Laboratoire des Signaux et Systèmes

CentraleSupélec

Université Paris-Saclay

Ph.D. Mid-Term Evaluation

Author:
Zhenyu Liao

Supervisors:
Prof. Romain Couillet
Prof. Yacine Chitour

CentraleSupélec

March 14, 2018

Curriculum Vitae

Research Summary

Contents

1 Description of the topic

The capacity of artificial intelligence (AI) systems to acquire their own knowledge, by extracting patterns from raw data is known as machine learning (ML). The introduction of machine learning enabled computers to "learn" from the real world and make decisions that appear subjective. Simple machine learning algorithms such as logistic regression or naive Bayes have been used in practice to treat as diverse problems as cesarean delivery recommendation or to separate legitimate e-mails from spam ones [15].

The performance of these simple algorithms relies largely on the *representation* of the data they are given. The representation of the data, which may vary from case to case depending on the given problem and contains crucial information to perform the task, is known as the *feature* of the data for the given task. For instance, in the case of distinguishing black cats from white ones rather than black dogs in pictures, color features may play a more important role than, for example, the features that capture the "look" or the "shape" of the animal.

Many AI tasks can be solved by designing the right set of features to be extracted for the task, then providing these features to a simple machine learning algorithm to make decisions. However, that is easier said than done and for many tasks it is difficult to know what features should be used so as to make a wise decision. For example, it is more subtile to say how each pixel in a picture weights so that the picture looks more like a cat that a dog.

Deep neural networks (DNN) try to attack this problem of extracting (possibly) high-level and abstract features from (often high dimensional and numerous) data by introducing representations that are expressed in terms of other simpler representations and impressive achievements have been obtained [23, 35]. Yet for all the success won with these methods, we have managed only a rudimentary understanding of why and in what contexts they work well. Many questions on

the design of the networks, such as how to decide the number of layers and the size of each layer, as well as what kind of activation function shall one choose, still remain unanswered and thus receive an unprecedented research interest.

In particular, the natural "high dimensional" aspect of the problem (also referred to as the "curse of dimensionality", representing the various phenomena that appear *only* in high dimensional space) is one of the main difficulties that prohibit the use of our intuitions in low-dimensional settings (such as the three-dimensional physical space of everyday experience) for the understanding of DNNs.

In this regard, random matrix theory (RMT), which dates back to the original work of Wishart [42], is a powerful tool in dealing with high dimensional random variables. Two salient features of random matrix-based analyses is that, they naturally introduce the ratio between the data dimension and their size that is of considerable practical interest and, more importantly, many existing results are *universal* in the sense that their expressions are *independent* of the distribution of random variables under study and are therefore suitable for real-world problems of various nature.

Nonetheless, the main difficulty that prevents already existing results to be used in understanding these machine learning algorithms is the nonlinear nature of the models. Real-world machine learning problems (to map the images of cats and dogs to associated classes, for example) often call for highly complex and nonlinear features and are thus in need of a proper adaptation of these matrix-based methods. Based on recent advances in RMT [13, 10, 29], with a sufficiently large size of high dimensional data, analyses become traceable for simple neural networks and may shed new light on the mechanism that combines the nonlinear activation and the statistical structures of the data in these methods with a solid theoretical basis, which is the original motivation of the thesis.

The goal of the Ph.D. thesis is to provide theoretical understanding on the performance of neural networks so as to improve the structure design as well as the training procedure of the networks used today. In the remainder of this article, for the first part in Section 2, we focus on the effects of dimensionality as well as nonlinearity in the design of neural networks, which allows for understandings and improvements of a large variety of learning algorithms, from supervised (randomly connected single-hidden-layer network) to non-supervised ones (random feature-based spectral clustering). Then for the second part in Section 3 we present a preliminary study on the temporal evolution of the training and the generalization performance of a linear single-layer network, trained by gradient descent, on a binary classification task. In Section 4 some other related contributions are listed without detailed description due to the space limitation. Section 5 concludes the article by outlining future research directions.

Notations: Boldface lowercase (uppercase) characters stand for vectors (matrices), and non-boldface scalars respectively. $\mathbf{1}_T$ is the column vector of ones of size T, and $\mathbf{I}_T$ the $T \times T$ identity matrix. The notation $(\cdot)^\mathsf{T}$ denotes the transpose operator. The norm $\|\cdot\|$ is the Euclidean norm for vectors and the

operator norm for matrices.

2 Random features and neural networks

This section is based on the following works:

> C. Louart, **Z. Liao**, R. Couillet, "A Random Matrix Approach to Neural Networks", (in press) Annals of Applied Probability, 2017.
>
> **Z. Liao**, R. Couillet, "On the Spectrum of Random Features Maps of High Dimensional Data", (submitted to) The 35th International Conference on Machine Learning (ICML'18), Stockholm, Sweden, 2018.

2.1 Motivation

Finding relevant features is one of the key steps for solving a machine learning problem. To this end, the backpropagation algorithm is probably the best-known method, with which superhuman performances are commonly achieved for specific tasks in applications such as computer vision [23] and many others [35]. But data-driven approaches such as the backpropagation method, in addition to being computationally demanding, fail to cope with limited amounts of available training data.

One successful alternative in this regard is the use of "random features", exploited both in feed-forward neural networks [17, 34], in large-scale kernel estimation [31, 40] and more recently in random sketching schemes [21]. Random feature maps consist in projections randomly exploring the set of nonlinear representations of the data, hopefully extracting features relevant to some given task. The nonlinearities make these representations more mighty but meanwhile theoretically more difficult to analyze and optimize.

Infinitely large random features maps are nonetheless well understood as they result in (asymptotically) equivalent kernels, the most popular example being random Fourier features and their limiting radial basis kernels [31]. Beyond those asymptotic results, recent advances in random matrix theory give rise to unexpected simplification on the understanding of the finite-dimensional version of these kernels, i.e., when the data number and size are large but of similar order as the random feature vector size [13, 10]. Following the same approach, in this work, we perform a spectral analysis on the Gram matrix of the random feature matrices. This matrix is of key relevance in many associated machine learning methods (e.g., randomly connected single-hidden-layer neural network [17] and spectral clustering [27]) and understanding its spectrum casts an indispensable light on their asymptotic performances.

The major objective of this work is to answer the following questions: How to choose the nonlinearity of activation with respect to the data (or task) at hand? What is the effect of the number of data as well as their dimension on the performance of these random feature-based algorithms? To answer the

questions above, we study the eigenspectrum of the random feature Gram matrix that is of crucial importance in understanding the performance of these random feature-based algorithms, as will be illustrated in the following section.

2.2 Main results

2.2.1 General results on random features

For a data matrix $\mathbf{X} = \left[\mathbf{x}_1, \ldots, \mathbf{x}_T\right] \in \mathbb{R}^{p\times T}$, we define the associated *random feature matrix* $\mathbf{\Sigma}$ of $\mathbf{X}$ by premultiplying some random matrix $\mathbf{W} \in \mathbb{R}^{n\times p}$ with i.i.d. standard Gaussian entries and then applied entry-wise some nonlinear activation function $\sigma(\cdot)$ so that $\mathbf{\Sigma} \equiv \sigma(\mathbf{WX}) \in \mathbb{R}^{n\times T}$, whose columns are simply $\sigma(\mathbf{W}\mathbf{x}_i)$ the associated random feature of $\mathbf{x}_i$.

