CIMAT Centro de Investigación en Matemáticas, A.C.

ANALYSIS OF SOME STATISTICAL PROPERTIES OF NEURAL NETWORKS

TESIS

Que para obtener el grado de

Maestro en Ciencias

con Orientación en

Probabilidad y Estadística

Presenta

Juan Francisco Mandujano Reyes

Director de Tesis:

Dr. Víctor Manuel Pérez Abreu Carrión

Autorización de la versión final

Guanajuato, Gto., 13 de septiembre de 2019

CIMAT
CENTRO DE INVESTIGACION EN MATEMATICAS A. C.

Centro de Investigación en Matemáticas, A.C.

Acta de Examen de Grado

Acta No.:	154
Libro No.:	002
Foja No.:	154

En la Ciudad de Guanajuato, Gto., siendo las 12:00 horas del día 13 de septiembre del año 2019, se reunieron los miembros del jurado integrado por los señores:

DR. ROGELIO RAMOS QUIROGA (CIMAT)
DR. OSCAR SUSANO DALMAU CEDEÑO (CIMAT)
DR. MARIO ALBERTO DÍAZ TORRES (CIMAT)

bajo la presidencia del primero y con carácter de secretario el segundo, para proceder a efectuar el examen que para obtener el grado de

MAESTRO EN CIENCIAS
CON ESPECIALIDAD EN PROBABILIDAD Y ESTADÍSTICA

Sustenta

JUAN FRANCISCO MANDUJANO REYES

en cumplimiento con lo establecido en los reglamentos y lineamientos de estudios de posgrado del Centro de Investigación en Matemáticas, A.C., mediante la presentación de la tesis

"ANALYSIS OF SOME STATISTICAL PROPERTIES
OF NEURAL NETWORKS "

Los miembros del jurado examinaron alternadamente al (la) sustentante y después de deliberar entre sí resolvieron declararlo (a):

APROBADO

DR. ROGELIO RAMOS QUIROGA
Presidente

DR. OSCAR SUSANO DALMAU CEDEÑO
Secretario

DR. MARIO ALBERTO DÍAZ TORRES
Vocal

CIMAT
DIRECCIÓN GENERAL

DR. VÍCTOR MANUEL RIVERO MERCADO
Director General

Acknowledgments

Quiero agradecer a las siguientes personas:

A mis padres, Paula Reyes Murillo y Juan Mandujano Romero por su trabajo y esfuerzo para apoyarme incondicionalmente en todo lo que hago.

A mi asesor, el Dr. Víctor Manuel Pérez Abreu Carrión, por la ayuda, los consejos y la confianza durante la realización de esta tesis. Gracias a todo el CIMAT, particularmente a mis profesores y en especial al Dr. Mario Díaz, por los comentarios sobre este trabajo.

A mis amigos y compañeros del CIMAT, en especial y sin importacia en el orden: Fernando, Gerardo, Gilberto, Isaías, Jesús, Joshue, Marcos, Melinda y Saúl.

A Emily por apoyarme y ayudarme siempre.

Por último quiero agradecer a CONACYT por la beca otorgada para realizar estudios de maestía.

Abstract

This thesis provides a theoretical study of the statistical properties of some neural network models by means of random matrix theory and model selection tools. We discuss the concentration inequalities approach, established in *Louart, Liao and Couillet* [1] in 2018 to study the performance of extreme learning machines. Some limiting spectral results of certain matrices, presented in *Pennington and Worah* [2] in 2017 and *Benigni y Péché* [3] in 2019, are studied, as well as two novel applications of a model selection approach to select a hyperparameter in extreme learning machines. Moreover, some original and applicable results about new and useful activation functions are presented, as well as conjectures related to the speed of the training of deep neural networks.

Key words

Extreme Learning Machine, Deep Learning, Random Matrix Theory, Model Selection, Concentration Inequality, Generalized Information Criterion, Generalized Cross-Validation, Marchenko–Pastur Law.

Index

CHAPTER 1

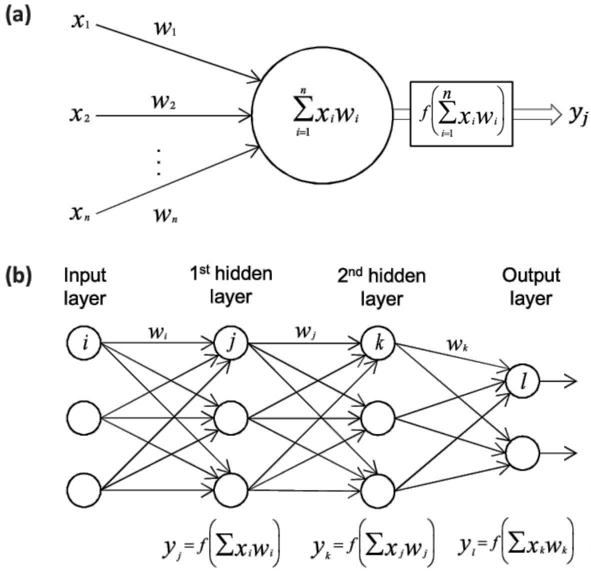
Introduction

In the mid-twentieth century, with the work of *Frank Rosenblatt*, the artificial neural networks emerged. The work of the psychologist *Donnald Hebb* and the increasing interest in using computers motivated the development of computational algorithms that, in a sense, imitate human learning. In 1958, *Frank Rosenblatt* made the first precursor of neural networks, the perceptron, which will be presented later in this section. Work of *Minsky and Papert* in 1969 showed that the perceptron was not able to solve some useful and easy problems in computer science, such as the learning of a linear function. Therefore, the perceptron was forgotten for more than a decade.

In 1975, *Paul Werbos* proposed the backpropagation method. However, it was not until 1985 that it was completely understood. At that point, there was a resurgence of neural networks because backpropagation could answer certain questions about the perceptron. Since then, there have been continuous advances in this area. Neural network applications in computer vision and machine learning in general, such as *Krichenvsky, Sutskever and Hilton* in 2012 and *Schmidhuber* in 2015, have provoked a really strong research interest. Nevertheless, the progress in neural networks has been achieved by the power of modern computers and the availability of large datasets rather than by mathematical and statistical results. Very recently, some authors, such as *Louart, Liao and Couillet* [1] in 2018 and *Pennington and Worah* [2] in 2017, have claimed that there is a lack of appropriate theoretical tools to completely understand these networks.

The subject of this thesis is a theoretical study of some neural network models by means of random matrix theory and model selection tools. Specifically, we use concentration inequalities and limiting spectral results of certain matrices, as well as a novel application of an information criterion and a cross-validation type of approach to select a hyperparameter. Moreover, some original and applicable results about some new and useful activation functions are presented, as well as some conjectures.

In order to understand the goal of this thesis, we first have to state the models considered. In the image below we can observe the structure of a neural network:



[1]

In a) we have a perceptron or neuron: here, a weighted $(w_1, ..., w_n)$ sum of the characteristics $(x_1, ..., x_n)$ of a datum is evaluated by a function (f), called the activation function, in order to explain a feature (y_j).

In b) we can see a neural network: an arrangement of neurons in layers connected by weights (w_i). A bias in each neuron is considered too. The first layer is the input layer, which has the data. In the output layer, we have the output of the algorithm.

For example, if we are considering an image classification task, the input layer has the

[1] Vieira, S., Pinaya, W. H., & Mechelli, A. (2017). Using deep learning to investigate the neuroimaging correlates of psychiatric and neurological disorders: Methods and applications. *Neuroscience & Biobehavioral Reviews*, 74, 58–75.

pictures and the output layer has the possible categories from which the network will choose. This classification can be a multiclass classification.

The neural network process consists of two phases. The training phase is when, on a known dataset, we tune (learn) the parameters (weights and bias) of the network by an optimization algorithm, such as stochastic gradient descent (SGD), in order to get the least misclassification error in terms of a loss function. The testing phase is when, fixing the parameters, we use a different dataset to study the performance of the algorithm by its misclassification error. From a statistical point of view, the training dataset is our data with which to tune the parameters of the model in order to minimize a loss function, and the testing dataset is a new dataset which will help to study how good is our model to solve a specific task (for example, classification).

In the early twenty-first century, the concept of a random neural network was developed. This is basically a neural network with some random variations, in our case, this will mean with random weights. This type of neural network is very popular because of its ability to solve optimization problems. The smaller number of trainable parameters is another attractive feature of this kind of network. In *Rahimi and Recht* in 2007 and *Saxe et al.* in 2011, it was stated that well designed randomly connected neural networks can achieve performances close to those of classic neural networks. Moreover, in *Cambria et al.* in 2015, it was established that intelligently designed single-layer random neural networks can reach superhuman capabilities, as presented in [1].

On the other hand, most of the recent neural network models initialize their parameters with random weights. This could be seen as the neural networks with random weights defining the initial loss landscape of the optimization. In *Pennington and Worah* [2] in 2017 and *Pennington and Bahri* [6] in 2017, it was claimed that these networks are closely related to random feature methods. They said there are important roles for these models in the field of neural networks, so they are important objects of study. The article [2], and our applications in Chapter 5, base their study on a physics paradigm: when we want to study a large and complex system, we make the assumption that its components are random variables, so we can obtain useful results using probability theory. By analogy, in nuclear physics, the Schödringer operator associated with an atom with a heavy nucleus can be replaced by a Hermitian random matrix, so the eigenvalues of this matrix correspond to the observed energy levels[2]. Modern neural network models are complex and large systems, so, as is stated in [2], it is natural to think about what insights we can obtain considering their components as random variables.

Let us note that we can describe the neural network process using matrices. For example, the input and output of each layer in the network are matrices. Thus, the random neural networks considered in the first part of this thesis, as well as the neural networks with initialized random weights used in the last chapter of this thesis, can be approached in terms of an initial random matrix model. Although these models involve random matrices, the application of the vast theory of random matrices to these systems is not straightforward. The main problem is that in many applications the activation functions are nonlinear. This nonlinearity could induce nonlinear dependence in the entries of the random matrices used to analyze the networks. Two approaches to deal with this problem are discussed in this work: using concentration inequalities, proposed by *Louart, Liao and Couillet* [1] in 2018 and *Louart and Couillet* [7] in 2018, and using the so called method of moments for random matrices, established in [2].

The goals of this thesis are the following: first, to discuss and summarize the two random matrix approaches above and show how these can help to obtain applicable results to neural networks. Second, to propose a novel information criterion and a cross-validation type method to select a hyperparameter considered in the model from [1]. Finally, based on results from *Beningni y Péché* [3] in 2019, to present a conjecture on the spectral distribution of the covariance output matrix of a multilayer neural network and its implications on training speed.

This thesis is organized as follows. In Chapter 2, "Preliminaries on Probability and Random Matrices," the main tools to understand the theoretical results of this thesis are presented:

[2] Wigner, Eugene P. "Random matrices in physics." *SIAM Review* 9.1 (1967): 1–23.

one part is on the more classic probability tools, such as concentration inequalities and sub-Gaussian random variables. Another section is on random matrix theory, another for the empirical spectral distribution, the Stieltjes transform, and the Marchenko–Pastur Theorem.

Chapter 3 summarizes the results of [1]: the concentration inequalities approach is used to obtain estimates of the asymptotic performance of a single layer random neural network. This is achieved using concentration results on a key matrix for the training and testing errors. We discuss here two particular important results in the random matrix area. The first one is a concentration inequality for quadratic forms for vectors with nonlinear dependent entries. The second one is a kind of Marchenko–Pastur theorem on the data, i.e., a characterization of its empirical asymptotic spectral distribution by means of its Stieltjes transform.

In [1] it is stated that its model can be seen naturally as a random ridge regression. Thus, this model employs a regularization hyperparameter. In Chapter 4 we propose two original applications of model selection to select this hyperparameter. This is a traditional statistical approach, first developed by the Generalized Information Criterion (GIC) and then by Generalized Cross-Validation (GCV). GIC is an information criterion based on bias correction for models that use estimation procedures more general than the maximum likelihood estimation. GCV can be seen as a weighted version of the ordinary cross-validation method. Our proposed results can be seen as a complement to those of [1] concerning the selection of the hyperparameter. It is important to note that these two methods, presented in Chapter 4, are developed for a dataset and do not consider the classical training and testing datasets in the field of neural networks. Nevertheless, we adopt this nomenclature in our statistical approach.

Finally, in Chapter 5, we first study some closely related work to the aforementioned kind of Marchenko–Pastur theorem in [1]. These results, due to [2] and [3], use the method of moments as the main tool. The first theorem on the asymptotic empirical spectral distribution of the covariance output matrix of a single layer neural network with random weights is presented. This theorem was proposed in [2]. They observed that under certain hypotheses on the properties of the activation function, the asymptotic empirical spectral distribution of the input covariance matrix of the network is conserved. Also, it was conjectured that this property holds for multilayer neural networks with random weights. They claimed that this feature could be beneficial for an increase of the training speed, an interesting property in the field of deep learning. In a recent work, in 2019, *Benigni and Péché* [3] proved this conjecture for a specific class of activation functions. In this part of the thesis, we present a suitable modification of this result. Moreover, we present a method for developing new useful activation functions based on classical activation functions. In order to show how these practical random matrix results can be applied to deep learning, in the final part of this chapter, some practical results are presented: we carried out experiments with different activation functions that pertain to our approach and the approach in [3]. We can conclude that our approach has important implications for the speed of training of deep neural networks.

The code used for the practical outcomes in this thesis can be found in a repository on GitHub.

CHAPTER 2

Preliminaries on Probability and Random Matrices

In this chapter we present the main definitions that will be used throughout this thesis. We first present the basic probability tools, and then some of the definitions and results of random matrix theory.

2.1 Probability Tools

2.1.1 Concentration Inequalitites

Let us first define the concentration inequalities:

Definition 1. *Let $x_1, ..., x_n$ be random variables with values in $\mathcal{X}$. Let $\zeta : \mathcal{X}^n \to \mathbb{R}$ be a function and $Z = \zeta(x_1, ..., x_n)$. A concentration inequality for Z is a bound like:*

$$\mathbb{P}(Z - \mathbb{E}(Z) \geq t) \leq g_z(t)$$

for $0 \leq t \leq t_0$. The left bound is defined in a similar way.

In order to illustrate this definition, consider X a nonnegative random variable and $t \in \mathbb{R}^+$. Markov's inequality ensures that

$$\mathbb{P}(X \geq t) \leq \frac{\mathbb{E}[X]}{t}.$$

Even more, if h is a strictly increasing nonnegative function, then

$$\mathbb{P}(h(X) \geq h(t)) \leq \frac{\mathbb{E}[h(X)]}{h(t)}.$$

Using the function $h(x) = x^2$ we get Chebyshev's inequality:

$$\mathbb{P}(|X - \mathbb{E}[X]| \geq t) = \mathbb{P}(|X - \mathbb{E}[X]|^2 \geq t^2) \leq \frac{\mathbb{E}[|X - \mathbb{E}[X]|^2]}{t^2} = \frac{\mathbb{V}[X]}{t^2}$$

Thus, we have found a concentration inequality for X.

2.1.2 Sub-Gaussian Random Variables

In many parts of this thesis we will use the concept of a sub-Gaussian random variable. We can think of this as a centered random variable such that its distribution tails decay at least as fast as a Gaussian distribution. A formal definition is

Definition 2. *Let W be a real-valued random variable. We say W is sub-Gaussian if there is some $b > 0$ such that for every real t,*

$$\mathbb{E}[e^{tW}] \leq e^{\frac{b^2t^2}{2}}.$$

Also, we can say W is a sub-Gaussian random variable with parameter b.

As an example, let us consider $W \sim N(0, \sigma^2)$. It is not difficult to show that for any $t \in \mathbb{R}$,

$$\mathbb{E}[e^{tW}] \leq e^{\frac{\sigma^2t^2}{2}}.$$

Thus, W is a sub-Gaussian random variable with parameter σ.

The following proposition confirms the intuition given previously.

Proposition. *If W is a sub-Gaussian random variable with parameter b, then*

$$\mathbb{E}[W] = 0, \text{ and } \mathbb{V}[W] \leq b^2.$$

The following proposition will be present in some assumptions in this thesis.

