sharp minima. *arXiv preprint arXiv:1609.04836*, 2016.

Krizhevsky, Alex, Sutskever, Ilya, and Hinton, Geoffrey E. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pp. 1097–1105, 2012.

LeCun, Yann, Cortes, Corinna, and Burges, Christopher JC. The MNIST database of handwritten digits, 1998.

Marčenko, Vladimir A and Pastur, Leonid Andreevich. Distribution of eigenvalues for some sets of random matrices. *Mathematics of the USSR-Sbornik*, 1(4):457, 1967.

Poggio, Tomaso, Rifkin, Ryan, Mukherjee, Sayan, and Niyogi, Partha. General conditions for predictivity in learning theory. *Nature*, 428(6981):419, 2004.

Saxe, Andrew M, McClelland, James L, and Ganguli, Surya. Exact solutions to the nonlinear dynamics of learning in deep linear neural networks. *arXiv preprint arXiv:1312.6120*, 2013.

Schmidhuber, Jürgen. Deep learning in neural networks: An overview. *Neural networks*, 61:85–117, 2015.

Silverstein, Jack W and Choi, Sang-Il. Analysis of the limiting spectral distribution of large dimensional random matrices. *Journal of Multivariate Analysis*, 54(2):295–309, 1995.

Srivastava, Nitish, Hinton, Geoffrey, Krizhevsky, Alex, Sutskever, Ilya, and Salakhutdinov, Ruslan. Dropout: A simple way to prevent neural networks from overfitting. *The Journal of Machine Learning Research*, 15(1): 1929–1958, 2014.

Vapnik, Vladimir. *The nature of statistical learning theory*. Springer science & business media, 2013.

Yao, Yuan, Rosasco, Lorenzo, and Caponnetto, Andrea. On early stopping in gradient descent learning. *Constructive Approximation*, 26(2):289–315, 2007.

Zhang, Chiyuan, Bengio, Samy, Hardt, Moritz, Recht, Benjamin, and Vinyals, Oriol. Understanding deep learning requires rethinking generalization. *arXiv preprint arXiv:1611.03530*, 2016.