In this section, we focus on the Gram matrix $\mathbf{G} \equiv \frac{1}{n}\mathbf{\Sigma}^\mathsf{T}\mathbf{\Sigma}$ of the random features, the entry (i, j) of which is given by

$$\mathbf{G}_{ij} = \frac{1}{n}\sigma(\mathbf{W}\mathbf{x}_i)^\mathsf{T}\sigma(\mathbf{W}\mathbf{x}_j) = \frac{1}{n}\sum_{k=1}^{n}\sigma(\mathbf{w}_k^\mathsf{T}\mathbf{x}_i)\sigma(\mathbf{w}_k^\mathsf{T}\mathbf{x}_j)$$

with $\mathbf{w}_k^\mathsf{T}$ the k-th row of $\mathbf{W}$. Note that all $\mathbf{w}_k$ follow the same distribution, so that taking expectation over $\mathbf{w} \equiv \mathbf{w}_k$ of the above equation one results in the average kernel matrix $\mathbf{\Phi}$, with

$$\mathbf{\Phi}_{ij} \equiv \mathbf{\Phi}(\mathbf{x}_i, \mathbf{x}_j) = \mathbb{E}_\mathbf{w}[\mathbf{G}_{ij}] = \mathbb{E}_\mathbf{w}[\sigma(\mathbf{w}^\mathsf{T}\mathbf{x}_i)\sigma(\mathbf{w}^\mathsf{T}\mathbf{x}_j)]. \tag{1}$$

Under some regularity condition on the function σ, we show in [25] that the large dimensional (large n, p, T) characterization of $\mathbf{G}$, in particular its eigenspectrum, is fully determined by $\mathbf{\Phi}$ and the ratio n/p, which will be discussed in detail in the following paragraphs.

To properly state our main results, the following assumptions will be needed.

Assumption 1 (Lipschitz σ)**.** *The function σ is Lipschitz continuous with parameter λ_σ.*

Assumption 2 (Growth rate)**.** *As $n \to \infty$,*

1. $0 < \liminf_n \min\{p/n, T/n\} \leq \limsup_n \max\{p/n, T/n\} < \infty$*, while λ_σ is kept constant*
2. $\mathbf{X}$ *has a bounded operator norm:* $\limsup_n \|\mathbf{X}\| < \infty$

As a standard preliminary step in the asymptotic random matrix analysis of the expectation $\mathbb{E}[\mathbf{Q}]$ of the *resolvent* of $\mathbf{G}$ given by

$$\mathbf{Q}(z) \equiv (\mathbf{G} - z\mathbf{I}_T)^{-1}$$

that is of central importance to determine the spectrum of $\mathbf{G}$, a convergence of quadratic forms based on the row vectors of $\mathbf{\Sigma}$ is necessary (see e.g., [26, 36]).

Such results are usually obtained by exploiting the independence (or linear dependence) in the vector entries. This not being the case here, as the entries of the vector $\sigma(\mathbf{X}^\mathsf{T}\mathbf{w})$ are in general not independent, we resort to a concentration of measure approach, as advocated in [19]. The following lemma, stated here in a non-asymptotic random matrix regime (that is, without necessarily resorting to Assumption 2), and thus of independent interest, provides this concentration result.

Lemma 1 (Concentration of quadratic forms)**.** *Let Assumptions 1 hold and* $\mathbf{A} \in \mathbb{R}^{T\times T}$ *such that* $\|\mathbf{A}\| \leq 1$. *For* $\mathbf{X} \in \mathbb{R}^{p\times T}$ *and* $\mathbf{w} \sim \mathcal{N}(\mathbf{0}_p, \mathbf{I}_p)$, *denote the random vector* $\boldsymbol{\sigma} \equiv \sigma(\mathbf{X}^\mathsf{T}\mathbf{w}) \in \mathbb{R}^T$. *Then,*

$$P\left(\left|\frac{1}{T}\boldsymbol{\sigma}^\mathsf{T}\mathbf{A}\boldsymbol{\sigma} - \frac{1}{T}\operatorname{tr}(\boldsymbol{\Phi}\mathbf{A})\right| > t\right) \leq Ce^{-\frac{cT}{\|\mathbf{X}\|^2\lambda_\sigma^2}\min\left(\frac{t^2}{t_0^2}, t\right)}$$

for $t_0 \equiv |\sigma(0)| + \lambda_\sigma\|\mathbf{X}\|\sqrt{\frac{p}{T}}$ *and* $C, c > 0$ *independent of all other parameters. In particular, under the additional Assumption 2,*

$$P\left(\left|\frac{1}{T}\boldsymbol{\sigma}^\mathsf{T}\mathbf{A}\boldsymbol{\sigma} - \frac{1}{T}\operatorname{tr}(\boldsymbol{\Phi}\mathbf{A})\right| > t\right) \leq Ce^{-cn\min(t,t^2)}$$

for some $c, c > 0$.

With the above result in place, the standard resolvent approaches of random matrix theory apply, providing our first theoretical finding as follows.

Theorem 1 (Asymptotic equivalent for $\mathbb{E}[\mathbf{Q}]$)**.** *Let Assumptions 1 and 2 hold and define* $\bar{\mathbf{Q}}(z)$ *as*

$$\bar{\mathbf{Q}}(z) \equiv \left(\frac{\boldsymbol{\Phi}}{1+\delta(z)} - z\mathbf{I}_T\right)^{-1}$$

where $\delta(z)$ *is implicitly defined as the unique solution of* $\delta(z) = \frac{1}{n}\operatorname{tr}(\boldsymbol{\Phi}\bar{\mathbf{Q}}(z))$. *Then, for all* $\epsilon > 0$, *there exists* $c > 0$ *such that*

$$\|\mathbb{E}[\mathbf{Q}] - \bar{\mathbf{Q}}\| \leq cn^{-\frac{1}{2}+\epsilon}.$$

As an immediate consequence of Theorem 1, we are able to perform asymptotic analysis on the training and test mean-square error of single-layer randomly connected neural networks as shown in Figure 1, also referred to as the extreme learning machines [17], which is essentially a ridge-regression on the random feature maps. More concretely, for a training data $\mathbf{X}$ with associated random feature $\boldsymbol{\Sigma}$, the output of the network is given by the inner product $\boldsymbol{\beta}^\mathsf{T}\boldsymbol{\Sigma} \in \mathbb{R}^{d\times T}$ for some $\boldsymbol{\beta} \in \mathbb{R}^{n\times d}$ to be designed. In this work we take the matrix $\boldsymbol{\beta}$ that minimizes the (penalized) mean-square error $\frac{1}{T}\|\boldsymbol{\beta}^\mathsf{T}\boldsymbol{\Sigma} - \mathbf{Y}\|_F^2 + \frac{n}{T}\gamma\|\boldsymbol{\beta}\|_F^2$, where we denote $\mathbf{Y} = \left[\mathbf{y}_1, \ldots, \mathbf{y}_T\right] \in \mathbb{R}^{d\times T}$ the associated target for the training data

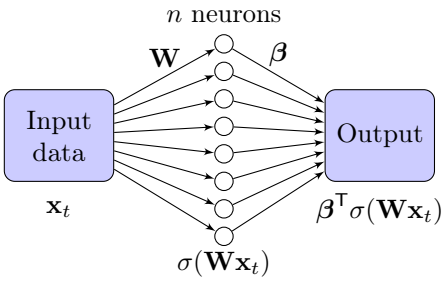

Figure 1: Illustration of a single-layer neural network

$\mathbf{X}$ and γ some regularization factor. Solving for $\boldsymbol{\beta}$, this leads to the explicit *ridge-regressor*

$$\boldsymbol{\beta} = \frac{1}{T}\boldsymbol{\Sigma}\left(\frac{1}{T}\boldsymbol{\Sigma}^\mathsf{T}\boldsymbol{\Sigma} + \frac{n}{T}\gamma\mathbf{I}_T\right)^{-1}\mathbf{Y}^\mathsf{T} = \frac{1}{n}\boldsymbol{\Sigma}\mathbf{Q}\left(-\gamma\right)\mathbf{Y}^\mathsf{T}$$

As can be seen from above, the resolvent $\mathbf{Q}$ naturally appears as a key quantity in the performance analysis of the neural network. Notably, the mean-square error E_{train} on the training dataset $\mathbf{X}$ is given by

$$E_{\text{train}} = \frac{1}{T}\|\mathbf{Y} - \boldsymbol{\beta}^\mathsf{T}\boldsymbol{\Sigma}\|_F^2 = \frac{\gamma^2}{T}\operatorname{tr}\left(\mathbf{Y}\mathbf{Q}^2\left(-\gamma\right)\mathbf{Y}^\mathsf{T}\right). \tag{2}$$

Once trained on $\mathbf{X}$, the network operates, for $\boldsymbol{\beta}$ now fixed, on an unseen dataset $\hat{\mathbf{X}} \in \mathbb{R}^{p\times\hat{T}}$ of size $\hat{T}$ with corresponding unknown target $\hat{\mathbf{Y}} \in \mathbb{R}^{d\times\hat{T}}$. The test performance of the network is thus measured by the test mean-square error

$$E_{\text{test}} = \frac{1}{\hat{T}}\|\hat{\mathbf{Y}} - \boldsymbol{\beta}^\mathsf{T}\hat{\boldsymbol{\Sigma}}\|_F^2 \tag{3}$$

with $\hat{\boldsymbol{\Sigma}} \equiv \sigma(\mathbf{W}\hat{\mathbf{X}})$ and the same $\boldsymbol{\beta}$ as in (2) (and thus only depends on $(\mathbf{X}, \mathbf{Y})$ and γ). By Theorem 1 and some further calculations, we have the following results on the training and generalization performance of the network.