Proposition. *Let φ be a Lipschitz continuous function. If W is a standard Gaussian random variable, then $\varphi(W) - \mathbb{E}(\varphi(W))$ is a sub-Gaussian random variable.*

2.2 Random Matrix Theory (RMT)

2.2.1 The Empirical Spectral Distribution

We can see the empirical spectral distribution, for any square matrix, as a probability distribution that places equal mass on each of its eigenvalues.

Definition 3. *Let M be an $n \times n$ symmetric matrix, not necessarily random. Let $\lambda_j(M)$, for $j = 1, ..., n$, be the n eigenvalues of M, including multiplicity. The empirical spectral distribution of M is*

$$\rho_M(t) = \frac{1}{n} \sum_{j=1}^{n} \mathbb{1}_{\lambda_j(M) \leq t}$$

If the limit exists (in the weak almost surely sense), the limiting spectral density is defined as

$$\rho(t) = \lim_{n \to \infty} \rho_M(t).$$

Thus, the empirical spectral distribution of a square matrix evaluated at t counts the number of eigenvalues less than or equal to t.

2.2.2 Stieltjes Transform

The Stieltjes Transform will be used in many theorems in this work. It is used to characterize the asymptotic empirical spectral distribution of certain matrices in Chapter 3 and Chapter 5. The Marchenko–Pastur Theorem is an example of how this can be achieved.

Definition 4. *For ρ a positive finite measure on $\mathbb{R}$, we define the Stieltjes transform $g_\rho : \mathbb{C}_+ \to \mathbb{C}_+$ of ρ by*

$$g_\rho(z) = \int_{\mathbb{R}} \frac{1}{x-z} d\rho(x)$$

where $\mathbb{C}_+ = \{z \in \mathbb{C} | Im(z) > 0\}$.

Let us note that the Stieljtes transform can be seen as a moment generating function: Let ρ be a probability measure with support a compact set $[-R, R]$. The geometric series ensures that

$$\begin{aligned} g_\rho(t) &= \int_{-R}^{R} \frac{1}{x-t} d\rho(x) \\ &= -\int_{-R}^{R} \sum_{n=o}^{\infty} \frac{x^n}{t^{n+1}} d\rho(x) \\ &= -\sum_{n=o}^{\infty} \frac{1}{t^{n+1}} \int_{-R}^{R} x^n d\rho(x) \\ &= -\sum_{n=o}^{\infty} \frac{m_n}{t^{n+1}} \end{aligned}$$

for $t \in \mathbb{C}_+$ with $|t| > R$. Therefore, under these conditions, g_ρ is a power series in $\frac{1}{t}$ whose coefficients are the moments of ρ. This is a tool which will be used in Chapter 5 for computing Stieltjes transforms.

The following two theorems are important for characterizing an asymptotic measure by its asymptotic Stieltjes transform.

Theorem 2.2.1. *Let $\rho, \rho_1, \rho_2, ...$ be probability measures. Then $\rho_n \overset{w}{\to} \rho$ if and only if $g_{\rho n}(t) \xrightarrow{n\to\infty} g_\rho(t)$ for all $t \in \mathbb{C}_+$.*

Theorem 2.2.2. ***Inversion formula****: Let ρ be a probability measure, and $a, b \in \mathbb{R}$. Then*

$$\frac{1}{2}[\rho(\{a\}) + \rho(\{b\})] + \rho((a,b)) = \lim_{y \downarrow 0} \frac{1}{\pi} \int_a^b Im[g_\rho(x+iy)]dx.$$

When ρ is absolutely continuous with respect to Lebesgue measure with density f_ρ, then

$$f_\rho(x) = \frac{1}{\pi} \lim_{y \downarrow 0} Im[g_\rho(x+iy].$$

This property ensures that there is a one-to-one correspondence between finite measures on $\mathbb{R}$ and Stieltjes transforms.

2.2.3 The Marchenko–Pastur Theorem

Let us make the following definition.

Definition 5. *Let X an $n \times p$ matrix with independently identically distributed centered entries with variance equal to 1. We define a Wishart type matrix B as the following $n \times n$ matrix:*

$$B = \frac{1}{p} X X^t.$$

If $x_1, ..., x_n$ are the columns of X_n, we can rewrite B as

$$B = \frac{1}{p} \sum_{k=1}^{p} x_k x_k^t.$$

Let us recall that if $\lambda_1, ..., \lambda_n$ are the eigenvalues of B, we can write the empirical spectral measure as

$$\rho_B = \frac{1}{n} \sum_{k=1}^{n} \delta_{\lambda k},$$

where δ_* is the Dirac delta. The following theorem give us the asymptotic empirical spectral distribution of B when $n, p \to \infty$ at the same rate:

Theorem 2.2.3. ***The Marchenko–Pastur Theorem**. Let $(x_{ij})_{i,j}$ be a family of independent identically distributed random variables (such as in the previous definition) such that $\mathbb{E}[x_{11}] = 0$ and $\mathbb{E}[x_{11}^2] = \sigma^2$ is finite. If $n, p \to \infty$ such that $\frac{n}{p} \to c \in (0, \infty)$, then*

$$\rho_B \overset{w}{\to} \rho_{MP} \textit{ almost surely,}$$

whose Stieltjes transform, $g(z)$, satisfies the following equation

$$z g(z)^2 + (z - c + 1) g(z) + 1 = 0,$$

with solution

$$g(z) = \frac{\sqrt{(z-\lambda^-)(z-\lambda^+)}}{2z} - \frac{1}{2} - \frac{1-c}{2z}$$

where $\lambda^{\pm} = \sigma^2 (1 \pm \sqrt{c})^2$.

Let us note that using the inversion formula we can get the distribution of ρ_{MP}, which is given by

$$\big(1-\frac{1}{c}\big)_+\delta_0(dx)+\frac{1}{2\pi c\sigma^2 x}\sqrt{(\lambda^+-x)(x-\lambda^-)}\mathbb{1}_{[\lambda-,\lambda+]}(x)dx$$

where $(.)_+ = \max(0,.)$. Many theorems similar to this will be presented in Chapter 5.

CHAPTER 3

Extreme Learning Machine Performance via RMT

In this chapter we summarize and discuss the results of [1]. In order to study the asymptotic performance of a single-layer random neural network (Extreme Learning Machine), a concentration inequality approach is presented. We first have to define the notation used in this chapter.

Notation

- $||.||$ denotes the Euclidean vectorial norm for vectors and the linear operator norm for matrices.
- $||.||_F$ denotes the Frobenius norm for matrices:

  $$||A||_{\mathrm{F}} = \sqrt{\sum_{i=1}^{m}\sum_{j=1}^{n}|a_{ij}|^2} = \sqrt{tr\,(A^*A)}$$

- $\sigma(M)$ denotes the entry-wise application of a function $\sigma : \mathbb{R} \to \mathbb{R}$ to the matrix M.
- $\sigma(v)$ denotes the entry-wise application of a function $\sigma : \mathbb{R} \to \mathbb{R}$ to the vector v.

- $\Im(a)$ denotes the imaginary part of the complex number a.

- 1_T denotes the T-dimensional vector with all its entries equal to 1.

For $a = \phi(b) \in \mathbb{R}^l$ with $l \geq 1$ and $b \sim N(0, I_l)$, we write $a \sim N_\phi(0, I_l)$.

3.1 The Model

Having stated the notations let us define the model and its process used in this chapter. We first describe the neural network model, then the training and testing phases are stated.

3.1.1 The neural network

We can view our random neural network as a ridge regression task on random feature maps. Let T be the size of the input training set and p the dimension of each input datum. We consider a single layer neural network with n neurons. Let $\sigma : \mathbb{R} \to \mathbb{R}$ be the activation function. In the next section we will impose some conditions on σ.

The process of the network is as follows:

- We multiply each $x \in \mathbb{R}^p$ by a random matrix $W \in \mathbb{R}^{n \times p}$, which yields the vector $Wx \in \mathbb{R}^n$. Note that this is a random neural network approach because we have a random weights matrix.

- We apply $\sigma : \mathbb{R} \to \mathbb{R}$ to Wx to get the vector $\sigma(Wx) \in \mathbb{R}^n$.

- Then the output is $\beta^T \sigma(Wx)$, with $\beta \in \mathbb{R}^{n \times d}$ a matrix to be designed. We call β the regression matrix.

In the process of using a neural network we have two phases. The first is called the **training** phase; here we learn (tune/estimate) the regression matrix β, using a known input–output dataset (X, Y), by minimizing a loss function (the mean square error with some regularization factor). This is similar to a classical ridge regression task. The second phase is called the

testing phase; here we evaluate how good the selection of β has been. Then we fix β and the network operates on a new input dataset $\hat{X}$ corresponding to new unknown output dataset $\hat{Y}$. It is of interest to evaluate the mean square error in this phase.

3.1.2 Training phase

In the training phase, we have:

- Known input data, a matrix $X = [x_1, ..., x_T] \in \mathbb{R}^{p\times T}$
- Known output data, a matrix $Y = [y_1, ..., y_T] \in \mathbb{R}^{d\times T}$
- We are looking for β that minimizes the loss function

$$\ell(\beta) = \frac{1}{T}\sum_{i=1}^{T} ||\beta^T\sigma(Wx_i) - y_i||^2 + \gamma||\beta||_F^2,$$

with $\gamma > 0$ being some regularization factor to be selected.

It is well known that the regularization factor γ is very important to the ridge regression. It is intuitive to think that a small γ can minimize the loss function; neverthless, this could result in a large test error. On the other hand, large values for γ induce over-fitting. In the future, we will find a way to adequately choose the factor γ.

The solution of the minimization problem is:

$$\beta = \frac{1}{T}\Sigma(\frac{1}{T}\Sigma^t\Sigma + \gamma I_T)^{-1}Y^t = \frac{1}{T}(\frac{1}{T}\Sigma\Sigma^t + \gamma I_n)^{-1}\Sigma Y^t,$$

with $\Sigma = \sigma(WX)$. We write

$$Q = (\frac{1}{T}\Sigma^t\Sigma + \gamma I_T)^{-1}.$$

Some of the literature refers to Q as the resolvent of $\frac{1}{T}\Sigma^t\Sigma$. Thus we can define the **training error** as

$$E_{train} = \frac{1}{T}||Y^T - \Sigma^t\beta||_F^2.$$

It is easy to show that, rewriting β in terms of Q, we can write

$$E_{train} = \frac{\gamma^2}{T}tr(Y^tYQ^2).$$

This is the expression of E_{train} that will be used from now on. Note that this expression depends on the selection of γ.

3.1.3 Testing phase

Now it is time to state the second phase components of the network. In the testing phase, we have:

- Known input data, a matrix $\hat{X} \in \mathbb{R}^{p\times\hat{T}}$.
- Unknown output data, a matrix $\hat{Y} \in \mathbb{R}^{d\times\hat{T}}$.

Note that T may be different than $\hat{T}$. With the regression matrix β fixed, we can define the test error as

$$E_{test} = \frac{1}{\hat{T}}||\hat{Y}^T - \hat{\Sigma}^T\beta||_F^2,$$

which corresponds to the mean-square error, where $\hat{\Sigma} = \sigma(W\hat{X})$.

Note that β is the matrix learned in the training phase; thus, it depends only on X, Y and γ. We want to determine a γ that minimizes E_{test}. When E_{test} is small, we can say that the network has good *generalization performance*. In the *testing phase results* Section (3.4) we will formulate a conjecture about E_{test} that, if proved, would allow us to satisfactorily choose the regularization factor γ.

3.2 Assumptions

To obtain the principal results, we need to make the following three assumptions:

1. Sub-Gaussian W: Let W be the result of evaluating a Lipschitz function ϕ on a Ginibre matrix with standard Gaussian entries, i.e.,

$$W = \phi(\tilde{W}),$$

where $\tilde{W}$ has independent and identically distributed $N(0,1)$ entries, with ϕ a λ_ϕ-Lipschitz function.

2. The function σ: The activation function $\sigma : \mathbb{R} \to \mathbb{R}$ is λ_σ-Lipschitz.

   This is usual for many of the activation functions used in practice, such as sigmoid functions, rectified linear unit, or the absolute value operator.

3. Growth rate: As $n \to \infty$,

$$0 < \liminf_n \min\{\frac{p}{n}, \frac{T}{n}\} \leq \limsup_n \max\{\frac{p}{n}, \frac{T}{n}\} < \infty,$$

   while $\gamma, \lambda_\phi, \lambda_\sigma > 0$, for d constant. Moreover, $\limsup_n ||X|| < \infty$ y $\limsup_n \max_{i,j} |Y_{ij}| < \infty$.

3.3 Training phase results

In this section, our main object of study is the random variable training error:

$$E_{train} = \frac{\gamma^2}{T} tr(Y^T Y Q^2).$$

We want to establish an asymptotic estimate of E_{train}. Under Assumption 3 (growth rate), we will get that E_{train} concentrates around its mean. Thus, $\mathbb{E}[Q^2]$ will be a central object in the asymptotic evaluation of E_{train}.

We establish the following notation to summarize some important quantities. Also, we introduce some key matrices for the main result about the training error.

Notation

- n is the number of neurons in our present random neural network model.
- T is the number of elements in the input training dataset.
- p is the dimension of each input datum.
- γ is the regularization factor.

Definition 6. *For $w \sim N_\phi(0, I_p)$ and $X = [x_1, ..., x_T] \in \mathbb{R}^{p\times T}$, the input data matrix, we define*

$$\Phi = \mathbb{E}(\sigma(w^T X)^T \sigma(w^T X)),$$

and

$$\bar{Q} = (\frac{n}{T}\frac{\Phi}{1+\delta} + \gamma I_T)^{-1},$$

where δ is the unique positive solution of $\delta = \frac{1}{T}tr[\Phi\bar{Q}]$. Finally, we define

$$\Psi = \frac{n}{T}\frac{\Phi}{1+\delta}.$$

The following theorem provides us an asymptotic evaluation of the training mean-square error of a single-layer random neural network. This is the application result we are looking for.

Theorem 3.3.1. ***(Under Assumptions 1, 2, and 3)*** *For every $\varepsilon > 0$, almost surely*

$$n^{\frac{1}{2}-\varepsilon}(E_{train} - \bar{E}_{train}) \to 0,$$

where

$$\bar{E}_{train} = \frac{\gamma^2}{T}tr\left[Y^T Y\bar{Q}\Big[\frac{\frac{1}{n}tr(\Psi\bar{Q}^2)}{1-\frac{1}{n}tr(\Psi\bar{Q})^2}\Psi + I_T\Big]\bar{Q}\right].$$

To achieve this goal, as we said previously, both $\mathbb{E}[Q^2]$ as well as $\mathbb{E}[Q]$ need to be estimated. In order to get these estimates, the following two theorems state the asymptotic equivalence for $\mathbb{E}[QAQ]$ and $\mathbb{E}[Q]$, for certain matrix A. Note that to evaluate practically Theorem 3.3.1, it is necessary to estimate the value of Φ for various popular σ activation functions. In a subsequent section we will discuss this estimation.

Theorem 3.3.2. ***(Under Assumptions 1, 2, and 3)*** *Let $A \in \mathbb{R}^{T\times T}$ be a symmetric nonnegative definite matrix which is either Φ or a matrix with uniformly bounded operator norm. Then, for all $\epsilon > 0$, there exists $c > 0$ such that, for all n*

$$||\mathbb{E}[QAQ] - \Big(\bar{Q}A\bar{Q} + \frac{\frac{1}{n}tr(\Psi\bar{Q}A\bar{Q})}{1-\frac{1}{n}tr\Psi^2\bar{Q}^2}\bar{Q}\Psi\bar{Q}\Big)|| \leq cn^{-\frac{1}{2}+\epsilon}.$$

For $\mathbb{E}[Q]$ we have the following result provided by the standard resolvent approach of random matrix theory.

Theorem 3.3.3. ***(Under Assumptions 1, 2, and 3)*** *For any $\varepsilon > 0$, there exists $c > 0$ such that*

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

Using this theorem along with a concentration result on $\frac{1}{T}trQ$, we have the following theorem on the spectral measure of $\frac{1}{T}\Sigma^T\Sigma$ wich can be seen as a nonlinear extension of the Marchenko–Pastur theorem.

Theorem 3.3.4 (Nonlinear M–P extension)**.** ***(Under Assumptions 1, 2, and 3)*** *Let $\lambda_1, ..., \lambda_T$ be the eigenvalues of $\frac{1}{T}\Sigma^T\Sigma$ and $\mu_n = \frac{1}{T}\sum_{i=1}^T \delta_{\lambda_i}$. Then, for any bounded continuous function f, with probability 1*

$$\int f d\mu_n - \int f d\bar{\mu}_n \to 0.$$

Where $\bar{\mu}_n$ is the measure defined through its Stieltjes transform: for $z \in \{a \in \mathbb{C}, \Im(a) > 0\}$

$$m_{\bar{\mu}_n}(z) = \frac{1}{T}tr(\frac{n}{T}\frac{\Phi}{1+\delta_z} - zI_T)^{-1},$$

with δ_z the unique solution in $\{a \in \mathbb{C}, \Im(a) > 0\}$ of

$$\delta_z = \frac{1}{T}tr\Phi(\frac{n}{T}\frac{\Phi}{1+\delta_z} - zI_T)^{-1}.$$

To prove Theorem 3.3.3 with the standard resolvent approach, we need a convergence of quadratic forms based on the row vectors of Σ [11]. This kind of result is usually obtained by exploiting the independence (or linear dependence) in the vector entries. The entries of $\sigma(X^Tw)$ are not independent; therefore, a different technique is required. To state the following nonasymptotic lemma, we use a concentration of measure approach [10].

Lemma 3.3.5 (Concentration of quadratic forms)**.** ***(Under Assumptions 1 and 2)*** *Suppose $A \in \mathbb{R}^{T\times T}$ satisfies $||A|| \leq 1$ and, for $X \in \mathbb{R}^{p\times T}$ and $w \sim N_\phi(0, I_p)$, define the random vector $\sigma = \sigma(w^TX) \in \mathbb{R}^T$. Then*

$$\mathbb{P}(|\frac{1}{T}\sigma^TA\sigma - \frac{1}{T}tr[\Phi A]| > t) \leq Ce^{-\frac{cT}{||X||^2\lambda_\phi^2\lambda_\sigma^2}\min(\frac{t^2}{t_0^2},t)},$$

for $t_0 = |\sigma(0)| + \lambda_\phi\lambda_\sigma||X||\sqrt{\frac{p}{T}}$ *and* $C, c > 0$ *independent of all other parameters.*

Adding Assumption 3, we have

$$\mathbb{P}(|\frac{1}{T}\sigma^T A\sigma - \frac{1}{T}tr[\Phi A]| > t) \leq Ce^{-cn\min(t,t^2)},$$

for some $C, c > 0$.

This lemma has been stated in a nonasymptotic random matrix regime, i.e., without Assumption 3, thus it has independent interest. The following lemma is the aforementioned result about the concentration on $\frac{1}{T}trQ$ used to obtain Theorem 3.3.4 [Nonlinear M–P extension], which is also interesting in Random Matrix Theory.

Lemma 3.3.6 (Concentration of the Stieltjes transform of μ_n)**.** ***Under Assumptions 1 and 2.***

For $z \in \mathbb{C} \setminus \mathbb{R}^+$*,*

$$\mathbb{P}\Big(|\frac{1}{T}tr(\frac{1}{T}\Sigma^T\Sigma - zI_T)^{-1} - \mathbb{E}[\frac{1}{T}tr(\frac{1}{T}\Sigma^T\Sigma - zI_T)^{-1}]| > t\Big) \leq Ce^{-\frac{cdist(z,\mathbb{R}^+)^2Tt^2}{\lambda_\sigma^2\lambda_\phi^2||X||^2}},$$

for some $C, c > 0$*, where* $dist(z, \mathbb{R}^2)$ *denotes the Hausdorff set distance. In particular, for* $z = -\gamma$ *with* $\gamma > 0$ *and under Assumption 3, we have*

$$\mathbb{P}\Big(|\frac{1}{T}trQ - \frac{1}{T}tr\mathbb{E}[Q]| > t\Big) \leq Ce^{-cnt^2}.$$

3.3.1 Sketch of the proof of Lemma 3.3.5

In order to show how to prove this lemma, we need to state the following classic theorem for the Lipschitz transformation of a vector with independent standard Gaussian entries.

Theorem 3.3.7 (Gaussian concentration inequality for Lipschitz functions)**.** [1]

Let $X_1, ..., X_n \sim N(0,1)$ *be iid real Gaussian variables, and let* $F : \mathbb{R}^n \to \mathbb{R}$ *be a* λ_F*−Lipschitz function. Then, for* $X = (X_1, ..., X_n)$ *and for all t,*

$$\mathbb{P}(|F(X) - \mathbb{E}F(X)| \geq t) \leq Ce^{-c\frac{t^2}{\lambda_F^2}},$$

for some absolute constants $C, c > 0$.

[1] Taken from Tao, 2012, theorem 2.1.12

We will use this theorem in some steps in the proof. Let us reformulate Lemma 3.3.5:

Lemma (**Under Assumptions 1 and 2**). Let $A \in \mathbb{R}^{T \times T}$ be such that $||A|| \leq 1$ and, for $X \in \mathbb{R}^{p \times T}$ and $w \sim N_\phi(0, I_p)$, define the random vector $\sigma = \sigma(w^T X) \in \mathbb{R}^T$. Then

$$\mathbb{P}(|\frac{1}{T}\sigma^T A\sigma - \frac{1}{T}tr[\Phi A]| > t) \leq Ce^{-\frac{cT}{||X||^2\lambda_\phi^2\lambda_\sigma^2}\min(\frac{t^2}{t_0^2},t)}$$

for $t_0 = |\sigma(0)| + \lambda_\phi\lambda_\sigma||X||\sqrt{\frac{p}{T}}$ and $C, c > 0$ independent of all other parameters.

Adding Assumption 3, we have

$$\mathbb{P}(|\frac{1}{T}\sigma^T A\sigma - \frac{1}{T}tr[\Phi A]| > t) \leq Ce^{-cn\min(t,t^2)}$$

for some $C, c > 0$.

Sketch of the proof:

Let us see that the application $w \mapsto \frac{1}{T}\sigma^t A\sigma$ is, in a sense, quadratic in w. For quadratic forms we do not have a Lipschitz application thus we cannot easily transfer a concentration of w to $\frac{1}{T}\sigma^t A\sigma$. Another approach is required. **We first will find a high probability bound on $\frac{1}{T}||\sigma||$ by a concentration inequality.**

Note that the function $\Psi : \mathbb{R}^p \to \mathbb{R}^T$, defined by $\Psi(\tilde{w}) = \frac{1}{\sqrt{T}}\sigma(\phi(\tilde{w})^t X)^t$, is a $\frac{1}{\sqrt{T}}\lambda_\sigma\lambda_\phi||X||$-Lipschitz function. Therefore we can use Theorem 3.3.7 to obtain:

$$\begin{aligned}\mathbb{P}(\big|||\Psi(\tilde{w})|| - \mathbb{E}[||\Psi(\tilde{w})||\big| \geq t]) &= \mathbb{P}(\big|||\frac{1}{\sqrt{T}}\sigma(w^t X)|| - \mathbb{E}[||\frac{1}{\sqrt{T}}\sigma(w^t X)|| \geq t]\big|) \\ &\leq Ce^{-c\frac{t^2 T}{\lambda_\sigma^2\lambda_\phi^2||X||^2}}\end{aligned}$$

for some $C, c > 0$ independent of all parameters.

The map $w \mapsto \sigma(w^t X)$ is also Lipschitz, so

$$\big|\|\sigma(w^t X)\| - \|\sigma(0)1_t^t\|\big| \leq \|\sigma(w^t X) - \sigma(0)1_t^t\| \leq \lambda_\sigma\|w\|\|X\|.$$

Since $\tilde{w} \sim N(0, I_p)$, we have

$$\mathbb{E}[\|\phi(\tilde{w})\|^2] \leq \lambda_\phi^2\mathbb{E}[\|\tilde{w}\|^2] = \lambda_\phi^2 p.$$

Now, using Jensen's Inequality,

$$\begin{aligned}\mathbb{E}\Big[\|\frac{1}{\sqrt{T}}\sigma(w^tX)\|\Big] &\leq |\sigma(0)| + \lambda_\sigma \mathbb{E}\Big[\frac{1}{\sqrt{T}}\|w\|\Big]\|X\| \\ &\leq |\sigma(0)| + \lambda_\sigma \sqrt{\mathbb{E}\Big[\frac{1}{T}\|w\|^2\Big]} \\ &\leq |\sigma(0)| + \lambda_\sigma\lambda_\phi \|X\| \sqrt{\frac{p}{T}}.\end{aligned}$$

Letting $t_0 = |\sigma(0)| + \lambda_\sigma\lambda_\phi\|X\|\sqrt{\frac{p}{T}}$, we find

$$\mathbb{P}(\|\frac{1}{\sqrt{T}}\sigma(w^tX)\| \geq t + t_0) \leq Ce^{-\frac{cTt^2}{\lambda_\phi^2\lambda_\sigma^2\|X\|^2}}.$$

For all $t \geq 4t_0$, we can obtain

$$\mathbb{P}(\|\frac{1}{\sqrt{T}}\sigma(w^tX)\| \geq t) \leq Ce^{-\frac{cTt^2}{2\lambda_\phi^2\lambda_\sigma^2\|X\|^2}}. \tag{3.1}$$

We define $\mathcal{A}_K = \{w : \|\sigma(w^tX)\| \leq K\sqrt{T}\}$. Partitioning on this event, we can show that the map $w \mapsto \frac{1}{\sqrt{T}}\sigma^tA\sigma$ is Lipschitz. There exists a $K \geq 4t_0$ such that

$$\mathbb{P}\Big(\big|\frac{1}{T}\sigma A\sigma^t - \frac{1}{T}tr[\Phi A]\big| > t\Big) \leq \mathbb{P}\Big(\{\big|\frac{1}{T}\sigma A\sigma^t - \frac{1}{T}tr[\Phi A]\big| > t\}, \mathcal{A}_K\Big) + \mathbb{P}(\mathcal{A}_K^c).$$

Because of (3.1) we can already bound $\mathbb{P}(\mathcal{A}_K^c)$. On the set $\mathcal{A}_K$, the function f, defined as $f(\sigma) = \sigma^tA\sigma$ is a Lipschitz map. Neverthless, the expression $\mathbb{P}\Big(\{\big|\frac{1}{T}\sigma A\sigma^t - \frac{1}{T}tr[\Phi A]\big| > t\}, \mathcal{A}_K\Big)$ does not allow applying Theorem 3.3.7. So, we consider instead $\tilde{f}$, a $K\sqrt{T}$-Lipschitz continuation to $\mathbb{R}^T$ of $f_{\mathcal{A}_K}$ (the restriction of f to $\mathcal{A}_K$).

Applying Theorem 3.3.7, we obtain

$$\mathbb{P}(|\tilde{f}(\sigma(w^tX)) - \mathbb{E}[\tilde{f}(\sigma(w^tX))]| \geq KTt) \leq e^{-\frac{cTt^2}{\|X\|^2\lambda_\sigma^2\lambda_\phi}}.$$

Then, we have

$$\begin{aligned}\mathbb{P}(\{|f(\sigma(w^tX)) - \mathbb{E}[\tilde{f}(\sigma(w^tX))]| \geq KTt\}, \mathcal{A}_K) &= \mathbb{P}(\{|\tilde{f}(\sigma(w^tX)) - \mathbb{E}[\tilde{f}(\sigma(w^tX))]| \geq KTt\}, \mathcal{A}_K) \\ &\leq e^{-\frac{cTt^2}{\|X\|^2\lambda_\sigma^2\lambda_\phi}}.\end{aligned}$$

Therefore, we need to bound the difference

$$\Delta = |\mathbb{E}[\tilde{f}(\sigma(w^tX))] - \mathbb{E}[f(\sigma(w^tX))]|.$$

Let μ_σ be the law of $\sigma(w^t X)$. Now, f and $\tilde{f}$ are equal on $\mathcal{A}_K$, and so

$$\Delta \leq \int_{\|\sigma\| \geq K\sqrt{T}} (|f(\sigma)| + |\tilde{f}(\sigma)|) d\mu_\sigma(\sigma).$$

Since $\|A\| \leq 1$, for $\|\sigma\| \geq K\sqrt{T}$, $\max(|f(\sigma)|, |\tilde{f}(\sigma)| \leq \|\sigma\|^2$. We can write

$$\begin{aligned}
\Delta &\leq 2\int_{\|\sigma\| \geq K\sqrt{T}} \|\sigma\|^2 d\mu_\sigma = 2\int_{\|\sigma\| \geq K\sqrt{T}} \int_{t=0}^{\infty} \mathbb{1}_{\|\sigma\|^2 \geq t} dt d\mu_\sigma \\
&= 2\int_{t=0}^{\infty} \mathbb{P}(\{\|\sigma\|^2 \geq t\}, \mathcal{A}_K^c) dt \\
&\leq 2\int_{t=0}^{K^2T} \mathbb{P}(\mathcal{A}_K^c) + 2\int_{t=K^2T}^{\infty} \mathbb{P}(\|\sigma(^tX)\|^2 \geq t) dt \\
&\leq 2K^2T\mathbb{P}(\mathcal{A}_K^c) + 2\int_{t=K^2T}^{\infty} Ce^{-\frac{ct}{2\lambda_\phi^2\lambda_\sigma^2\|X\|^2}} dt \\
&\leq 2CTK^2 e^{-\frac{cTK^2}{2\lambda_\phi^2\lambda_\sigma^2\|X\|^2}} + \frac{2C\lambda_\phi^2\lambda_\sigma^2\|X\|^2}{c} e^{-\frac{cTK^2}{2\lambda_\phi^2\lambda_\sigma^2\|X\|^2}}.
\end{aligned}$$

We need the fact that for $x \in \mathbb{R}$, $xe^{-x} \leq e^{-1} \leq 1$, and recalling that $K \geq 4t_0 \geq \lambda_\sigma\lambda_\phi\|X\|\sqrt{\frac{p}{T}}$, we obtain

$$\Delta \leq \frac{6C}{c}\lambda_\phi^2\lambda_\sigma^2\|X\|^2.$$

So, we have

$$\mathbb{P}(\{|f(\sigma(w^tX)) - \mathbb{E}[f(\sigma(w^tX))]| \geq KTt + \Delta\}, \mathcal{A}_K) \leq Ce^{-\frac{cTt^2}{\lambda_\phi^2\lambda_\sigma^2\|X\|^2}}.$$

As before, for $t \geq \frac{4\Delta}{KT}$,

$$\mathbb{P}(\{|f(\sigma(w^tX)) - \mathbb{E}[f(\sigma(w^tX))]| \geq KTt\}, \mathcal{A}_K) \leq Ce^{-\frac{cTt^2}{2\lambda_\phi^2\lambda_\sigma^2\|X\|^2}}.$$

We are almost done with the proof. To achieve this, we need an appropiate control of the concentration results. We need to avoid the condition $t \geq \frac{4\Delta}{KT}$ using the fact that probabilities are less than one. We want to replace C by λC with $\lambda \geq 1$, such that for $t \geq \frac{4\Delta}{KT}$

$$\lambda Ce^{-c\frac{Tt^2}{2\|X\|^2\lambda_\phi^2\lambda_\sigma^2}} \geq 1.$$

If we take for instance $\lambda \leq \frac{1}{C}e^{\frac{18C^2}{c}}$, the above inequality holds. Therefore, taking $\lambda = \max(1, \frac{1}{C}e^{\frac{18C^2}{c}})$, we get for every $t > 0$

$$\mathbb{P}(\{|f(\sigma(w^tX)) - \mathbb{E}[f(\sigma(w^tX))]| \geq KTt\}, \mathcal{A}_K) \leq \lambda Ce^{-\frac{cTt^2}{2\lambda_\phi^2\lambda_\sigma^2\|X\|^2}},$$

along with

$$\mathbb{P}(\mathcal{A}_K^c) \leq Ce^{-\frac{cTK^2}{2\lambda_\phi^2\lambda_\sigma^2\|X\|^2}},$$

we have

$$\mathbb{P}(|f(\sigma(w^tX)) - \mathbb{E}[f(\sigma(w^tX))]| \geq KTt) \leq (\lambda+1)Ce^{-\frac{cTt^2}{2\lambda_\phi^2\lambda_\sigma^2\|X\|^2}}.$$

Now, with $K = \max(4t_0, \sqrt{t})$,

$$\mathbb{P}(|f(\sigma(w^tX)) - \mathbb{E}[f(\sigma(w^tX))]| \geq KTt) \leq (\lambda+1)Ce^{-\frac{cT\min(\frac{t^2}{16t_0^2},t)}{2\lambda_\phi^2\lambda_\sigma^2\|X\|^2}}.$$

3.4 Testing phase results

To talk about the asymptotic estimate of

$$E_{test} = \frac{1}{\hat{T}}||\hat{Y}^T - \hat{\Sigma}^T\beta||_F^2,$$

where $\hat{\Sigma} = \sigma(W\hat{X})$, we first have to extend the previously established definitions of Φ and Ψ.