Theorem 2 (Asymptotic training and test mean-square error)**.** *Let Assumptions 1 and 2 hold. For $E_{\text{train}}, E_{\text{test}}$ given by (2) and (3), $\bar{\mathbf{Q}}(z)$ defined as in Theorem 1, denote*

$$\bar{E}_{\text{train}} = \frac{\gamma^2}{T}\operatorname{tr}\mathbf{Y}\bar{\mathbf{Q}}\left[\frac{\frac{1}{n}\operatorname{tr}\left(\bar{\mathbf{Q}}\boldsymbol{\Psi}\bar{\mathbf{Q}}\right)}{1-\frac{1}{n}\operatorname{tr}\left(\boldsymbol{\Psi}\bar{\mathbf{Q}}\boldsymbol{\Psi}\bar{\mathbf{Q}}\right)}\boldsymbol{\Psi} + \mathbf{I}_T\right]\bar{\mathbf{Q}}\mathbf{Y}^\mathsf{T}$$

$$\begin{aligned}\bar{E}_{\text{test}} = {} & \frac{1}{\hat{T}}\|\hat{\mathbf{Y}} - \hat{\mathbf{Y}}\bar{\mathbf{Q}}\boldsymbol{\Psi}_{\mathbf{X}\hat{\mathbf{X}}}\|_F^2 \\ & + \frac{\frac{1}{n}\operatorname{tr}\mathbf{Y}\bar{\mathbf{Q}}\boldsymbol{\Psi}\bar{\mathbf{Q}}\mathbf{Y}^\mathsf{T}}{1-\frac{1}{n}\operatorname{tr}\left(\boldsymbol{\Psi}\bar{\mathbf{Q}}\boldsymbol{\Psi}\bar{\mathbf{Q}}\right)}\left[\frac{1}{\hat{T}}\operatorname{tr}\boldsymbol{\Psi}_{\hat{\mathbf{X}}\hat{\mathbf{X}}} - \frac{1}{\hat{T}}\operatorname{tr}\left(\mathbf{I}_T + \frac{n}{T}\gamma\bar{\mathbf{Q}}\right)\boldsymbol{\Psi}_{\mathbf{X}\hat{\mathbf{X}}}\boldsymbol{\Psi}_{\hat{\mathbf{X}}\mathbf{X}}\bar{\mathbf{Q}}\right].\end{aligned}$$

Then, for all $\epsilon > 0$,

$$n^{\frac{1}{2}-\epsilon}\left(E_{\text{train}} - \bar{E}_{\text{train}}\right) \to 0$$
$$n^{\frac{1}{2}-\epsilon}\left(E_{\text{test}} - \bar{E}_{\text{test}}\right) \to 0$$

almost surely, with $\boldsymbol{\Psi}(z) \equiv \frac{\boldsymbol{\Phi}}{1+\delta(z)}$ *and more generally*

$$\boldsymbol{\Phi}_{\mathbf{AB}} \equiv \mathbb{E}_{\mathbf{w}}\left[\sigma(\mathbf{w}^{\mathsf{T}}\mathbf{A})^{\mathsf{T}}\sigma(\mathbf{w}^{\mathsf{T}}\mathbf{B})\right]$$
$$\boldsymbol{\Psi}_{\mathbf{AB}}(z) \equiv \frac{\boldsymbol{\Phi}_{\mathbf{AB}}}{1+\delta(z)}$$
$$\boldsymbol{\Phi} \equiv \boldsymbol{\Phi}_{\mathbf{XX}}, \quad \boldsymbol{\Psi}(z) \equiv \boldsymbol{\Psi}_{\mathbf{XX}}(z)$$

where we ignore the argument $z = -\gamma$ *for* $\bar{\mathbf{Q}}(z)$ *and* $\boldsymbol{\Psi}(z)$ *for notational simplicity.*

With Theorem 2 at hand, it remains to compute the matrix $\boldsymbol{\Phi}$ for a thorough evaluation of the performance of single-layer randomly connected neural networks. To this end, we first consider the data $\mathbf{X}$ to be deterministic and use the fact that when the entries of $\mathbf{W}$ follow a standard Gaussian distribution, we can compute the generic form $\boldsymbol{\Phi}(\mathbf{a},\mathbf{b}) = \mathbb{E}_{\mathbf{w}}[\sigma(\mathbf{w}^{\mathsf{T}}\mathbf{a})\sigma(\mathbf{w}^{\mathsf{T}}\mathbf{b})]$ by applying the integral trick from [41], for a large set of nonlinear functions $\sigma(\cdot)$ and arbitrary vector $\mathbf{a},\mathbf{b}$ of appropriate dimension. We list the results for commonly used functions in Table 1.

Table 1: $\boldsymbol{\Phi}(\mathbf{a},\mathbf{b})$ for different $\sigma(\cdot)$, $\angle(\mathbf{a},\mathbf{b}) \equiv \frac{\mathbf{a}^{\mathsf{T}}\mathbf{b}}{\|\mathbf{a}\|\|\mathbf{b}\|}$.

$\sigma(t)$	$\boldsymbol{\Phi}(\mathbf{a},\mathbf{b})$
t	$\mathbf{a}^{\mathsf{T}}\mathbf{b}$
$\max(t,0)$	$\frac{1}{2\pi}\|\mathbf{a}\|\|\mathbf{b}\|\left(\angle(\mathbf{a},\mathbf{b})\arccos\left(-\angle(\mathbf{a},\mathbf{b})\right) + \sqrt{1-\angle(\mathbf{a},\mathbf{b})^2}\right)$
$\lvert t\rvert$	$\frac{2}{\pi}\|\mathbf{a}\|\|\mathbf{b}\|\left(\angle(\mathbf{a},\mathbf{b})\arcsin\left(\angle(\mathbf{a},\mathbf{b})\right) + \sqrt{1-\angle(\mathbf{a},\mathbf{b})^2}\right)$
$1_{t>0}$	$\frac{1}{2} - \frac{1}{2\pi}\arccos\left(\angle(\mathbf{a},\mathbf{b})\right)$
$\text{sign}(t)$	$\frac{2}{\pi}\arcsin\left(\angle(\mathbf{a},\mathbf{b})\right)$
$\varsigma_2 t^2 + \varsigma_1 t + \varsigma_0$	$\varsigma_2^2\left(2\left(\mathbf{a}^{\mathsf{T}}\mathbf{b}\right)^2 + \|\mathbf{a}\|^2\|\mathbf{b}\|^2\right) + \varsigma_1^2\mathbf{a}^{\mathsf{T}}\mathbf{b} + \varsigma_2\varsigma_0\left(\|\mathbf{a}\|^2 + \|\mathbf{b}\|^2\right) + \varsigma_0^2$
$\cos(t)$	$\exp\left(-\frac{1}{2}\left(\|\mathbf{a}\|^2 + \|\mathbf{b}\|^2\right)\right)\cosh(\mathbf{a}^{\mathsf{T}}\mathbf{b})$
$\sin(t)$	$\exp\left(-\frac{1}{2}\left(\|\mathbf{a}\|^2 + \|\mathbf{b}\|^2\right)\right)\sinh(\mathbf{a}^{\mathsf{T}}\mathbf{b})$
$\text{erf}(t)$	$\frac{2}{\pi}\arcsin\left(\frac{2\mathbf{a}^{\mathsf{T}}\mathbf{b}}{\sqrt{(1+2\|\mathbf{a}\|^2)(1+2\|\mathbf{b}\|^2)}}\right)$
$\exp(-\frac{t^2}{2})$	$\frac{1}{\sqrt{(1+\|\mathbf{a}\|^2)(1+\|\mathbf{b}\|^2)-(\mathbf{a}^{\mathsf{T}}\mathbf{b})^2}}$

To corroborate the findings in Theorem 2 we consider the task of classifying the popular MNIST image database [24], composed of grayscale handwritten digits of size 28×28, with a neural network composed of $n = 512$ units. We represent here each image as a $p = 784$-size vector; 1 024 images of sevens and

1 024 images of nines were extracted from the database and were evenly split in 512 training and test images, respectively. The database images were jointly centered and scaled so to fall close to the setting of Assumption 2 on $\mathbf{X}$ and $\hat{\mathbf{X}}$ (an admissible preprocessing intervention). The columns of the output values $\mathbf{Y}$ and $\hat{\mathbf{Y}}$ were taken as unidimensional ($d = 1$) with $\mathbf{Y}_j, \hat{\mathbf{Y}}_j \in \{-1, 1\}$ depending on the image class. Figure 2 displays the simulated (averaged over 100 realizations of $\mathbf{W}$) versus theoretical values of E_{train} and E_{test} for three choices of Lipschitz continuous functions $\sigma(\cdot)$, as a function of γ. We observe an almost perfect match between the simulations and our theoretical results from Theorem 2 for not so large n, p, T.

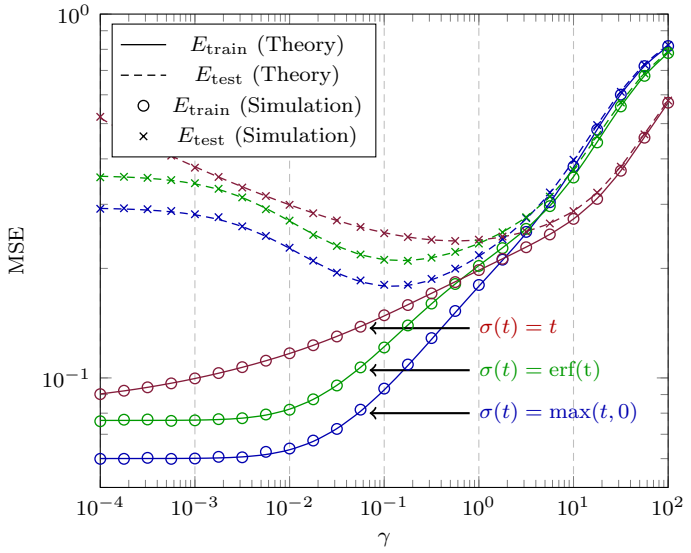

Figure 2: Performance of the network for Lipschitz σ, as a function of γ, for MNIST data (number 7 and 9), $n = 512$, $T = \hat{T} = 1024$, $p = 784$.

2.2.2 Random features for Gaussian mixture models

The results presented above tell us the influence of dimensionality on the performance of neural networks, as well as the role played by the nonlinear activation via Table 1. Nonetheless, they are insufficient to provide a criterion for the choice of the nonlinearity, for a given task. To study the mutual influence between different nonlinearities and the statistical structure of the data, we assume in addition that $\mathbf{x}_1, \ldots, \mathbf{x}_T \in \mathbb{R}^p$ are independent data vectors, each belonging to one of K distribution classes $\mathcal{C}_1, \ldots, \mathcal{C}_K$. Class $\mathcal{C}_a$ has cardinality T_a, for all

$a \in \{1, \dots, K\}$. The data vector $\mathbf{x}_i$ follows a Gaussian mixture model[1], i.e.,

$$\mathbf{x}_i = \frac{1}{\sqrt{p}} \boldsymbol{\mu}_a + \boldsymbol{\omega}_i$$

with $\boldsymbol{\omega}_i \sim \mathcal{N}(\mathbf{0}, \frac{1}{p}\mathbf{C}_a)$ for some mean $\boldsymbol{\mu}_a \in \mathbb{R}^p$ and covariance $\mathbf{C}_a \in \mathbb{R}^{p \times p}$ of associated class $\mathcal{C}_a$ that satisfy the following assumptions.

Assumption 3 (Growth rate of GMM)**.** *As $n \to \infty$,*

1. *for each $a \in \{1, \dots, K\}$, $\frac{T_a}{T} \to c_a \in (0, 1)$*
2. *$\|\boldsymbol{\mu}_a\| = O(1)$*
3. *letting $\mathbf{C}^\circ \equiv \sum_{a=1}^K \frac{T_a}{T} \mathbf{C}_a$ and for each $a \in \{1, \dots, K\}$, $\mathbf{C}_a^\circ \equiv \mathbf{C}_a - \mathbf{C}^\circ$, then $\|\mathbf{C}_a\| = O(1)$ and $\frac{1}{\sqrt{p}} \operatorname{tr}(\mathbf{C}_a^\circ) = O(1)$*
4. *for technical convenience we assume in addition that $\tau \equiv \frac{1}{p} \operatorname{tr}(\mathbf{C}^\circ)$ converges in $(0, \infty)$.*

Assumption 3 ensures that the information about data means or covariances is neither too simple nor impossible to be extracted from the data, as closely investigated in [10].

The crux of interpreting the results in both Theorem 1 and 2 as a function of the activation function $\sigma(\cdot)$ lies on the average kernel $\boldsymbol{\Phi}$ which is the expectation of the Gram matrix as given in (1). Since the Gram matrix $\mathbf{G}$ describes the correlation of data in the *feature space*, it is natural to recenter $\mathbf{G}$, and thus $\boldsymbol{\Phi}$ by pre- and post-multiplying a projection matrix $\mathbf{P} \equiv \mathbf{I}_T - \frac{1}{T}\mathbf{1}_T\mathbf{1}_T^\mathsf{T}$. In the case of $\boldsymbol{\Phi}$, we get

$$\boldsymbol{\Phi}_c \equiv \mathbf{P}\boldsymbol{\Phi}\mathbf{P}.$$

Let us now introduce the key steps of our present analysis. Under Assumption 3, observe that for $\mathbf{x}_i \in \mathcal{C}_a$ and $\mathbf{x}_j \in \mathcal{C}_b$, $i \neq j$,

$$\mathbf{x}_i^\mathsf{T}\mathbf{x}_j = \underbrace{\boldsymbol{\omega}_i^\mathsf{T}\boldsymbol{\omega}_j}_{O(p^{-1/2})} + \underbrace{\boldsymbol{\mu}_a^\mathsf{T}\boldsymbol{\mu}_b/p + \boldsymbol{\mu}_a^\mathsf{T}\boldsymbol{\omega}_j/\sqrt{p} + \boldsymbol{\mu}_b^\mathsf{T}\boldsymbol{\omega}_i/\sqrt{p}}_{O(p^{-1})}$$

which allows one to perform a Taylor expansion around 0 as $p, T \to \infty$, to give a reasonable approximation of nonlinear functions of $\mathbf{x}_i^\mathsf{T}\mathbf{x}_j$, such as those appearing in $\boldsymbol{\Phi}_{ij}$ (see again Table 1). For $i = j$, one has instead

$$\|\mathbf{x}_i\|^2 = \underbrace{\|\boldsymbol{\omega}_i\|^2}_{O(1)} + \underbrace{\|\boldsymbol{\mu}_a\|^2/p + 2\boldsymbol{\mu}_a^\mathsf{T}\boldsymbol{\omega}_i/\sqrt{p}}_{O(n^{-1})}.$$

From $\mathbb{E}_{\boldsymbol{\omega}_i}[\|\boldsymbol{\omega}_i\|^2] = \operatorname{tr}(\mathbf{C}_a)/p$ it is convenient to further write $\|\boldsymbol{\omega}_i\|^2 = \operatorname{tr}(\mathbf{C}_a)/p + \big(\|\omega_i\|^2 - \operatorname{tr}(\mathbf{C}_a)/p\big)$, where $\operatorname{tr}(\mathbf{C}_a)/p = O(1)$ and $\|\omega_i\|^2 - \operatorname{tr}(\mathbf{C}_a)/p =$

[1] We normalize the data by $\frac{1}{\sqrt{p}}$, together with Assumption 3 to ensure $\|\mathbf{x}_i\| = O(1)$ with high probability, as in consistent with Assumption 2.