Definition 7. *For all pairs of matrices A and B with p rows and an arbitrary number of columns, we define*

$$\Phi_{AB} = \mathbb{E}[\sigma(w^TA)^T\sigma(w^TB)],$$

and

$$\Psi_{AB} = \frac{n}{T}\frac{\Phi_{AB}}{1+\delta},$$

where $w \sim N_\phi(0, I_p)$. In particular, $\Phi = \Phi_{XX}$ and $\Psi = \Psi_{XX}$.

The following unproven statement has been intuitively derived by the concentration arguments used to prove the training phase results.

Conjecture 3.4.1. ***(Under Assumptions 1 and 2)*** *If $\hat{X}$ and $\hat{Y}$ satisfy the same conditions as X and Y in Assumption 3, then, for all $\varepsilon > 0$*

$$n^{\frac{1}{2}-\varepsilon}(E_{test} - \bar{E}_{test}) \to 0$$

almost surely, with

$$\bar{E}_{test} = \frac{1}{\hat{T}}||\hat{Y}^T - \Psi_{X\hat{X}}^T \bar{Q} Y^T||_F^2 + \frac{\frac{1}{n} tr Y^T Y \bar{Q} \Psi \bar{Q}}{1 - \frac{1}{n} tr(\Psi \bar{Q})^2} \left[\frac{1}{\hat{T}} tr \Psi_{\hat{X}\hat{X}} - \frac{1}{\hat{T}} tr(I_T + \gamma \bar{Q})(\Psi_{X\hat{X}} \Psi_{\hat{X}X} \bar{Q}) \right].$$

As we said previously for Theorem 3.3.1, in order to make practical use of this result, we need to evaluate Φ_{AB} for some σ activation functions. In the following section we will provide an estimate for this matrix.

3.5 Estimation of Φ_{AB}

In this part we will present some evaluations of

$$\Phi_{AB} = \mathbb{E}[\sigma(w^t A)^t \sigma(w^t B)],$$

for arbitrary matrices A and B and different activation functions $\sigma(.)$, letting the mapping $\phi(.)$ be the identity. For more details, we refer the reader to section 3.3 of [1]. The evaluation depends on the study of the evaluation of its entries, i.e., let $a, b \in \mathbb{R}^p$ be arbitrary vectors, thus

$$\begin{aligned} \Phi_{ab} &= \mathbb{E}[\sigma(w^t a)^t \sigma(w^t b)] \\ &= (2\pi)^{\frac{-p}{2}} \int \sigma((\phi(\tilde{w}))^t a)^t \sigma((\phi(\tilde{w}))^t b). \end{aligned}$$

In [1], the evaluation is obtained for some of the most popular activation functions in neural networks through various integration tricks. The identity function, the erf, the absolute value function, and the famous rectified linear unit (ReLU) function are considered. Note that these functions satisfy Assumption 2, i.e., they are Lipschitz functions.

	$\sigma(t)$	Φ_{ab}
identity	t	$a^t b$
erf	$\frac{2}{\sqrt{\pi}} \int_0^t e^{-u^2} du$	$\frac{2}{pi}(\frac{2a^t b}{\sqrt{(1+2\\|a\\|^2)(1+2\\|b\\|^2)}})$
absolute value	$\lvert t \rvert$	$\frac{2}{\pi}\\|a\\|\\|b\\|(<(a,b)(<(a,b)) + \sqrt{1-<(a,b)^2})$
ReLU	$\max(t,0)$	$\frac{1}{2\pi}\\|a\\|\\|b\\|(<(a,b)(-<(a,b)) + \sqrt{1-<(a,b)^2})$

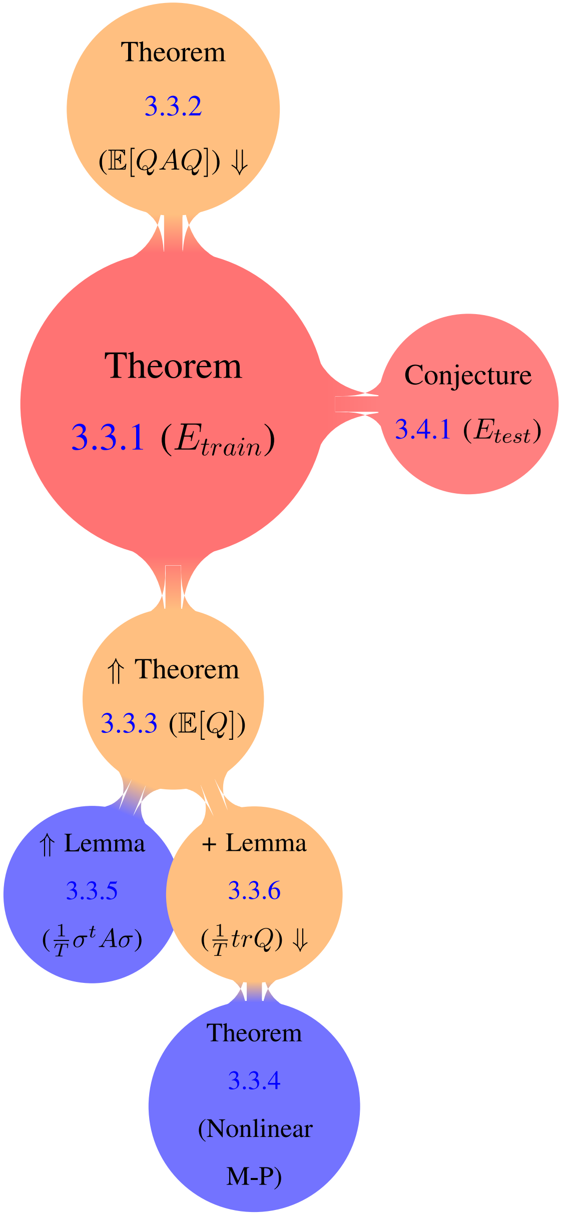
where

$$<(a,b) = \frac{a^t b}{\|a\|\|b\|}.$$

These evaluations allow us to apply the results of this section to real problems.

3.6 Diagram of Results

In the following diagram we can observe a diagram of the previously presented results from [1]. We have in red the Random Neural Networks area results, in blue the interesting Random Matrices area results, and in orange the auxiliary results which basically come from Random Matrix Theory.



Note that Conjecture 3.4.1 could help us to select the hyperparameter γ, as mentioned in [1]. However, a statistical criterion to select this regularization hyperparameter is developed in the next chapter.

CHAPTER 4

Model Selection for applications to Extreme Learning Machines

In the present section we will present two model selection approaches to selecting the regularization parameter γ. Our first results are based on a generalization of the classical technique of Akaike's information criterion (AIC). This framework is called the generalized information criterion (GIC) [4]. The development of the GIC had the purpose of obtaining an information criterion for models that employ estimation procedures other than the maximum likelihood method. The second result presented in this chapter is the so called Generalized Cross-Validation approach, which can be seen as a weighted version of the Ordinary Cross-Validation approach [5].

4.1 The Generalized Information Criterion (GIC)

The GIC was introduced by Konishi and Kitagawa [4] in 1996. It is known that the GIC can be applied to evaluate statistical models constructed by the maximum penalized likelihood procedure. Earlier in this thesis, we claimed that we can see our random neural network model as a mere linear ridge regression task on random feature maps. Thus, it is natural to consider the GIC as a potential technique for selecting the regularization parameter γ.

In order to study the application of the GIC to our neural network, we will describe the fundamentals of a functional statistics and robust estimators approach to use this technique. Also we will fit our random neural network model into the framework of the generalized information criterion.

4.1.1 Fundamentals of the GIC

In the process of statistical inference, we select an parametric family of probability distributions $\{f(x|\theta); \theta \in \Theta\}$ that serves as an approximation to the true distribution $G(x)$ that generates our data. The model parameter is estimated based on the data which comes from the true distribution $G(x)$, but not from $f(x|\theta)$.

Let T be a real-valued function defined on D, the set of all distributions on the sample space. In this framework, we assume that the parameter θ is given by a real-valued function of the distribution G, i.e., a functional $T(G)$. Given data $\{x_1, ..., x_n\}$, the estimator for θ is

$$\hat{\theta}(x_1, ..., x_n) = T(\hat{G}),$$

where G is replaced with the empirical distribution $\hat{G}$. We can say that the estimator depends on data only through the empirical distribution. Now we can state the following definition:

Definition 8. *Let D be the set of all distributions on $\mathbb{R}$. A* ***statistical functional*** *is a real-valued map $T : D \to \mathbb{R}$.*

From this point of view, we will define an object that describes the effect of an infinitesimal contamination at a point in the estimation procedure. This concept comes from the field of robust statistics.

Definition 9. *Given a functional $T(G)$, the* ***directional derivative*** *with respect to the distri-*

bution G is the real-valued function $T^{(1)}(x;G)$ that satisfies

$$\begin{aligned}\lim_{\varepsilon\to 0}\frac{T((1-\varepsilon)G+\varepsilon H)-T(G)}{\varepsilon} &= \frac{\partial}{\partial\varepsilon}\{T((1-\varepsilon)G+\varepsilon H)\}\Big|_{\varepsilon=0}\\ &= \int T^{(1)}(x;G)d\{H(x)-G(x)\},\end{aligned}$$

for any distribution $H(x)$. To ensure uniqueness, we have

$$\int T^{(1)}(x;G)dG(x)=0.$$

Then we can write

$$\lim_{\varepsilon\to 0}\frac{T((1-\varepsilon)G+\varepsilon H)-T(G)}{\varepsilon}=\int T^{(1)}(x;G)dH(x).$$

Remark. *Note that if we take H to be the delta function δ_x that has the probability of 1 at the point x, then we have*

$$\begin{aligned}\lim_{\varepsilon\to 0}\frac{T((1-\varepsilon)G+\varepsilon\delta_x)-T(G)}{\varepsilon} &= \frac{\partial}{\partial\varepsilon}\{T((1-\varepsilon)G+\varepsilon\delta_x)\}\Big|_{\varepsilon=0}\\ &= \int T^{(1)}(x;G)d\delta_x\\ &= T^{(1)}(x;G).\end{aligned}$$

In the field of robust statistics, $T^{(1)}(x;G)$ is called the ***influence function****.*

Now we have everything needed to define the generalized information criterion. Before stating the definition of the GIC, we have to remark on some theoretical aspects. As we said earlier in this section, the GIC can be seen as an extension of the AIC to a more general information criterion by relaxing the following assumptions made for the use of the AIC:

- estimation is by maximum likelihood,
- and we are working in a parametric family of distributions including the true model.

As we expected, the developement of the GIC is similar to that for the AIC. They are both bias correction based methods for the selection of models.

Definition 10. *Let $\{z_1, ..., z_n\}$ be data which comes from the true distribution G, and let $\hat{G}$ be the empirical distribution based on the data. Let $\{f(x|\theta); \theta \in \Theta \subset \mathbb{R}^p\}$ be an adopted parametric statistical model, with density f. For T, a p-dimensional statistical functional, let $\hat{\theta} = T(\hat{G})$ be an estimator for θ. An information criterion for evaluating the statistical model $f(x|\hat{\theta})$ is given by*

$$GIC = -2\sum_{i=1}^{n} \log f(z_i; \hat{\theta}) + \frac{2}{n}\sum_{i=1}^{n} tr\{T^{(1)}(z_i, \hat{G})\big(\frac{\partial \log f(z_i; \hat{\theta})}{\partial \theta}\big)^t\Big|_{\theta=\hat{\theta}}\}.$$

When we want to select the best model from various different models, we select the model for which the GIC is smallest. An application of this criterion is to select the hyperparameters of a statistical model.

4.1.2 GIC for M-estimator

We aim to use the generalized information criterion to select the regularization parameter γ in our present random neural network model. We previously said that our network can be seen as a linear ridge regression model on random feature maps. We will show that the estimators of this kind of linear models are a specific case of M-estimators. Then, we need to establish the expression of GIC for these M-estimators.

To achieve this objective we first have to define what an M-estimator is.

Definition 11. *Let $\{f(x; \theta) : \theta \in \Theta\}$ be a parametric statistical model for $\mathcal{X} = \{x_1, ..., x_n\}$ an independent dataset. Given a function $\psi : \mathcal{X} \times \Theta \to \mathbb{R}^p$, an M-estimator $\hat{\theta}$ is an estimator that solves the equation*

$$\frac{1}{n}\sum_{i=1}^{n} \psi(x_i; \theta) = 0.$$

Example. ***Maximum Likelihood Estimator***

Consider a probability distribution with density $f(x; \theta)$ ($\theta \in \Theta \subset \mathbb{R}$). Let $x_1, ..., x_n$ be n independent observations generated from this distribution (the true distribution). We estimate θ based on these n observations. The maximum likelihood estimator $\hat{\theta}_{ML}$ is given *by the solution of the equation*

$$\frac{1}{n}\sum_{i=1}^{n}\frac{\partial}{\partial\theta}\log f(x_i;\theta)=0.$$

Then, $\hat{\theta}_{ML}$ is an M-estimator associated to the function $\psi(x_i;\theta)=\frac{\partial}{\partial\theta}\log f(x_i;\theta)$.

Recall that the expression of GIC depends on the influence function $T^{(1)}(.,.)$ of the estimator. Now we will establish an expression for the influence function of a general M-estimator.

Theorem 4.1.1. *Given an independent dataset $Z=\{z_1,...,z_n\}$ generated from the true distribution G, let $\{f(x,\theta):\theta\in\Theta\subset\mathbb{R}^q\}$ be a parametric statistical model for the data. Let $\hat{\theta}$ be an M-estimator associated with the function $\psi:Z\times\Theta\to\mathbb{R}^q$, where $\hat{\theta}=T(\hat{G})$, for $\hat{G}$, the empirical distribution and T is a statistical functional. Then,*

$$T^{(1)}(x,G)=-\Big[\int\frac{\partial}{\partial\theta}\psi(z;\theta)\Big|_{\theta=T(G)}dG(z)\Big]^{-1}\psi(x;T(G)).$$

Using the result from Theorem 4.1.1 in Definition 10, we can obtain the GIC expression for M-estimators.

Theorem 4.1.2. *Let $\{z_1,...,z_n\}$ be data which comes from the true distribution G, and let $\hat{G}$ be the empirical distribution based on the data. Let $\{f(x|\theta);\theta\in\Theta\subset\mathbb{R}^p\}$ be an adopted parametric statistical model, with density f. For T, a p-dimensional statistical functional, let $\hat{\theta}=T(\hat{G})$ be an M-estimator for θ associated to the function $\psi:Z\times\Theta\to\mathbb{R}^q$. Then, the GIC for evaluating the statistical model $f(x|\hat{\theta})$ is given by*

$$GIC=-2\sum_{i=1}^{n}\log f(z_i;\hat{\theta})+2tr\{R(\psi,\hat{G})^{-1}Q(\psi,\hat{G})\},$$

where,

$$R(\psi,\hat{G})=-[\frac{1}{n}\sum_{i=1}^{n}\frac{\partial}{\partial\theta}\psi(z_i;\theta)\Big|_{\theta=\hat{\theta}}]^t$$

and

$$Q(\psi,\hat{G})=\frac{1}{n}\sum_{i=1}^{n}\psi(z_i;\theta)[\frac{\partial}{\partial\theta}\log f(z_i;\theta)]^t\Big|_{\theta=\hat{\theta}}.$$

4.2 The GIC for Extreme Learning Machines

We first will write our $\hat{\beta}$ as an M-estimator. Then, we will use Theorem 4.1.2 to find the GIC for our random neural network seen as a ridge regression task.