$O(n^{-1/2})$. By definition $\tau \equiv \operatorname{tr}(\mathbf{C}^\circ)/p = O(1)$ and exploiting again Assumption 2 one results in,

$$\|\mathbf{x}_i\|^2 = \underbrace{\tau}_{O(1)} + \underbrace{\operatorname{tr}(\mathbf{C}_a^\circ)/p + \|\boldsymbol{\omega}_i\|^2 - \operatorname{tr}(\mathbf{C}_a)/p}_{O(n^{-1/2})} + \underbrace{\|\boldsymbol{\mu}_a\|^2/p + 2\boldsymbol{\mu}_a^\mathsf{T}\boldsymbol{\omega}_i/\sqrt{p}}_{O(n^{-1})}$$

which allows for a Taylor expansion of nonlinear functions of $\|\mathbf{x}_i\|^2$ around τ, as has been done for $\mathbf{x}_i^\mathsf{T}\mathbf{x}_j$.

From Table 1, it appears that, for every listed $\sigma(\cdot)$, $\boldsymbol{\Phi}(\mathbf{x}_i, \mathbf{x}_j)$ is a smooth function of $\mathbf{x}_i^\mathsf{T}\mathbf{x}_j$ and $\|\mathbf{x}_i\|$, $\|\mathbf{x}_j\|$, despite their possible discontinuities (for example, the ReLU function and $\sigma(t) = |t|$). The above results therefore allow for an entry-wise Taylor expansion of the matrix $\boldsymbol{\Phi}$ in the large p, T limit.

A critical aspect of the analysis where random matrix theory comes into play now consists in developing $\boldsymbol{\Phi}$ as a sum of matrices arising from the Taylor expansion and ignoring terms that give rise to a vanishing operator norm, so as to find an asymptotic equivalent matrix $\tilde{\boldsymbol{\Phi}}$ such that $\|\boldsymbol{\Phi} - \tilde{\boldsymbol{\Phi}}\| \to 0$ as $p, T \to \infty$, as described in detail in the following section. This analysis provides a simplified asymptotically equivalent expression for $\boldsymbol{\Phi}$ with all nonlinearities removed, which is the crux of the present study.

In the remainder of this section, we shall use the following notations for random elements,

$$\boldsymbol{\Omega} \equiv \left[\boldsymbol{\omega}_1, \ldots, \boldsymbol{\omega}_T\right] \in \mathbb{R}^{p \times T}$$
$$\boldsymbol{\phi} \equiv \left\{\|\boldsymbol{\omega}_i\|^2 - \mathbb{E}\left[\|\boldsymbol{\omega}_i\|^2\right]\right\}_{i=1}^T \in \mathbb{R}^T$$

as well as for deterministic elements[2],

$$\mathbf{M} \equiv \left[\boldsymbol{\mu}_1, \ldots, \boldsymbol{\mu}_K\right] \in \mathbb{R}^{p \times K}$$
$$\mathbf{t} \equiv \left\{\frac{1}{\sqrt{p}} \operatorname{tr} \mathbf{C}_a^\circ\right\}_{a=1}^K \in \mathbb{R}^K$$

$$\mathbf{J} \equiv \left[\mathbf{j}_1, \ldots, \mathbf{j}_K\right] \in \mathbb{R}^{T \times K}$$
$$\mathbf{S} \equiv \left\{\frac{1}{p} \operatorname{tr}(\mathbf{C}_a \mathbf{C}_b)\right\}_{a,b=1}^K \in \mathbb{R}^{K \times K}$$

where $\mathbf{j}_a \in \mathbb{R}^T$ denotes the canonical vector of class $\mathcal{C}_a$ such that $(\mathbf{j}_a)_i = \delta_{\mathbf{x}_i \in \mathcal{C}_a}$.

Theorem 3 (Asymptotic Equivalent of Φ_c)**.** *Let Assumption 2 and 3 hold and $\boldsymbol{\Phi}_c$ be defined as $\boldsymbol{\Phi}_c \equiv \mathbf{P}\boldsymbol{\Phi}\mathbf{P}$, with $\boldsymbol{\Phi}$ given in (1). Then, as $n \to \infty$, for all $\sigma(\cdot)$ given in Table 1,*

$$\|\boldsymbol{\Phi}_c - \tilde{\boldsymbol{\Phi}}_c\| \to 0$$

[2] As a reminder here, $\mathbf{M}$ stands for *means*, $\mathbf{t}$ accounts for (difference in) *traces* while $\mathbf{S}$ for the "*shapes*" of covariances.

almost surely, with $\tilde{\mathbf{\Phi}}_c = \mathbf{P}\tilde{\mathbf{\Phi}}\mathbf{P}$ *and*

$$\tilde{\mathbf{\Phi}} \equiv d_1 \left(\mathbf{\Omega} + \mathbf{M}\frac{\mathbf{J}^\mathsf{T}}{\sqrt{p}} \right)^\mathsf{T} \left(\mathbf{\Omega} + \mathbf{M}\frac{\mathbf{J}^\mathsf{T}}{\sqrt{p}} \right) + d_2 \mathbf{U}\mathbf{B}\mathbf{U}^\mathsf{T} + d_0 \mathbf{I}_T$$

where we recall that $\mathbf{P} \equiv \mathbf{I}_T - \frac{1}{T}\mathbf{1}_T\mathbf{1}_T^\mathsf{T}$ *and*

$$\mathbf{U} \equiv \left[\tfrac{\mathbf{J}}{\sqrt{p}}, \boldsymbol{\phi} \right]$$
$$\mathbf{B} \equiv \begin{bmatrix} \mathbf{t}\mathbf{t}^\mathsf{T} + 2\mathbf{S} & \mathbf{t} \\ \mathbf{t}^\mathsf{T} & 1 \end{bmatrix}$$

with the coefficients d_0, d_1, d_2 *given in Table 2.*

Table 2: Coefficients d_i in $\tilde{\mathbf{\Phi}}_c$ for different $\sigma(\cdot)$.

$\sigma(t)$	d_0	d_1	d_2
t	0	1	0
$\max(t,0) \equiv \mathrm{ReLU}(t)$	$\left(\frac{1}{4} - \frac{1}{2\pi}\right)\tau$	$\frac{1}{4}$	$\frac{1}{8\pi\tau}$
$\lvert t \rvert$	$\left(1 - \frac{2}{\pi}\right)\tau$	0	$\frac{1}{2\pi\tau}$
$1_{t>0}$	$\frac{1}{4} - \frac{1}{2\pi}$	$\frac{1}{2\pi\tau}$	0
$\mathrm{sign}(t)$	$1 - \frac{2}{\pi}$	$\frac{2}{\pi\tau}$	0
$\varsigma_2 t^2 + \varsigma_1 t + \varsigma_0$	$2\tau^2\varsigma_2^2$	ς_1^2	ς_2^2
$\cos(t)$	$\frac{1}{2} + \frac{e^{-2\tau}}{2} - e^{-\tau}$	0	$\frac{e^{-\tau}}{4}$
$\sin(t)$	$\frac{1}{2} - \frac{e^{-2\tau}}{2} - \tau e^{-\tau}$	$e^{-\tau}$	0
$\mathrm{erf}(t)$	$\frac{2}{\pi}\left(\arccos\left(\frac{2\tau}{2\tau+1}\right) - \frac{2\tau}{2\tau+1}\right)$	$\frac{4}{\pi}\frac{1}{2\tau+1}$	0
$\exp(-\frac{t^2}{2})$	$\frac{1}{\sqrt{2\tau+1}} - \frac{1}{\tau+1}$	0	$\frac{1}{4(\tau+1)^3}$

Theorem 3 tells us as a corollary (from Corollary 4.3.15 in [16], for example) that the maximal difference between the eigenvalues of $\mathbf{\Phi}_c$ and $\tilde{\mathbf{\Phi}}_c$ vanishes asymptotically as $p, T \to \infty$. Similarly the distance between the "isolated eigenvectors[3]" also vanishes. This is of tremendous importance as the determination of the leading eigenvalues and eigenvectors of $\mathbf{\Phi}_c$ (that contain crucial information for clustering, for example) can be studied from the equivalent problem performed on $\tilde{\mathbf{\Phi}}_c$ and becomes mathematically more tractable.

On closer inspection of Theorem 3, the matrix $\tilde{\mathbf{\Phi}}$ is expressed as the sum of three terms, weighted respectively by the three coefficients d_0, d_1 and d_2, that depend on the nonlinear function $\sigma(\cdot)$ via Table 2. Note that the statistical structure of the data $\{\mathbf{x}_i\}_{i=1}^T$ (namely the means in $\mathbf{M}$ and the covariances in $\mathbf{t}$ and $\mathbf{S}$) is perturbed by random fluctuations ($\mathbf{\Omega}$ and $\boldsymbol{\phi}$) and it is thus impossible to get rid of these noisy terms by wisely choosing the function $\sigma(\cdot)$.

[3] Eigenvectors that correspond to the eigenvalues found at a non-vanishing distance from the other eigenvalues.

However, there does exist a balance between the means and covariances, that provides some instructions in the appropriate choice of the nonlinearity. From Table 2, the functions $\sigma(\cdot)$ can be divided into the following three groups:

- *mean-oriented*, where $d_1 \neq 0$ while $d_2 = 0$: this is the case of the functions t, $1_{t>0}$, $\text{sign}(t)$, $\sin(t)$ and $\text{erf}(t)$, which asymptotically track only the difference in means (i.e., $\mathbf{t}$ and $\mathbf{S}$ disappear from the expression of $\tilde{\mathbf{\Phi}}_c$);

- *covariance-oriented*, where $d_1 = 0$ while $d_2 \neq 0$: this concerns the functions $|t|$, $\cos(t)$ and $\exp(-t^2/2)$, which asymptotically track only the difference in covariances;

- *balanced*, where both $d_1, d_2 \neq 0$: here for the ReLU function $\max(t, 0)$ and the quadratic function $\varsigma_2 t^2 + \varsigma_1 t + \varsigma_0$.

We complete this section by showing that our theoretical results in Theorem 3, derived from Gaussian mixture models, show an unexpected close match in practice when applied to some real-world datasets. We consider two different types of classification tasks: one on handwritten digits of the popular MNIST [24] database (number 6 and 8), and the other on epileptic EEG time series data [2] (set B and E). These two datasets are typical examples of means-dominant (handwritten digits recognition) and covariances-dominant (EEG times series classification) tasks. This is numerically confirmed in Table 3.

Table 3: Empirical estimation of (normalized) differences in means and covariances of the MNIST (Figure 3) and epileptic EEG (Figure 4) datasets.

	$\|\mathbf{M}^\mathsf{T}\mathbf{M}\|$	$\|\mathbf{t}\mathbf{t}^\mathsf{T} + 2\mathbf{S}\|$
MNIST data	172.4	86.0
EEG data	1.2	182.7

Handwritten digits recognition We perform random feature-based spectral clustering on data matrices that consist of $T = 32, 64$ and 128 randomly selected vectorized images of size $p = 784$ from the MNIST dataset. Means and covariances are empirically obtained from the full set of 11 769 MNIST images (5 918 images of number 6 and 5 851 of number 8). Comparing the matrix $\mathbf{\Phi}_c$ built from the data and the theoretically equivalent $\tilde{\mathbf{\Phi}}_c$ obtained as if the data were Gaussian with the (empirically) computed means and covariances, we observe an extremely close fit in the behavior of the eigenvalues and the leading eigenvector in Figure 3. The k-means method is then applied to the leading two eigenvectors of the matrix $\mathbf{G}_c$ that consists of $n = 32$ random features to perform unsupervised classification, with resulting accuracies (averaged over 50 runs) reported in Table 4. As remarked from Table 3, the mean-oriented $\sigma(t)$ functions are expected to outperform the covariance-oriented ones in this task, which is consistent with the results in Table 4.

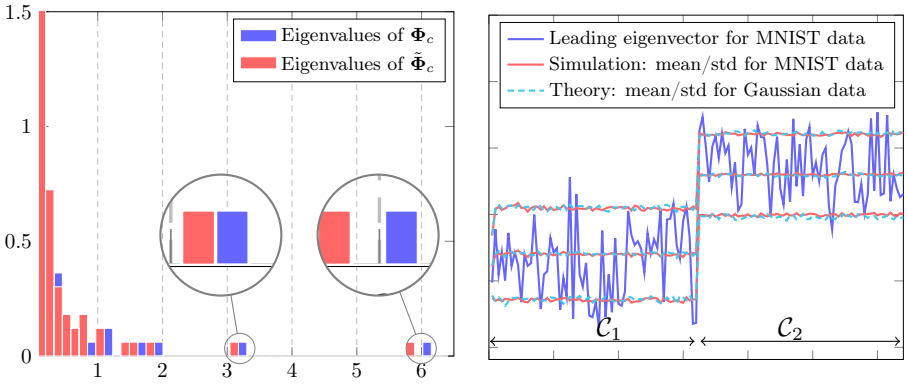

Figure 3: Eigenvalue distribution of $\mathbf{\Phi}_c$ and $\tilde{\mathbf{\Phi}}_c$ for the MNIST data (left) and leading eigenvector of $\mathbf{\Phi}_c$ for the MNIST and Gaussian mixture data (right) with a width of ± 1 standard deviations (generated from 500 trials). With the ReLU function, $p = 784, T = 128$ and $c_1 = c_2 = \frac{1}{2}, \mathbf{j}_1 = [\mathbf{1}_{T_1}; \mathbf{0}_{T_2}]$ and $\mathbf{j}_2 = [\mathbf{0}_{T_1}; \mathbf{1}_{T_2}]$.

EEG time series classification The epileptic EEG dataset[4], developed by the University of Bonn, Germany, is described in [2]. The dataset consists of five subsets (denoted A-E), each containing 100 single-channel EEG segments of 23.6-sec duration. Sets A and B were collected from surface EEG recordings of five healthy volunteers, while sets C, D and E were collected from the EEG records of the pre-surgical diagnosis of five epileptic patients. Here we perform random feature-based spectral clustering on $T = 32, 64$ and 128 randomly picked EEG segments of length $p = 100$ from the dataset. Means and covariances are empirically estimated from the full set (4 097 segments of set B and 4 097 segments of set E). Similar behavior of eigenpairs as for Gaussian mixture models

[4] http://www.meb.unibonn.de/epileptologie/science/physik/eegdata.html.

Table 4: Classification accuracies for random feature-based spectral clustering with different $\sigma(t)$ on the MNIST dataset.

	$\sigma(t)$	$T = 32$	$T = 64$	$T = 128$
mean-oriented	t	85.31%	**88.94%**	87.30%
	$1_{t>0}$	86.00%	82.94%	85.56%
	$\text{sign}(t)$	81.94%	83.34%	85.22%
	$\sin(t)$	85.31%	87.81%	**87.50%**
	$\text{erf}(t)$	**86.50%**	87.28%	86.59%
cov-oriented	$\lvert t \rvert$	62.81%	60.41%	57.81%
	$\cos(t)$	62.50%	59.56%	57.72%
	$\exp(-\frac{t^2}{2})$	64.00%	60.44%	58.67%
balanced	$\text{ReLU}(t)$	82.87%	85.72%	82.27%

is once more observed in Figure 4. After k-means classification on the leading two eigenvectors of the (centered) Gram matrix composed of $n = 32$ random features, the accuracies (averaged over 50 runs) are reported in Table 5.

As opposed to the MNIST image recognition task, from Table 5 it is easy to check that the covariance-oriented functions (i.e., $\sigma(t) = |t|$, $\cos(t)$ and $\exp(-t^2/2)$) far outperform any other with almost perfect classification accuracies. It is particularly interesting to note that the popular ReLU function is suboptimal in both tasks, but never performs very badly, thereby offering a good risk-performance tradeoff.