Recall that we have a deterministic data matrix $X = [x_1, ..., x_T] \in \mathbb{R}^{p\times T}$, a random weights matrix $W \in \mathbb{R}^{n\times p}$ and a Lipschitz activation function σ. For simplicity we will write the ridge regression task in terms of $z_i = \sigma(Wx_i)$, for each $i = 1, 2, ..., T$. We will study a Gaussian ridge regression task with known variance and another with an unknown variance.

4.2.1 Gaussian model with known variance

We consider the statistical model

$$\{f(y|z;\beta) : \beta \in \mathbb{R}^{n\times d}\},$$

where $f(y_i|z_i;\beta)$ is a normal density with mean $\beta^t z_i$ and known variance s. Thus, we have that

$$f(y_i|z_i;\beta) = (2\pi)^{-\frac{d}{2}}|sI_d|^{-\frac{1}{2}}e^{-\frac{1}{2}(y_i-\beta^t z_i)^t(sI_d)^{-1}(y_i-\beta^t z_i)},$$

and so

$$\log f(y_i|z_i;\beta) = -\frac{d}{2}\log(2\pi) - \frac{d}{2}\log(s) - \frac{1}{2s}(y_i - \beta^t z_i)^t(y_i - \beta^t z_i).$$

As we said earlier, we are looking for β that minimizes the loss function

$$\ell(\beta) = \frac{1}{T}\sum_{i=1}^{T} ||\beta^T z_i - y_i||^2 + \gamma||\beta||_F^2.$$

Differentiating $\ell(\beta)$ with respect to β we obtain the equation

$$2\gamma\beta + 2\frac{1}{T}\sum_{i=1}^{T} z_i(\beta^t z_i - y_i)^t = 0$$

$$\frac{1}{T}\sum_{i=1}^{T}[z_i z_i^t\beta - z_i y_i^t + \gamma\beta] = 0$$

$$\frac{1}{T}\sum_{i=1}^{T}[z_i z_i^t + \gamma I_n]\beta - z_i y_i^t = 0,$$

where I_n is the $n \times n$ identity matrix. Then, according to Definition 11, we can say $\hat{\beta} = \frac{1}{T}(\frac{1}{T}\Sigma\Sigma^t + \gamma I_n)^{-1}\Sigma Y^t$ is an M-estimator associated to the function $\psi(y_i|z_i;\beta) = [z_i z_i^t + \gamma I_n]\beta - z_i y_i$. In this case, the expression for the GIC is as follows

$$GIC = -2\sum_{i=1}^{T} \log f(y_i|z_i;\hat{\beta}) + 2tr\{R(\psi,\hat{G})^{-1}Q(\psi,\hat{G})\},$$

where

$$R(\psi,\hat{G}) = -[\frac{1}{T}\sum_{i=1}^{T}\frac{\partial}{\partial\beta}\psi(y_i|z_i;\beta)\Big|_{\beta=\hat{\beta}}]^t$$

and

$$Q(\psi,\hat{G}) = \frac{1}{T}\sum_{i=1}^{T}\psi(y_i|z_i;\beta)[\frac{\partial}{\partial\beta}\log f(y_i|z_i;\beta)]^t\Big|_{\beta=\hat{\beta}}.$$

Therefore, we need to compute the expressions $R(\psi,\hat{G})$ and $Q(\psi,\hat{G})$. We have

$$\frac{\partial}{\partial\beta}\psi(y_i|z_i;\beta)\Big|_{\beta=\hat{\beta}} = [z_i z_i^t + \gamma I_n],$$

and so

$$R(\psi,\hat{G}) = -[\frac{1}{T}\sum_{i=1}^{T}\frac{\partial}{\partial\beta}\psi(y_i|z_i;\beta)\Big|_{\beta=\hat{\beta}}]^t = -[\frac{1}{T}\Sigma\Sigma^t + \gamma I_n]. \tag{4.1}$$

Note that

$$\frac{\partial}{\partial\beta}\log f(y_i|z_i;\beta) = -\frac{1}{2s}(z_i z_i^t\beta - z_i y_i^t),$$

and so

$$\begin{aligned}\frac{1}{T}\sum_{i=1}^{T}\psi(y_i|z_i;\beta)[\frac{\partial}{\partial\beta}\log f(y_i|z_i;\beta)]^t\Big|_{\beta=\hat{\beta}} =& -\frac{1}{2s}\frac{1}{T}\sum_{i=1}^{T}[(z_i z_i^t + \gamma I_n)\hat{\beta} - z_i y_i^t][(z_i z_i^t\hat{\beta} - z_i y_i^t)]^t \\ =& -\frac{1}{2s}\frac{1}{T}\{\sum_{i=1}^{T}(z_i z_i^t\hat{\beta} - z_i y_i^t)(z_i z_i^t\hat{\beta} - z_i y_i^t)^t \\ &+ \gamma\sum_{i=1}^{T}\hat{\beta}(z_i z_i^t\hat{\beta} - z_i y_i^t)^t\}.\end{aligned}$$

We have, for Q,

$$Q(\psi,\hat{G}) = -\frac{1}{2s}\frac{1}{T}\{\sum_{i=1}^{T}(z_i z_i^t\hat{\beta} - z_i y_i^t)(z_i z_i^t\hat{\beta} - z_i y_i^t)^t + \gamma\sum_{i=1}^{T}\hat{\beta}(z_i z_i^t\hat{\beta} - z_i y_i^t)^t\}. \tag{4.2}$$

Now, substituting (4.3) and (4.4) into the expression for the GIC, we have

$$GIC = -2\sum_{i=1}^{T} -\frac{d}{2}\log(2\pi) - \frac{d}{2}\log(s) - \frac{1}{2s}(y_i - \hat{\beta}^t z_i)^t(y_i - \hat{\beta}^t z_i)+$$
$$2\frac{1}{2s}tr\{[\frac{1}{T}\Sigma\Sigma^t + \gamma I_n]^{-1}\frac{1}{T}\{\sum_{i=1}^{T}(z_i z_i^t\hat{\beta} - z_i y_i^t)(z_i z_i^t\hat{\beta} - z_i y_i^t)^t + \gamma\sum_{i=1}^{T}\hat{\beta}(z_i z_i^t\hat{\beta} - z_i y_i^t)^t\}\}.$$

We can leave out the terms $-\frac{d}{2}\log(2\pi)$, $-\frac{d}{2}\log(s)$ and s, because they do not depend on γ. Then, we can write

$$GIC = \sum_{i=1}^{T}(y_i - \hat{\beta}^t z_i)^t(y_i - \hat{\beta}^t z_i)+$$
$$tr\{[\frac{1}{T}\Sigma\Sigma^t + \gamma I_n]^{-1}\frac{1}{T}\{\sum_{i=1}^{T}(z_i z_i^t\hat{\beta} - z_i y_i^t)(z_i z_i^t\hat{\beta} - z_i y_i^t)^t + \gamma\sum_{i=1}^{T}\hat{\beta}(z_i z_i^t\hat{\beta} - z_i y_i^t)^t\}\}.$$

Now, we can choose γ by minimizing the GIC. We made a program to study the application of this result. The practical outcomes are presented in the next subsection.

4.2.2 Gaussian model with unknown variance

Before presenting the applications of the GIC for a single layer random neural network, we will obtain an expression for the GIC for a Gaussian model with unknown variance. This part is analogous to the previous one. In this case, we consider the statistical model

$$\{f(y|z;\beta,s) : \beta \in \mathbb{R}^{n\times d}, s \in \mathbb{R}_+\},$$

where $f(y_i|z_i;\beta,s)$ is a normal density with mean $\beta^t z_i$ and variance sI_d. Thus, we have

$$f(y_i|z_i;\beta,s) = (2\pi)^{-\frac{d}{2}}|sI_d|^{-\frac{1}{2}}e^{-\frac{1}{2}(y_i-\beta^t z_i)^t(sI_d)^{-1}(y_i-\beta^t z_i)},$$

and so

$$\log f(y_i|z_i;\beta,s) = -\frac{d}{2}\log(2\pi) - \frac{d}{2}\log(s) - \frac{1}{2s}(y_i - \beta^t z_i)^t(y_i - \beta^t z_i).$$

Recall that we showed before that $\hat{\beta} = \frac{1}{T}(\frac{1}{T}\Sigma\Sigma^t + \gamma I_n)^{-1}\Sigma Y^t$ is an M-estimator associated to the function $\psi_1(y_i|z_i;\beta,s) = [z_i z_i^t + \gamma I_n]\beta - z_i y_i$.

We will estimate the variance of the model by maximizing the log-likelihood; thus, we are looking for the s that maximizes

$$\sum_{i=1}^{T} \log f(y_i|z_i;\beta,s) = \sum_{i=1}^{T}[-\frac{d}{2}\log(2\pi) - \frac{d}{2}\log(s) - \frac{1}{2s}(y_i-\beta^t z_i)^t(y_i-\beta^t z_i)],$$

differentiating along s we obtain

$$\begin{aligned} \sum_{i=1}^{T}[-\frac{d}{2s} + \frac{1}{2s^2}\|\beta^t z_i - y_i\|^2] = 0 \\ \frac{1}{T}\sum_{i=1}^{T}[-ds + \|\beta^t z_i - y_i\|^2] = 0. \end{aligned}$$

Then, $\hat{s} = \frac{1}{Td}\sum_{i=1}^{T}\|\hat{\beta}^t z_i - y_i\|^2$ is an M-estimator associated to the function $\psi_2(y_i|z;\beta,s) = \|\beta^t z_i - y_i\|^2 - sd$. Therefore

$$\begin{pmatrix} \frac{1}{T}(\frac{1}{T}\Sigma\Sigma^t + \gamma I_n)^{-1}\Sigma Y^t \\ (\frac{1}{Td}\sum_{i=1}^{T}\|\hat{\beta}^t z_i - y_i\|^2)I_d \end{pmatrix}$$

is an M-estimator associated to the function

$$\psi(y_i|z;\beta,s) = \begin{pmatrix} [z_i z_i^t + \gamma I_n]\beta - z_i y_i^t \\ (\|\beta^t z_i - y_i\|^2 - sd)I_d \end{pmatrix}$$

Now the expression for the GIC is

$$GIC = -2\sum_{i=1}^{T}\log f(y_i|z_i;\hat{\beta},\hat{s}) + 2tr\{R(\psi,\hat{G})^{-1}Q(\psi,\hat{G})\},$$

where

$$R(\psi,\hat{G}) = -[\frac{1}{T}\sum_{i=1}^{T}\frac{\partial}{\partial(\beta,s)}\psi(y_i|z_i;\beta,s)\Big|_{\beta=\hat{\beta},s=\hat{s}}]^t$$

and

$$Q(\psi,\hat{G}) = \frac{1}{T}\sum_{i=1}^{T}\psi(y_i|z_i;\beta,s)[\frac{\partial}{\partial(\beta,s)}\log f(y_i|z_i;\beta,s)]^t\Big|_{\beta=\hat{\beta},s=\hat{s}}.$$

We need to compute $R(\psi,\hat{G})$ and $Q(\psi,\hat{G})$. We have

$$\frac{\partial}{\partial(\beta,s)}\psi(y_i|z_i;\beta,s)\Big|_{\beta=\hat{\beta},s=\hat{s}} = \begin{pmatrix} [z_i z_i^t +_n] & 0 \\ 2[z_i z_i^t\beta - z_i y_i^t] & -d \end{pmatrix},$$

and so

$$R(\psi,\hat{G}) = -[\frac{1}{T}\sum_{i=1}^{T}\frac{\partial}{\partial\beta}\psi(y_i|z_i;\beta,s)\Big|_{\beta=\hat{\beta},s=\hat{s}}]^t = -[\frac{1}{T}\sum_{i=1}^{T}\begin{pmatrix}[z_iz_i^t+\gamma I_n] & 0_{(n\times d)}\\ 2[z_iz_i^t\hat{\beta}-z_iy_i^t]^t & -dI_d\end{pmatrix}]. \tag{4.3}$$

Where $0_{(n\times d)}$ denotes the $n \times d$ matrix with all its entries equal to 0. Let us note that

$$\frac{\partial}{\partial(\beta,s)}\log f(y_i|z_i;\beta,s) = \begin{pmatrix}-\frac{1}{s}(z_iz_i^t\beta - z_iy_i^t)\\ (-\frac{d}{2s}+\frac{1}{2s^2}\|\beta^t z_i - y_i\|^2)I_d\end{pmatrix},$$

then we have, for Q:

$$Q(\psi,\hat{G}) = \frac{1}{T}\sum_{i=1}^{T}\begin{pmatrix}[z_iz_i^t+\gamma I_n]\hat{\beta}-z_iy_i^t\\ (\|\hat{\beta}^t z_i - y_i\|^2 - \hat{s}d)I_d\end{pmatrix}\Big(-\tfrac{1}{\hat{s}}(z_iz_i^t\hat{\beta}-z_iy_i^t)^t,\quad (-\tfrac{d}{2\hat{s}}+\tfrac{1}{2\hat{s}^2}\|\hat{\beta}^t z_i - y_i\|^2)I_d\Big) \tag{4.4}$$

Now, substituting (4.3) and (4.4) into the expression for the GIC, we have

$$GIC = -2\sum_{i=1}^{T}[-\frac{d}{2}\log(2\pi) - \frac{d}{2}\log(\hat{s}) - \frac{1}{2\hat{s}}(y_i - \hat{\beta}^t z_i)^t(y_i - \hat{\beta}^t z_i)] + 2tr\{R(\psi,\hat{G})^{-1}Q(\psi,\hat{G})\}$$

We can leave out the term $-\frac{d}{2}\log(2\pi)$, because it does not depend on γ. Then, we can write

$$GIC = \sum_{i=1}^{T}[d\log(\hat{s}) + \frac{1}{\hat{s}}(y_i - \hat{\beta}^t z_i)^t(y_i - \hat{\beta}^t z_i)] + 2tr\{R(\psi,\hat{G})^{-1}Q(\psi,\hat{G})\}$$

Now, we choose γ by minimizing the GIC.

4.2.3 GIC Practical Outcomes

In this section, we present the results of a simulation study similar to that in [1]. We have a classification task with the popular MNIST image dataset, composed of grayscale handwritten digits of size 28 × 28. We use $n = 512$ neurons and W, the weights matrix, with standard Gaussian entries. The aim is to classify nines and sevens. For this application, each image is a $p = 784$ dimensional vector, and we have $T = 1024$ training images and $T = 1024$ testing images. The output Y and $\hat{Y}$ are vectors ($d = 1$) such that $Y_{1j}, \hat{Y}_{1j} \in \{-1, 1\}$, depending on the image class.

Figure 4.1 presents the generated simulation performance of the network. The selection of the hyperparameter can be achieved using the GIC (with known variance) developed in this chapter.

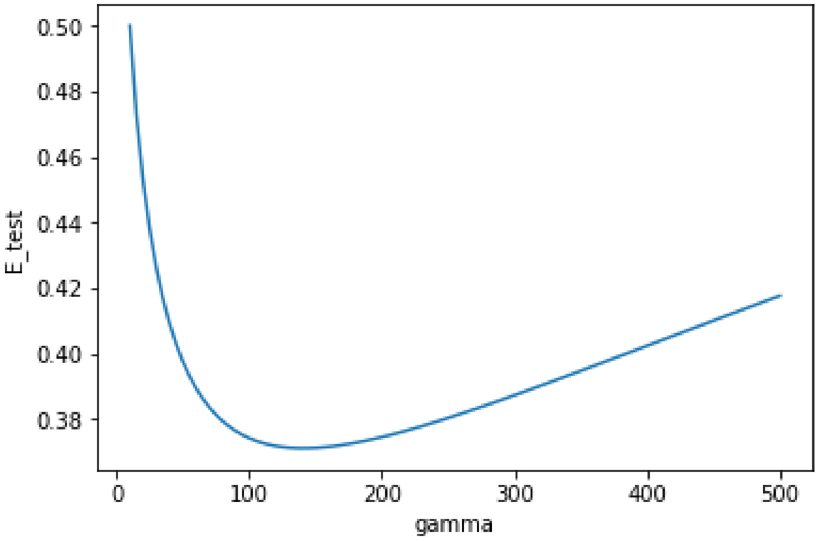


Figure 4.1: Plot of test error for different values of γ

We compute the GIC(γ) for this network using a Python script. The case of known variance was chosen for simplicity. In the following figure we can see the graph for this criterion. Note that the γ that minimizes the GIC is between 300 and 400.