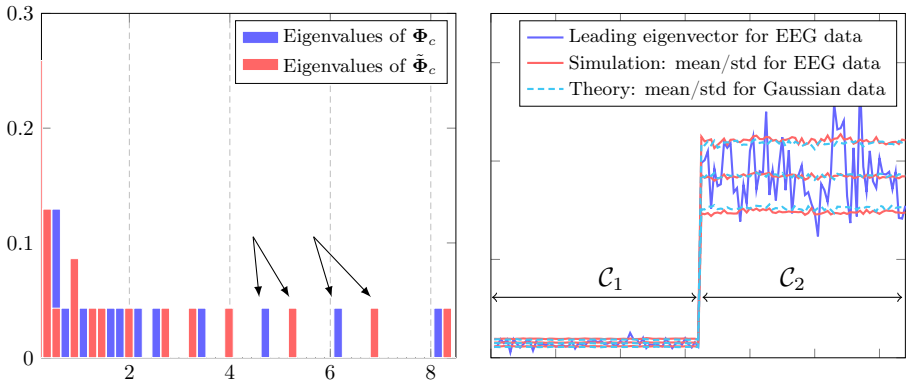

Figure 4: Eigenvalue distribution of $\mathbf{\Phi}_c$ and $\tilde{\mathbf{\Phi}}_c$ for the epileptic EEG data (left) and leading eigenvector of $\mathbf{\Phi}_c$ for the EEG and Gaussian mixture data (right) with a width of ± 1 standard deviations (generated from 500 trials). With the ReLU function, $p = 100, T = 128$ and $c_1 = c_2 = \frac{1}{2}, \mathbf{j}_1 = [\mathbf{1}_{T_1}; \mathbf{0}_{T_2}]$ and $\mathbf{j}_2 = [\mathbf{0}_{T_1}; \mathbf{1}_{T_2}]$.

Table 5: Classification accuracies for random feature-based spectral clustering with different $\sigma(t)$ on the epileptic EEG dataset.

	$\sigma(t)$	$T = 32$	$T = 64$	$T = 128$
mean-oriented	t	71.81%	70.31%	69.58%
	$1_{t>0}$	65.19%	65.87%	63.47%
	$\text{sign}(t)$	67.13%	64.63%	63.03%
	$\sin(t)$	71.94%	70.34%	68.22%
	$\text{erf}(t)$	69.44%	70.59%	67.70%
cov-oriented	$\lvert t\rvert$	99.69%	99.69%	99.50%
	$\cos(t)$	99.00%	99.38%	99.36%
	$\exp(-\frac{t^2}{2})$	**99.81**%	**99.81**%	**99.77**%
balanced	$\text{ReLU}(t)$	84.50%	87.91%	90.97%

3 The learning dynamics of neural networks

This section is based on the following work:

> **Z. Liao**, R. Couillet, "The Dynamics of Learning: A Random Matrix Approach", (submitted to) The 35th International Conference on Machine Learning (ICML'18), Stockholm, Sweden, 2018.

3.1 Motivation

In the work presented above, we investigate the effect of dimensionality and nonlinearity on the representation of the data and, as a consequence, on the performance of neural networks. But how to find more powerful and more informative representations to facilitate the decision-making process of the computer? To this end, we are in demand of a deeper understanding on the training process of neural networks. Nowadays deep neural networks are *almost all* trained by gradient descent. Meanwhile, some of the models exhibit remarkably good generalization performance when applied to unseen data of similar nature, despite the fact that they have far more model parameters than the number of training samples that they are trained on, 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 [39, 5, 30]. 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 [38] or the L_2-penalization (weight decay) as reported in [23]; or implicit, as in the case of the early stopping strategy [43] or the stochastic gradient descent algorithm itself [44]. But none of them is capable of providing a compelling explanation on the impressive generalization performance of today's deep networks.

Inspired by the recent line of works [33, 1], in this work 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 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 [14, 23], may lead to a degradation of the network performance.

3.2 Main results

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 $\mathbf{X} = \big[\mathbf{x}_1, \ldots, \mathbf{x}_n\big] \in \mathbb{R}^{p \times n}$ 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 [7] 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 [33, 1], 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

$$\mathbf{w}(t) = e^{-\frac{\alpha t}{n}\mathbf{X}\mathbf{X}^\mathsf{T}}\mathbf{w}_0 + \left(\mathbf{I}_p - e^{-\frac{\alpha t}{n}\mathbf{X}\mathbf{X}^\mathsf{T}}\right)(\mathbf{X}\mathbf{X}^\mathsf{T})^{-1}\mathbf{X}\mathbf{y} \tag{4}$$

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 $\frac{1}{n}\mathbf{X}\mathbf{X}^\mathsf{T} = \mathbf{V}\mathbf{\Lambda}\mathbf{V}^\mathsf{T}$ and $e^{\mathbf{\Lambda}}$ is a diagonal matrix with elements equal to the exponential of the elements of $\mathbf{\Lambda}$. 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 [1] 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 (4) 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 [3].

With the explicit expression of $\mathbf{w}(t)$ in (4), 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, in addition to Assumption 2 and 3 (in particular, here we take $K = 2$ and $\mathbf{C}_a = \mathbf{I}_p$), we shall initialize the gradient descent algorithm with random weight as follows.

Assumption 4 (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 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 $\mathcal{C}_2$. 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.

Theorem 4 (Generalization performance)**.** *Let Assumptions 2-4 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 $m(z)$ determined by the popular Marčenko–Pastur equation [26]

$$m(z) = \frac{1-c-z}{2cz} + \frac{\sqrt{(1-c-z)^2-4cz}}{2cz} \tag{5}$$

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

More interestingly, (E, V) can be rewritten in a more readable way. First, note from Figure 5 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

$$\nu(dx) = \frac{\sqrt{(x-\lambda_-)^+(\lambda_+-x)^+}}{2\pi c x}dx + \left(1-\frac{1}{c}\right)^+\delta(x) \tag{6}$$

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 [37] and follow the idea in [4] 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) as follows

$$E = \int \frac{1 - f_t(x)}{x} \mu(dx) \tag{7}$$

$$V = \frac{\|\boldsymbol{\mu}\|^2 + c}{\|\boldsymbol{\mu}\|^2} \int \frac{(1 - f_t(x))^2}{x^2} \mu(dx) + \sigma^2 \int f_t^2(x) \nu(dx) \tag{8}$$

where we recall $f_t(x) = \exp(-\alpha t x)$, $\nu(x)$ given by (6) and denote the measure

$$\mu(dx) \equiv \frac{\sqrt{(x - \lambda_-)^+ (\lambda_+ - x)^+}}{2\pi (\lambda_s - x)^+} dx + \frac{(\|\boldsymbol{\mu}\|^4 - c)^+}{\|\boldsymbol{\mu}\|^2} \delta_{\lambda_s}(x) \tag{9}$$

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 $\|\boldsymbol{\mu}\|^2 = \sqrt{c}$.

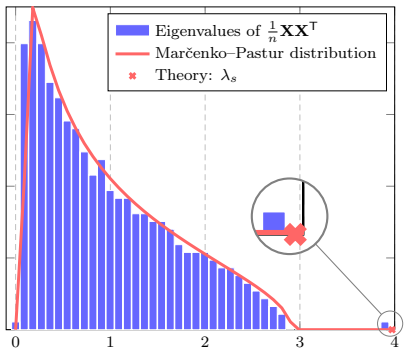

Figure 5: Eigenvalue distribution of $\frac{1}{n}\mathbf{X}\mathbf{X}^\mathsf{T}$ for $\boldsymbol{\mu} = [1.5; \mathbf{0}_{p-1}]$, $p = 512$, $n = 1\,024$ and $c_1 = c_2 = 1/2$.

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 (5), may be determined through more elaborate (often implicit) formulations. Our analysis scheme can be easily extended to handle these more intricate cases.

On a closer inspection of the expressions in (7) and (8), we have a first and important remark as follows.

Remark 1 (Optimal generalization performance)**.** *Note from* (9) *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 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 [15] and at the same time to mitigate the drop of performance due to large σ^2.