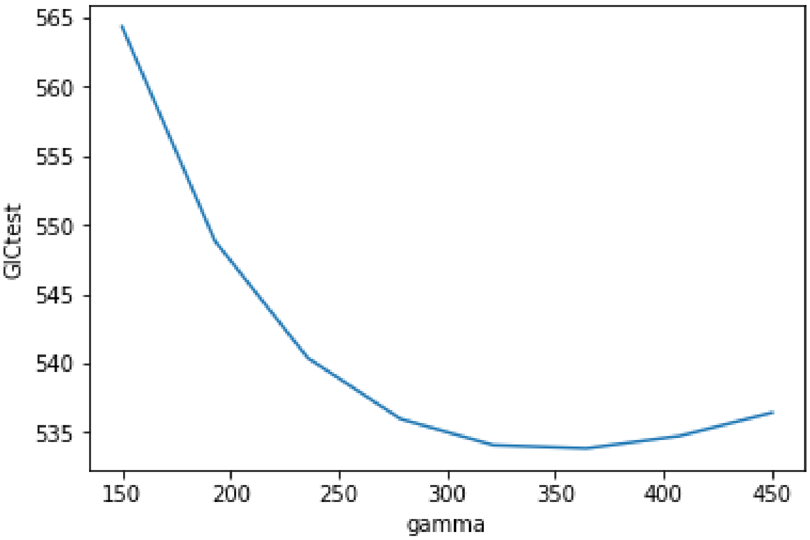


Figure 4.2: GIC for testing dataset for different values of γ

We can use an optimization library in Python to get the selected γ. As we mentioned in Chapter 3, the estimations of E_{test} and E_{train} could help in this selection. However, GIC is a statistical tool developed with the aim of selecting γ.

4.3 Generalized Cross-Validation (GCV)

In this part of the chapter we will study another model selection approach for selecting the hyperparameter γ. This approach is the Generalized Cross-Validation (GCV). This method was presented by Gene, Wahba and Heath [5] in 1979. We first state the basic definitions and then study the properties of GCV in our present model.

Definition 12. *Consider the linear model*

$$Y = \Sigma^t \beta + \varepsilon,$$

with $\varepsilon \sim N(0, sI)$ *and* $Y, \varepsilon \in \mathbb{R}^T$*,* $\beta \in \mathbb{R}^{n\times d}, \Sigma \in \mathbb{R}^{n\times T}$*. We define the Generalized Cross-Validation as*

$$GCV(\gamma) = \frac{\frac{1}{T}\|(I - A(\gamma))Y\|^2}{[\frac{1}{T}Tr(I - A(\gamma))]^2},$$

where

$$A(\gamma) = \Sigma^t(\Sigma\Sigma^t + T\gamma I)^{-1}\Sigma.$$

Recall that we have a deterministic data matrix $X = [x_1, ..., x_T] \in \mathbb{R}^{p\times T}$, a random weights matrix $W \in \mathbb{R}^{n\times p}$, and a Lipschitz activation function σ. For simplicity, we will write the ridge regression task in terms of $z_i = \sigma(Wx_i)$, for each $i = 1, 2, ..., T$. Therefore, we consider the statistical model

$$\{f(y|z;\beta, s) : \beta \in \mathbb{R}^{n\times d}, s \in \mathbb{R}_+\},$$

where $f(y_i|z_i;\beta, s)$ is a normal density with mean $\beta^t z_i$ and variance sI_d. Thus, we have

$$f(y_i|z_i;\beta, s) = (2\pi)^{-\frac{d}{2}}|sI_d|^{-\frac{1}{2}}e^{-\frac{1}{2}(y_i-\beta^t z_i)^t(sI_d)^{-1}(y_i-\beta^t z_i)},$$

and our estimator can be seen to be

$$\hat{\beta}(\gamma) = (\Sigma\Sigma^t + T\gamma I)^{-1}\Sigma Y.$$

This approach is compatible with Definition 12. Let $T(\gamma)$ be the mean square error in the $\Sigma^t\beta$ estimation,

$$T(\gamma) = \frac{1}{T}\|\Sigma^t\beta - \Sigma^t\hat{\beta}(\gamma)\|^2.$$

With a simple computation, we have that

$$\mathbb{E}[T(\gamma)] = \frac{1}{T}\|(I - A(\gamma))\Sigma^t\beta\|^2 + \frac{s}{T}Tr(A^2(\gamma)).$$

Note that an unbiased estimator $\hat{T}(\gamma)$ of $\mathbb{E}[T(\gamma)]$, for $n < T$, is given by

$$\hat{T}(\gamma) = \frac{1}{T}\|(I - A(\gamma))Y\|^2 - \frac{2\hat{s}}{T}Tr(I - A(\gamma)) + \hat{s}$$

with $\hat{s} = \frac{1}{T-n}\|(I - \Sigma^t(\Sigma\Sigma^t)^{-1}\Sigma)Y\|^2$. In our case, the restriction $n < T$ means that we have fewer neurons than training (or testing) data.

Minimizing Mallow's criterion is a way to select the hyperparameter γ; this could be seen as minimizing $T\frac{\hat{T}(\gamma)}{\hat{s}}$. In [5], an estimate arrived at by minimizing $\hat{T}$ is called an RR (range risk) estimate.

We will see that the GCV estimate is, for a huge number of training data T, an estimate for γ which approximately minimizes $\mathbb{E}[T(\gamma)]$, without an estimator for s. Note that because there is no need to estimate s, we can use GCV on problems where $T - n$ is small, a very common case in the field of neural networks, i.e., with the size of the training dataset close to the number of neurons. Another case when this property leads us to use GCV is where the real model could be

$$y_i = \sum_{j=1}^{\infty} \sigma_{ij}\beta_j + \varepsilon_i$$

for $i = 1, 2, ..., T$. This is related to the case where we have a huge number of neurons.

4.3.1 Similarities with Ordinary Cross-Validation

There is another method for estimating γ from the data without either knowledge of s or even estimators for s, the Ordinary Cross-Validation (OCV) proposed by Allen. In [5] it is explained why GCV can be expected to be generally better that OCV. There it is also stated that there are arguments for thinking that any good estimator of γ should be invariant under rotations of the coordinate system, when β and ε have spherical normal distributions (the distribution has circular symmetry: diagonal covariance matrix with equal variances). They showed, using the singular value decomposition of Σ^t, that GCV is a rotation-invariant form of the OCV.

We will see that GCV can be seen as a weighted version of OCV. The OCV works as follows. Let $\hat{\beta}^{(k)}(\gamma)$ be the estimator of β removing the kth data point y_k. The idea of OGC is that if γ is a good slelection, then the kth row of $X\beta^{(k)}(\gamma)$ should be close to y_k. The OCV method selects γ by minimizing

$$OCV(\gamma) = \frac{1}{T}\sum_{k=1}^{T} \|[X\beta^{(k)}(\gamma)]_k - y_k\|^2.$$

Using the Sherman–Morrison–Woodbury formula, we can write

$$OCV(\gamma) = \frac{1}{T}\|B(\gamma)(I - A(\gamma))Y\|^2$$

with $B(\gamma) = diag(\frac{1}{(1-a_{ii}(\gamma))})$, and $a_{ii}(\gamma)$ the ith entry of the diagonal of $A(\gamma)$.

Then, using this formula we can write

$$GCV(\gamma) = \frac{1}{T}\|B(\gamma)(I - A(\gamma))Y\|^2 w_k^{(\gamma)},$$

where

$$w_k^{(\gamma)} = \frac{1 - a_{kk}(\gamma)}{1 - \frac{1}{T}Tr(A(\gamma))}.$$

4.3.2 Some Properties of the GCV

In this part we will present some properties of the GCV and discuss the implications for our neural network model. These properties are presented in [5] and we refer the reader to that article to see the proofs.

Theorem 4.3.1. ***The GCV theorem***
Let $\mu_1 = \frac{1}{T}Tr(A(\gamma))$, $\mu_2 = \frac{1}{T}Tr(A^2(\gamma))$, *and*

$$b^2 = \frac{1}{T}\|(I - A(\gamma))\Sigma^t\beta\|^2$$

. Then,

$$\frac{\mathbb{E}[T(\gamma)] - \mathbb{E}[GCV(\gamma)] + s}{\mathbb{E}[T(\gamma)]} = \frac{-\mu_1(2-\mu_1)}{(1-\mu_1)^2} + \frac{s}{b^2 + s\mu_2}\frac{\mu_1^2}{(1-\mu_1)^2},$$

and therefore

$$\frac{\left|\mathbb{E}[T(\gamma)] - \mathbb{E}[GCV(\gamma)] + s\right|}{\mathbb{E}[T(\gamma)]} < \left(2\mu_1 + \frac{\mu_1^2}{\mu_2}\right)\frac{1}{(1-\mu_1)^2},$$

when $0 < \mu_1 < 1$.

Note that from this theorem it follows that if

$$\lim_{T\to\infty} \mu_1 = 0$$

and

$$\lim_{T\to\infty} \frac{\mu_1^2}{\mu_2} = 0,$$

then the difference $/|\mathbb{E}[T(\gamma)] - \mathbb{E}[GCV(\gamma)] + s\|$ is small compared to $\mathbb{E}[T(\gamma)]$. This fact suggests that a γ selected by minimizing the GCV is preferable to one obtained by minimizing the OCV if one aims to choose γ to minimize

$$\frac{1}{T}\mathbb{E}_{Y^*}[\|Y^* - \Sigma\beta(\gamma)\|^2],$$

where Y^* is future data, in our case this could be seen as a testing data, and $\mathbb{E}_{Y^*}$ is the expectation with respect the distribution of Y^*. Then, we can think that this result helps to achieve a good testing performance by selecting γ using the GCV.

The following result is a corollary of the previous theorem.

Corollary 4.3.1.1. *Let*

$$h(\gamma) = \Big(2\mu_1 + \frac{\mu_1^2}{\mu_2}\Big)\frac{1}{(1-\mu_2)^2},$$

and γ_0 be the minimizer of $\mathbb{E}[T(\gamma)]$. Then $\mathbb{E}[GCV[\gamma]]$ always has a minimum $\tilde{\gamma}$ so that

$$I^0 = \frac{\mathbb{E}[T(\tilde{\gamma})]}{\mathbb{E}[T(\gamma^0)]}$$

(called the expectation inefficiency) satisfies

$$I^0 \leq \frac{1 + h(\gamma^0)}{1 - h(\tilde{\gamma})}.$$

We aim for an expectation inefficiency close to 1. Note that if $h(\gamma^0)$ and $h(\tilde{\gamma})$ are small, then the mean square error at $argmin_\gamma \mathbb{E}[GCV(\gamma)]$ is not much bigger than the minimum mean square error $min_\gamma \mathbb{E}[T(\gamma)]$.

The following theorem considers the case when $d = 1$, i.e., when β is a vector instead of a matrix. Let us suppose $\beta \sim N(0, aI)$ and denote by $\mathbb{E}_\beta$ the expectation with respect to this distribution.

Theorem 4.3.2. *$\mathbb{E}_\beta[GCV(\gamma)]$ has the same minimizer of $\mathbb{E}_\beta[T(\gamma)]$ and it is $\hat{\gamma} = \frac{s}{Ta}$.*

This is an interesting theorem proved in [5]. We will not use this theorem in our applications because it requires knowledge of s.

4.4 GCV for Extreme Learning Machines

4.4.1 GCV Practical Outcomes

In this subsection, we present the results of the GCV using the same conditions as for the GIC in Section 4.2. We have a classification task with the MNIST dataset, composed of grayscale handwritten digits of size 28×28. We consider $n = 512$ neurons and W, the weights matrix, with standard Gaussian entries. As before, we aim to classify nines and sevens. We know each image is a $p = 784$ dimensional vector, and we have $T = 1024$ training images and $T = 1024$ testing images. The output Y and $\hat{Y}$ are vectors ($d = 1$) such that $Y_{1j}, \hat{Y}_{1j} \in \{-1, 1\}$ depending on the image class.

Figure 4.3 presents the generated simulation performance of the network. The following one is the training error for different values of γ.

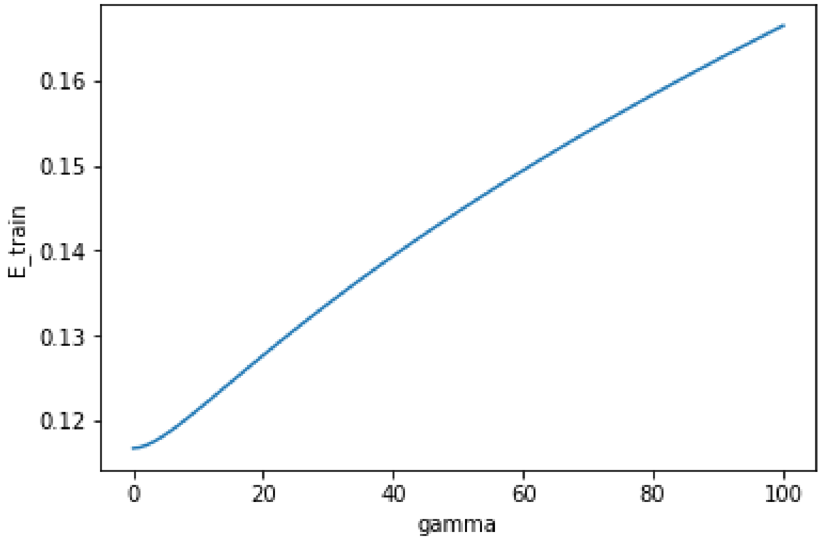


Figure 4.3: Training error for different values of γ

In Figure 4.4 we have the testing error for different values of γ. Note that a good γ could be selected between 80 and 150. Nevertheless, the selection of the hyperparameter can be achieved using the GCV method.

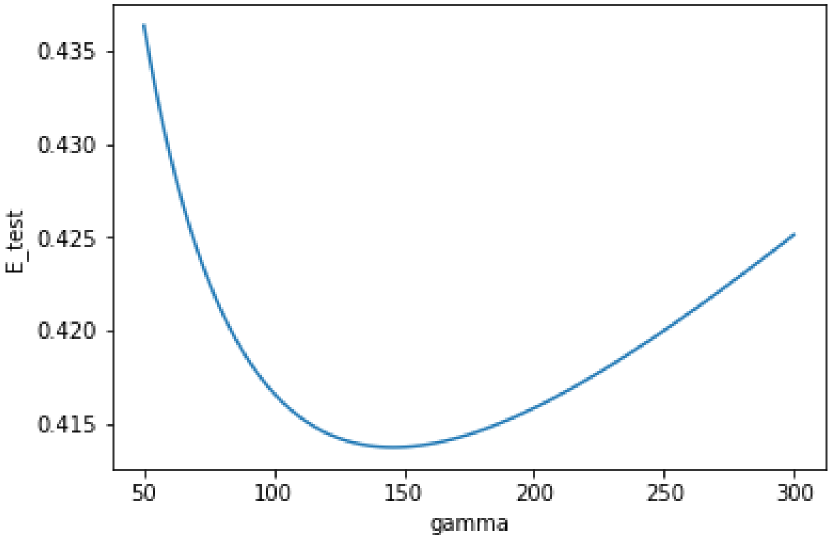


Figure 4.4: Testing error for different values of γ

We compute GCV(γ) for this network using a Python script. In Figure 4.5 we can see the graph of this criterion for the training set. Note that the γ that minimizes the GIC In English, acronyms are not italicised. In fact, in English, even a multiletter mathematical variable is not italicised if it is based on English words, such as 'log' and 'GIC'. is between 100 and 140.

In Figure 4.6 we can see the GCV for the testing dataset. Let us note that the optimal γ is almost the same as for the training dataset. This could be seen as a property that follows from Theorem 4.3.1.

Using the optimizer function *scipy.optimize.minimize* in *Python*, we got:

- The minimum in the training dataset is 120.0

- The minimum in the testing dataset is 121.0

For the following computations, we will use $\hat{\gamma} = 120.0$: in order to study the implications of Theorem 4.3.1 and Corollary 4.3.1.1, we have to compute μ_1 and μ_2 for the training datasets. We obtain the following results.

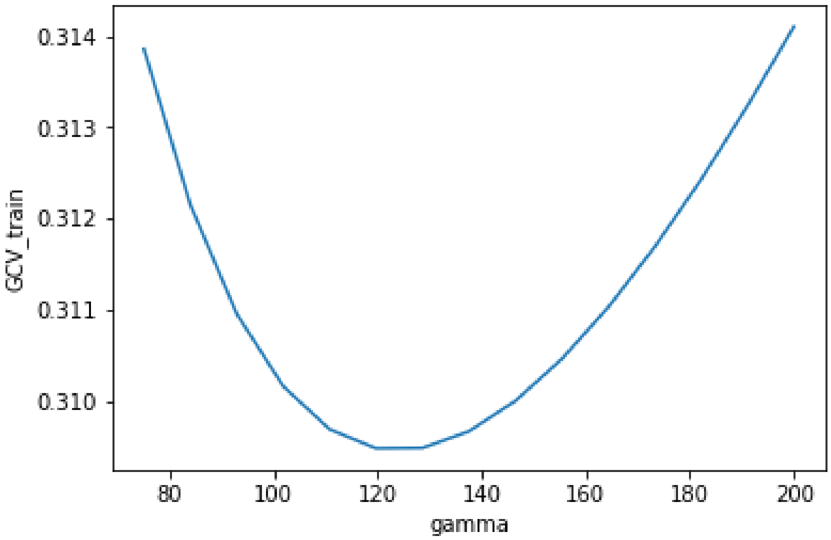


Figure 4.5: GCV for training dataset for different values of γ

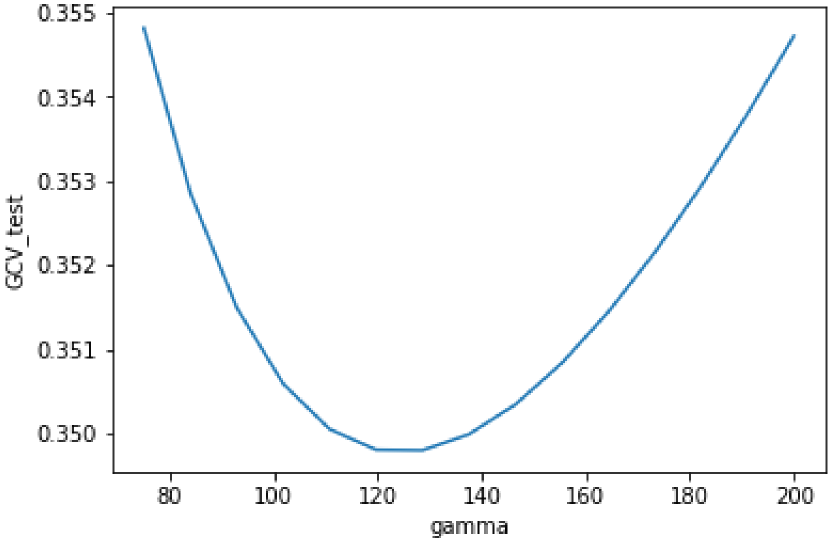


Figure 4.6: GCV for testing dataset for different values of γ

$$\mu_1(\hat{\gamma}) = 0.2505$$

$\mu_2(\hat{\gamma}) = 0.1561$, then we have

$$\frac{\mu_1^2}{\mu_2} = 0.4019.$$

Then, from Theorem 4.3.1, we can say that the difference $\mathbb{E}[T(\hat{\gamma})] - \mathbb{E}[GCV(\hat{\gamma})] + s$ is not too big in relation to $\mathbb{E}[T(\hat{\gamma})]$, that is because $0 < \mu_1 < 1$, then,

$$\frac{\left|\mathbb{E}[T(\hat{\gamma})] - \mathbb{E}[GCV(\hat{\gamma})] + s\right|}{\mathbb{E}[T(\hat{\gamma})]} < \left(2(0.2505) + \frac{0.2505^2}{0.1561}\right)\frac{1}{(1-0.2505)^2} = 1.6.$$

In conclusion, the expression for the GCV is immediately computable, following Definition 12. We saw that there is no need to estimate the variance s to use the GCV. This is an advantage of the GCV over other criteria, such as Mallow's criterion or the GIC in this chapter. Moreover, using the GCV allows us to study models without the restriction $n < T$, an important property in extreme learning machines.

Theorem 4.3.1 could imply that the GCV selection of γ is good enough to achieve a good testing performance. This is supported by Figure 4.6 and Figure 4.5. Finally, we can say that using the GCV method is a good, quick, and totally statistical way to select the hyperparameter γ.

CHAPTER 5

Training Speed in Multilayer Neural Networks

To present the main results of this chapter and its implications for the speed of the training of multilayer neural networks, we first have to talk about the basic results that motivated them. The first section is about the theoretical results on the data covariance matrix in a single layer neural network [2]. Then, we will discuss some results of the data covariance matrix for a mutilayer neural network [3]. Moreover, we present a suitable modification of a result of [3], as well as its implications for the training speed.

5.1 Data Covariance Matrix of a Single Layer Neural Network

This section presents the results from [2], where the method of moments is the main tool to achieve them. Here, a similar model of extreme learning machines will be used, as will the notation used in the previous sections. We have a single layer random neural network with n neurons. However, in this case, we will consider a random data matrix. Then, for this network we have:

- $X \in \mathbb{R}^{p \times T}$, a random data matrix with independent and identically distributed entries $N(0, \sigma_x^2)$.
- $Y \in \mathbb{R}^{d \times T}$, a random output data matrix.

Similar to the first chapter, to obtain the results, we need to make the following assumptions:

1. W Gaussian: Let $W \in \mathbb{R}^{n \times T}$ be a random weight matrix with independent and identically distributed entries $N(0, \sigma_w^2/p)$.
2. The function σ: The activation function $\sigma : \mathbb{R} \to \mathbb{R}$ has zero Gaussian mean and finite Gaussian moments, i.e.,
$$\int \frac{1}{\sqrt{2\pi}} e^{\frac{z^2}{2}} \sigma(\sigma_w \sigma_x z) dz = 0$$
and, for $k > 1$,
$$\left| \int \frac{1}{\sqrt{2\pi}} e^{\frac{z^2}{2}} (\sigma(\sigma_w \sigma_x z))^k dz \right| < \infty.$$
3. The growth rate: There are fixed constants ψ, ϕ such that
$$\lim_{p,T \to \infty} \frac{p}{T} = \phi,$$
and
$$\lim_{p,n \to \infty} \frac{p}{n} = \psi.$$

As previously stated, we have $\Sigma = \sigma(WX) \in \mathbb{R}^{n \times T}$. We are interested in the Gram matrix
$$M = \frac{1}{T} \Sigma \Sigma^t \in \mathbb{R}^{n \times n}.$$

We now recall the following definitions, presented in Chapter 2, but now using these notations:

Definition 13. *Let $\lambda_j(M)$, for $j = 1, ..., n$, be the n eigenvalues of M including multiplicity. The empirical spectral density of the matrix M is*

$$\rho_M(t) = \frac{1}{n}\sum_{j=1}^{n}\delta(t - \lambda_j(M)),$$

where δ is the Dirac delta function. If the limit exists, the limiting spectral density is defined as

$$\lim_{n\to\infty}\rho_M(t).$$

Definition 14. *For $z \in \mathbb{C}\backslash supp(\rho_M)$ we define G, the Stieltjes transform of ρ_M, as*

$$G(z) = \int \frac{\rho_M(t)}{z-t}dt = -\frac{1}{n}\mathbb{E}_{W,X}[tr(M - zI_n)^{-1}].$$

Let us note that here, the expresion $(M - zI_n)^{-1}$ is the resolvent of M. The following formula, called the inversion formula, allows us to recover the spectral density from its Stieltjes transform:

$$\rho_M(\lambda) = \frac{1}{\pi}\lim_{\varepsilon\to 0^+}\Im(G(\lambda + i\varepsilon)).$$

5.1.1 The Main Result

The following theorem provides a way to obtain G as the solution to a polynomial equation of the fourth degree.

Theorem 5.1.1. *Assume that ζ and η are as follows:*

$$\eta = \int \frac{1}{\sqrt{2\pi}}e^{\frac{z^2}{2}}(\sigma(\sigma_w\sigma_x z))^2 dz,$$

and

$$\zeta = \Big[\sigma_w\sigma_x\int \frac{1}{\sqrt{2\pi}}e^{\frac{z^2}{2}}\sigma'(\sigma_w\sigma_x z)dz\Big].$$

The Stieltjes transform of the empirical spectral density of M satisfies

$$G(z) = \frac{\psi}{z}P(\frac{1}{z\psi}) + \frac{1-\psi}{z},$$

where,

$$P = 1 + (\eta - \zeta)tP_\phi P_\psi + \frac{P_\phi P_\psi t\zeta}{1 - P_\phi P_\psi t\zeta},$$

and

$$P_\phi = 1 + (P-1)\phi, P_\psi = 1 + (P-1)\psi.$$

Idea of the proof:

The main idea of the proof is to use the method of moments for random matrices to compute the limiting spectral distribution. Thus, the moments of ρ_M are of interest. We can establish an asymptotic expansion of $G(z)$ for large z. We have the Laurent series,

$$G(z) = \sum_{k=0}^{\infty} \frac{m_k}{z^{k+1}},$$

where m_k is the kth moment of ρ_M,

$$m_k = \int t^k \rho_M(t)dt = \frac{1}{n}\mathbb{E}[trM^k].$$

If we have computed the m_k, then we can use the Laurent series and the inversion formula to obtain the density ρ_M. The m_k will be computed by expanding in powers of M inside the trace,

$$\frac{1}{n}\mathbb{E}[trM^k] = \frac{1}{nT^k}\mathbb{E}\Big[\sum_{i_1,\ldots,i_k \in [n_1], \mu_1,\ldots,\mu_k \in [T]} \Sigma_{i_1\mu_1}\Sigma_{i_2\mu_1}\Sigma_{i_2\mu_2}\Sigma_{i_3\mu_2}\ldots\Sigma_{i_k\mu_k}\Sigma_{i_1\mu_k}\Big].$$

Then we evaluate the leading contributions to the sum as the dimension of the matrix goes to infinity. We translate this problem into two subproblems. First, enumerating certain connected outer-planar graphs, and then evaluating integrals that correspond to cycles in those graphs.

Interesting Limiting Cases

$\eta = \zeta$

We can use a Hermite polynomial expansion of σ to show that $\eta = \zeta$ if and only if σ is a linear function. We refer the reader to [2] for more details. In this case, a similar result as that in Dupic and Castillo 2014 is obtained.

$\zeta = 0$

This assumption, along with $\eta = 1$, notably simplifies the expressions in Theorem 5.1.1. Thus, we have

$$zG^2 + ((1 - \frac{\psi}{\phi})z - 1)G + \frac{\psi}{\phi} = 0,$$

which is satisfied by the Stieltjes transform of the Marchenko–Pastur distribution with shape $\frac{\phi}{\psi}$. Note that when $\psi = 1$, we get the limiting spectral distribution of XX^t, which implies that $\Sigma\Sigma^t$ and XX^t have the same limiting spectral distribution. This is a really interesting result that can be seen as an *isospectral* nonlinear transformation.

Note: We can take $\eta = 1$ without loss of generality. The general case can be recovered by rescaling z.

Therefore, we can say that for $\psi = 1$ and activation functions that satisfy $\zeta = 0$, the limiting spectral distribution of the data covariance matrix is unchanged after passing though a single layer of the network. A result that presents an extension of this result will be discussed later. It was conjectured in [2] that this property is satisfied by arbitrary layers of the network. In that article, we can see a simulation study that provides supporting numerical evidence.

Remark. *Application of Theorem 5.1.1 to the asymptotic performance of random feature methods.*

As an application of Theorem 5.1.1 to the performance of a single layer neural network (such as in Section 3 of this thesis), we can state the following result. This problem setup and analysis is similar to that of [1], but here we are interested in a memorization task where the netwrok is trained on random input–output pairs. Under the assumptions stated in the introduction of this section, we focus on minimizing the loss function

$$\ell(\beta) = \frac{1}{2dT}\|Y - B^t\Sigma\|_F^2 + \gamma\|\beta\|_F^2.$$

Recall that $\Sigma = \sigma(WX)$, where $X \in \mathbb{R}^{p\times T}$ is the matrix of T p-dimensional features, $Y \in \mathbb{R}^{d\times T}$ is the matrix of T d-dimensional targets, and $W \in \mathbb{R}^{n\times p}$ is the matrix of random weights. We aim to learn (tune/estimate) the matrix β. The solution of the mimization problem *is*

$$\hat{\beta} = \frac{1}{T}\Sigma(\frac{1}{T}\Sigma^t\Sigma + \gamma I_T)^{-1}Y^t.$$

Write $Q = (\frac{1}{T}\Sigma^t\Sigma + \gamma I_T)^{-1}$*, the resolvent of* $\frac{1}{T}\Sigma^t\Sigma$*. We take* X *and* Y *to be independent and with independent Gaussian entries. The expected training loss is given by*

$$\begin{aligned}E_{train} &= \mathbb{E}_{W,X,Y}[\ell(\beta)] \\ &= \mathbb{E}_{W,X,Y}[\frac{\gamma^2}{T}tr[Y^tYQ^2]] \\ &= \mathbb{E}_{W,X}[\frac{\gamma^2}{T}trQ^2] \\ &= -\frac{\gamma^2}{T}\frac{\partial}{\partial\gamma}\mathbb{E}_{W,X}[trQ].\end{aligned}$$

Recall the definition of the Stieltjes transform:

$$G(z) = -\frac{1}{n}\mathbb{E}_{W,X}[tr(M - zI_n)^{-1}].$$

From this, it is evident that the expression for E_{train} *is related to* $G(-\gamma)$*. Nevertheless, Theorem 5.1.1 was obtained for* $\Sigma\Sigma^t$*, and* Q *contains* $\Sigma^t\Sigma$*. Fortunately, these two matrices differ only by a finite number of eigenvalues equal to zero. Thus, after some calculations, we have*

$$\frac{1}{T}\mathbb{E}_{W,X}[trQ] = \frac{(1-\frac{\phi}{\psi})}{\gamma} - \frac{\phi}{\psi}G(-\gamma).$$

From the expression for the Stieltjes transform in Theorem 5.1.1 and its derivative with respect to z*, an equation for* $G'(z)$ *can be obtained by computing the resultant of the two polynomials and eliminating* $G(z)$*. Then, we can obtain an equation for* E_{train}*. See [2] and its supplementary material for more details.*

5.2 Data Covariance Matrix of a Multilayer Neural Network

In this section we present the results from [3] and we give a different structure for the sake of providing the reader a better understanding. In the last section, we presented a theorem about the Stieltjes transform of the spectral distribution of the data covariance matrix in a single layer random neural network. We assumed W and X to be random matrices with independent and identically normal distributed entries. An extension of that result is presented here, where W and X have sub-Gaussian distributions. We also present a multilayer case of Theorem 5.1.1. Hereafter, we will adopt a sightly different notation, which will allow us to talk about more than one layer in a neural network.

5.2.1 Model

Consider a random neural network model, where:

- $X \in \mathbb{R}^{n_0 \times m}$ is a random data matrix with independent and identically distributed entries with distribution ν_1.
- $W \in \mathbb{R}^{n_1 \times n_0}$ is a random matrix with independent and identically distributed entries with distribution ν_2. W is called a weight matrix.