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

$$\frac{\boldsymbol{\mu}^\mathsf{T}\mathbf{w}_{LS}}{\|\mathbf{w}_{LS}\|} = \frac{\|\boldsymbol{\mu}\|^2}{\sqrt{\|\boldsymbol{\mu}\|^2+c}} \sqrt{1 - \min\left(c, \frac{1}{c}\right)}. \tag{10}$$

Comparing (10) 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 [1]. However, the point $c = 1$ is a singularity for (10), but not for $\frac{E}{\sqrt{V}}$ given by (7) and (8). 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 this section, 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 [6] 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 [22], that can be naturally linked to the ratio n_{batch}/p in the random matrix analysis.

4 Other related contributions

Besides the work presented above in Section 2 and 3 but still in the general context of RMT, the Ph.D. student also works on the random matrix-based analysis of kernel methods, where some interesting results have been reported in the following works

> **Z. Liao**, R. Couillet, "Random Matrices Meet Machine Learning: A Large Dimensional Analysis of LS-SVM", IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP'17), New Orleans, USA, 2017.
>
> **Z. Liao**, R. Couillet, "A Large Dimensional Analysis of Least Squares Support Vector Machines", (submitted to) Journal of Machine Learning Research, 2017

In parallel, the Ph.D. student works with the co-supervisor Prof. Yacine CHITOUR on the learning dynamics and the geometry of loss surface of deep neural networks, from a viewpoint of "geometry and control" rather than probabilistic, with an article in preparation in this regard. Nonetheless, neither of those research activities will be presented in the report due to the space limitation.

5 Future work

In the same vein as the work performed in the first part for the analysis of neural networks, the second part of the thesis will concentrate on the following three directions.

Inner product kernels and activations in neural networks. Despite the results established in Section 2.2.2, we are still looking for a more simple and more intrinsic connection between the spectrum of the Gram matrix and the activation function $\sigma(\cdot)$ to replace the burdensome computation of the coefficients d_1, d_2 for each $\sigma(\cdot)$. In this respect, inspired by the work of [8] on the spectrum of inner product kernel matrix, we leverage the powerful tool of orthogonal polynomials [12] that are ideal to handle entry-wise (nonlinear) function σ of random variable x by expanding $\sigma(x)$ as the sum of orthogonal polynomials of x. We believe more fundamental understanding on the nonlinearity can be provided without the (unnecessary) assumption on the Lipschitz continuity of $\sigma(\cdot)$.

The loss landscape of deep networks. The optimization of deep networks is considered difficult mainly due to the highly non-convex nature of the loss function. In particular, it has been believed to be problematic to have numerous saddle points as dominant critical points that may plague optimization, as well as the existence of many local minimals that may lead to poor generalization performance. There is no shortage of prior work in this direction, see e.g., [11, 9, 20, 28], where random matrix theory naturally appears as a powerful

tool for evaluating the eigenvalue distribution of the Hessian matrix so as to understand the nature of critical points (to distinguish saddle points from local minimals, for example). Nonetheless, the existing results today are limited to rather simple or unrealistic network structure and thus leave open the possibility for more general and elaborate models that are of greater practical significance.

Convolution neural networks. Convolutional neural networks (CNNs) are a specialized kind of neural networks that use *convolution* in place of general matrix multiplication for processing data that has a known grid-like topology. CNNs are proposed because training with backpropagation on fully connected layers becomes computational expensive for high dimensional data. Moreover, it is believed that the convolution operation leverages several important ideas that helps improve the training of neural networks, such as sparse connections, parameter sharing and equivalent representations [15] and is thus indispensable to ensure satisfactory performance of the network. In particular, it has been observed in practice that random filters performed only slightly worse than pretrained ones [18, 32]. Such study may shed new light on the design of more vision-oriented neural networks.

References

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

[2] Ralph G Andrzejak, Klaus Lehnertz, Florian Mormann, Christoph Rieke, Peter David, and Christian E Elger. Indications of nonlinear deterministic and finite-dimensional structures in time series of brain electrical activity: Dependence on recording region and brain state. *Physical Review E*, 64(6):061907, 2001.

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

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

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

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

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

[8] Xiuyuan Cheng and Amit Singer. The spectrum of random inner-product kernel matrices. *Random Matrices: Theory and Applications*, 2(04):1350010, 2013.

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

[10] Romain Couillet, Florent Benaych-Georges, et al. Kernel spectral clustering of large dimensional data. *Electronic Journal of Statistics*, 10(1):1393–1454, 2016.

[11] Yann N Dauphin, Razvan Pascanu, Caglar Gulcehre, Kyunghyun Cho, Surya Ganguli, and Yoshua Bengio. Identifying and attacking the saddle point problem in high-dimensional non-convex optimization. In *Advances in neural information processing systems*, pages 2933–2941, 2014.

[12] Percy Deift. *Orthogonal polynomials and random matrices: a Riemann-Hilbert approach*, volume 3. American Mathematical Soc., 1999.

[13] Noureddine El Karoui et al. The spectrum of kernel random matrices. *The Annals of Statistics*, 38(1):1–50, 2010.

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

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

[16] Roger A Horn and Charles R Johnson. *Matrix analysis*. Cambridge university press, 2012.

[17] Guang-Bin Huang, Hongming Zhou, Xiaojian Ding, and Rui Zhang. Extreme learning machine for regression and multiclass classification. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, 42(2):513–529, 2012.

[18] Kevin Jarrett, Koray Kavukcuoglu, Yann LeCun, et al. What is the best multi-stage architecture for object recognition? In *Computer Vision, 2009 IEEE 12th International Conference on*, pages 2146–2153. IEEE, 2009.

[19] Noureddine El Karoui. Concentration of measure and spectra of random matrices: Applications to correlation matrices, elliptical distributions and beyond. *The Annals of Applied Probability*, pages 2362–2405, 2009.

[20] Kenji Kawaguchi. Deep learning without poor local minima. In *Advances In Neural Information Processing Systems*, pages 586–594, 2016.

[21] Nicolas Keriven, Anthony Bourrier, Rémi Gribonval, and Patrick Pérez. Sketching for large-scale learning of mixture models. In *Acoustics, Speech and Signal Processing (ICASSP), 2016 IEEE International Conference on*, pages 6190–6194. IEEE, 2016.

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

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

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

[25] Cosme Louart, Zhenyu Liao, and Romain Couillet. A random matrix approach to neural networks. *arXiv preprint arXiv:1702.05419*, 2017.

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

[27] Andrew Y Ng, Michael I Jordan, and Yair Weiss. On spectral clustering: Analysis and an algorithm. In *Advances in neural information processing systems*, pages 849–856, 2002.

[28] Jeffrey Pennington and Yasaman Bahri. Geometry of neural network loss surfaces via random matrix theory. In *International Conference on Machine Learning*, pages 2798–2806, 2017.

[29] Jeffrey Pennington and Pratik Worah. Nonlinear random matrix theory for deep learning. In *Advances in Neural Information Processing Systems*, pages 2634–2643, 2017.

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

[31] Ali Rahimi and Benjamin Recht. Random features for large-scale kernel machines. In *Advances in neural information processing systems*, pages 1177–1184, 2008.

[32] Andrew M Saxe, Pang Wei Koh, Zhenghao Chen, Maneesh Bhand, Bipin Suresh, and Andrew Y Ng. On random weights and unsupervised feature learning. In *ICML*, pages 1089–1096, 2011.

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

[34] Simone Scardapane and Dianhui Wang. Randomness in neural networks: an overview. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, 7(2), 2017.

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

[36] Jack W Silverstein and ZD Bai. On the empirical distribution of eigenvalues of a class of large dimensional random matrices. *Journal of Multivariate analysis*, 54(2):175–192, 1995.

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

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

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

[40] Andrea Vedaldi and Andrew Zisserman. Efficient additive kernels via explicit feature maps. *IEEE transactions on pattern analysis and machine intelligence*, 34(3):480–492, 2012.

[41] Christopher KI Williams. Computing with infinite networks. *Advances in neural information processing systems*, pages 295–301, 1997.

[42] John Wishart. The generalised product moment distribution in samples from a normal multivariate population. *Biometrika*, pages 32–52, 1928.

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

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