Both distributions have zero mean and the variance is given by: for each i, j we have,

$$\mathbb{E}[X_{ij}^2] = \sigma_x^2,$$
$$\mathbb{E}[W_{ij}^2] = \sigma_w^2.$$

Let us make the following assumptions:

1. W, X Sub-Gaussian: This assumption concerns the tail of W and X: there exist constants $\vartheta_w, \vartheta_x > 0$ and $\alpha > 1$ such that for any $t > 0$ we have

$$\mathbb{P}(|W_{11} > t|) \leq e^{-\vartheta_w t^\alpha} \text{ and } \mathbb{P}(|X_{11} > t|) \leq e^{-\vartheta_x t^\alpha}.$$

2. The function f: We consider a smooth activation function $f : \mathbb{R} \to \mathbb{R}$ with zero Gaussian mean:

$$\int f(\sigma_w \sigma_x x) \frac{e^{x^2/2}}{\sqrt{2\pi}} dx = 0.$$

Additionally, we suppose there that exist positive constants C_f, c_f and $A_0 > 0$ such that for any $A \geq A_0$ and any $n \in \mathbb{N}$ we have

$$\sup_{x \in [-A,A]} |f^{(n)}(x)| \leq C_f A^{c_f n}.$$

3. The growth rate: The dimensions of both the columns and the rows of each matrix grow together: there exist positive constants ϕ and ψ such that

$$\lim_{m\to\infty} \frac{n_0}{m} = \phi, \text{and}$$
$$\lim_{m\to\infty} \frac{n_0}{n_1} = \psi.$$

Remark. *Note that Assumption 1 and the central limit theorem give us that there exists a constant $C > 0$ such that*

$$\mathbb{P}\Big(\Big|\frac{1}{\sqrt{n_0}}\sum_{k=1}^{n_0} W_{1k}X_{k1}\Big| > t\Big) \leq Ce^{-\frac{t^2}{2}}.$$

Assumption 2 guarantees that the activation function is real analytic, which is actually a strong assumption. Nevertheless, commonly used functions fall within this framework, such as the sigmoid $f(x) = (1 - e^{-x})^{-1}$ or the softplus $f(x) = \log(1 - e^{x})$ (a smooth variant of ReLU).

As before, our main object of study is the following random matrix:

$$M = \frac{1}{m}YY^* \in \mathbb{R}^{n_1 \times n_1},$$

where $Y = f(\frac{WX}{\sqrt{n_0}})$. Denote by $(\lambda_1, ..., \lambda_{n_1})$ the eigenvalues of M. As in Definition 13, we define the empirical spectral distribution of M associated to the activation function f, by

$$\mu_{n_1}^{(f)} = \frac{1}{n_1}\sum_{i=1}^{n_1} \delta_{\lambda_i}.$$

5.2.2 Main Results

Now we have the framework to state the main theorems of this chapter. The two following theorems extend results of [2] to a more general framework.

Theorem 5.2.1. *There exists a deterministic compactly supported measure μ such that*

$$\lim_{n_1\to\infty} \mu_{n_1}^{(f)} = \mu$$

weakly almost surely.

Let us define the following parameters:

Definition 15. *Let σ_w and σ_x be as we stated before. The parameters of the activation function f are:*

$$\theta_1(f) = \int f^2(\sigma_w\sigma_x x)\frac{e^{-x^2/2}}{\sqrt{2\pi}}dx, \text{ and}$$
$$\theta_2(f) = \Big(\sigma_w\sigma_x \int f'(\sigma_w\sigma_x x)\frac{e^{-x^2/2}}{\sqrt{2\pi}}dx\Big)^2.$$

The following theorem is an immediate extension of Theorem 5.1.1, where W and X have sub-Gaussian entries.

Theorem 5.2.2. *The measure μ is characterized through a self-consistent equation for its Stieltjes transform defined for $z \in \mathbb{C}\setminus\mathbb{R}$ by*

$$G(z) = \int \frac{d\mu(x)}{x-z}.$$

Now write

$$H(z) = \frac{\psi-1}{\psi} + \frac{z}{\psi}G(z),$$
$$H_\phi(z) = 1-\phi+\phi H(z), \text{ and } H_\psi(z) = 1-\psi-\psi H(z).$$

With $\theta_1(f)$, $\theta_2(f)$ being the parameters of the activation function f, we have the following fourth-order self-consistent equation,

$$H(z) = 1 - \frac{H_\phi(z)H_\psi(z)(\theta_1(f)-\theta_2(f))}{\psi z} + \frac{H_\phi(z)H_\psi(z)\theta_2(f)}{\psi z - H_\phi(z)H_\psi(z)\theta_2(f)}.$$

Up to this point, the model corresponds to passing the input data X through one layer and applying the function f. To talk about reinserting the output data Σ through a new layer (a multilayer neural network), we have to update the notation and assumptions. So, let us denote by L the number of layers in the neural network, and consider for each $p \in \{0, ..., L-1\}$ a family of independent matrices $W^{(p)} \in \mathbb{R}^{n_{p+1} \times n_p}$ where $(n_p)_p$ is a family of increasing sequences of integers.

1. W, X Sub-Gaussian: We suppose that all the matrix entries $(W_{ij}^{(p)})_{ij}$, $a \leq i \leq n_{p+1}$, $1 \leq j \leq n_p$ for each p are independent and identically distributed with zero mean and variance σ_w^2. Consider $X \in \mathbb{R}^{n_0 \times m}$ with independent and identically distributed entries with zero mean and variance σ_x^2. In a similar way to the single layer case, suppose $W^{(p)}$ and X to be sub-Gaussian, for each p.

2. The function f: As before, we consider a smooth activation function $f : \mathbb{R} \to \mathbb{R}$ with zero Gaussian mean:
$$\int f(\sigma_w \sigma_x x) \frac{e^{x^2/2}}{\sqrt{2\pi}} dx = 0.$$
Additionally, we suppose there that exist positive constants C_f, c_f and $A_0 > 0$ such that for any $A \geq A_0$ and any $n \in \mathbb{N}$ we have
$$\sup_{x \in [-A, A]} |f^{(n)}(x)| \leq C_f A^{c_f n}.$$
Note that this assumption guarantees that f is real analytic. There are commonly used activation functions that satisfy this, such as the sigmoid $f(x) = (1 + e^{-x})^{-1}$ and the softplus $f(x) = log(1 + e^x)$.

3. The growth rate: The dimensions of both the columns and the rows of each matrix grow together: there exist positive sequences $(\phi_p)_p$ and $(\psi_p)_p$ such that
$$\lim_{m \to \infty} \frac{n_0}{m} = \phi_p, \text{ and}$$
$$\lim_{m \to \infty} \frac{n_p}{n_{p+1}} = \psi_p.$$

Then, we can define the sequence of random matrices

$$Y^{(p+1)} = f\Big(\frac{\sigma_x}{\sqrt{\theta_1(f)}} \frac{W^{(p)} Y^{(p)}}{\sqrt{n_p}} \Big) \in \mathbb{R}^{n_{p+1} \times m},$$

with $Y^{(0)} = X$. The scaling is chosen to normalize the variance of the entries of $Y^{(p)}$ at every layer. This process is similar to the batch normalization presented in [15], which improves the training speed. This fact motivates the study of the training speed of multilayer neural networks by choosing activation functions that satisfy Assumption 2. We will study this later.

The object of study is the following matrix,

$$M^{(L)} = \frac{1}{m} Y^{(L)} Y^{(L)*}$$

and

$$\mu_{n_L}^{(f)} = \frac{1}{n_L} \delta_{\lambda_i^{(L)}},$$

where $(\lambda_k^{(L)})_k$ are the eigenvalues of $M^{(L)}$.

We will state two theorems. The first one is for polynomial activation functions and the second one is for analytic bounded activation functions.

Theorem 5.2.3. *Given an integer L, suppose that f is a bounded function such that Assumption 2 holds. If $\theta_2(f) = 0$, then the asymptotic empirical spectral distribution $\mu_{n_L}^{(f)}$ is given almost surely by the Marchenko–Pastur distribution of shape parameter $\frac{\phi}{\psi_0\psi_1 \ldots \psi_{L-1}}$.*

The following conjecture is inspired by the simulation experiments in Section 5.3. There, we describe the process of modifying an activation function to make it satisfy the assumptions of Conjecture 5.2.1. Additionally, we show that some activation functions that satisfy Conjecture 5.2.1 are better than those functions that satisfy Theorem 5.2.3. We can see this conjecture as an extension of Theorem 5.2.3.

Conjecture 5.2.1. *Given an integer L, suppose that g is a bounded analytic function. Let f be a function such that*

$$f(x) = g(x) - c_1 x - c_2,$$

where $c_1 = \mathbb{E}[g'(\sigma_x \sigma_w z)]$ and $c_2 = \mathbb{E}[g(\sigma_x \sigma_w z)]$ and the expectation is taken with respect to a standard Gaussian z. Then the asymptotic empirical spectral distribution $\mu_{n_L}^{(f)}$ is given almost surely by the Marchenko–Pastur distribution of shape parameter $\frac{\phi}{\psi_0\psi_1 \ldots \psi_{L-1}}$.

Remark. *Let us note $f(x) = g(x) - c_1 x - c_2$ in the conjecture is not bounded, not the case of Theorem 5.2.3.*

5.3 Application to Training Speed

In this part we will study the applications of Theorem 5.2.3 and Conjecture 5.2.1 to the training speed in multilayer neural networks. We will impose an additional assumption, that the activation function satisfies $\theta_1(f) = 1$, for the easy use of the stated model $Y^{(p+1)}$. We first will verify the effects of using activation functions that satisfy Conjecture 5.2.1.

For the applications, we suppose $\sigma_w = \sigma_x = 1$ and we ensure this by scaling the data matrix X and initializing the entries of $W^{(0)}$ to be standard normal. We summarize the features of the activation functions as follows:

1. f real analytic
2. f bounded
3. $\int f(x)\frac{e^{x^2/2}}{\sqrt{2\pi}}dx = 0.$
4. The parameters of the function are such that:

$$\theta_1(f) = \int f^2(x)\frac{e^{-x^2/2}}{\sqrt{2\pi}}dx = 1$$
$$\theta_2(f) = \Big(\int f'(x)\frac{e^{-x^2/2}}{\sqrt{2\pi}}dx\Big)^2 = 0$$

Note we can see items 3 and 4 above as:

- $\mathbb{E}[f(x)] = 0,$
- $\mathbb{E}[f^2(x)] = 1$ and
- $\mathbb{E}[f'(x)] = 0,$

where the expectation is taken with respect to a standard Gaussian. So, if we have a real analytic and bounded function, we can use the following process to make an activation function. We compute the following constants:

- $c_1 = \mathbb{E}[f'(x)]$
- $c_2 = \mathbb{E}[f(x)]$
- $c_3 = \mathbb{E}[(f(x) - c_1 x - c_2)^2]$

Therefore, the function

$$F(x) = c_3^{-1/2}[f(x) - c_1 x - c_2]$$

satisfies our assumptions to be an activation function. We use this on the following classical activation functions:

	$f(x)$	$F(x)$
sigmoid	$(1-e^{-x})^{-1}$	$[(1-e^{-x})^{-1} - 0.206621x - 0.5][\frac{1}{\sqrt{0.000686813}}]$
sinus	$\sin(x)$	$[\sin(x) - 0.606531x][\frac{1}{\sqrt{0.0644529}}]$
tanh	$\tanh x$	$[\tanh x - 0.605706x][\frac{1}{\sqrt{0.0274157}}]$

We will show some experimental results with these three activation functions later. It is important to mention that we add to the activation function in layer $p+1$ the factor $(\frac{1}{\sqrt{np}})$ to ensure we are using the model

$$Y^{(p+1)} = f\Big(\frac{\sigma_x}{\sqrt{\theta_1(f)}}\frac{W^{(p)}Y^{(p)}}{\sqrt{n_p}}\Big) \in \mathbb{R}^{n_{p+1}\times m},$$

where $\frac{\sigma_x}{\sqrt{\theta_1(f)}} = 1$, $W^{(p)}$ is the weights matrix and $Y^{(p)}$ is the output matrix of the pth layer.

5.3.1 Motivation

In [2], it is claimed that the spectral distribution of the data covariance matrix for the pth layer determines how the input signals become distorted or stretched as they spread through the layers. Moreover, they said that highly skewed distributions imply a poor training conditioning, in the sense that it becomes slower.

This agrees with [15], where it is stated that when a deep neural network is implemented, the training becomes complicated due to the fact that the inputs of each layer depend on all previous parameters (the weights in our case). They stated that this phenomenon (called internal covariate shift) slows down the training by requiring a lower learning rate in the optimization and a more careful parameter initialization.

Our approach treats this problem. By a *distribution shift*, we refer to the change of spectral distribution of the input covariance matrix of each layer. Based on [15], we can say that the distribution shift afects the training by the fact that the optimization needs to continuously adapt to the new distribution. This idea is supported by an article by Shimodaira in 2000.

The theorems of the previous section ensure that the limiting spectral distribution of the data matrix will be preserved as it propagates through the layers. This is achieved by scaling and choosing an activation function. Hereafter, we will study, by simulations, the effects of adopting this approach.

We consider the problem of classifying the Fashion MNIST dataset, which consists of images of clothes. It has $60,000$ training examples, $10,000$ testing examples, and 10 classes (T-shirt, Pants, Pullover shirt, Dress, Coat, Sandal, Shirt, Sneaker, Bag, Ankle boot). We work with 28×28 grayscale images. The output layer has ten neurons, with *softmax* (or normalized exponential function) as the activation function. This will allow us to interpret the output as a probability distribution over the predicted output classes.

Remark. *The following experiments were made using the Python Deep Learning library Keras running on top of TensorFlow. We plot the performance on training and validation sets for the classical (blue) and the modified (red) activation functions.*

5.3.2 Experiments: Fashion MNIST, F not bounded

Sigmoid

Here we use a fully connected neural network model with $L = 3$ hidden layers, with 200, 350, and 250 neurons, respectively. The neural network uses 20 epochs, a batch size of 32, and the optimizer algorithm RMSprop. The red line is for the neural network using $F(x)$ as activation function (in all layers) and the blue line is for the $f(x)$ (classical) case. We adopted the categorical crossentropy as the loss function.

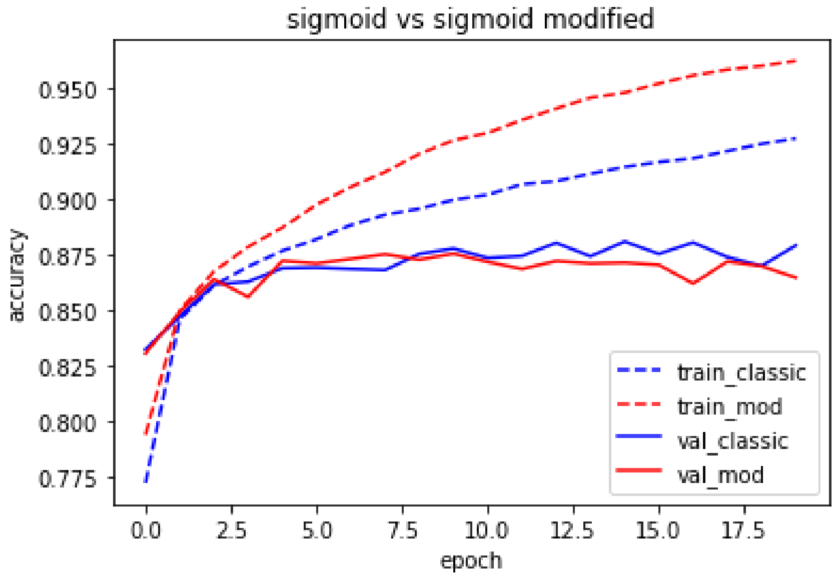


Figure 5.1: Performance of neural network with sigmoid and modified sigmoid as activation functions

Note that the accuracy is substantially improved in the $F(x)$ case for training. This could be interpreted as needing fewer training steps to achieve a specific accuracy rate. In the validation set, we get approximately the same accuracy. Nevertheless, the red line reached its maximum before the blue line. This means we are getting the same performance in fewer training steps. Note that the blue line in the validation set is above the red line for epochs greater than 7.5, which could be explained as the network is getting over-fitted faster than the classical case.

Sinus

We have a fully connected neural network model with $L = 3$ hidden layers, of 200, 300, and 250 neurons, respectively. The neural network uses 20 epochs, a batch size of 32, and RMSprop as the optimizer algorithm. The red line is for the neural network using $F(x)$ as activation function (in all layers) and the blue line is for the $f(x)$ (classical) case. We adopted the categorical crossentropy as the loss function.

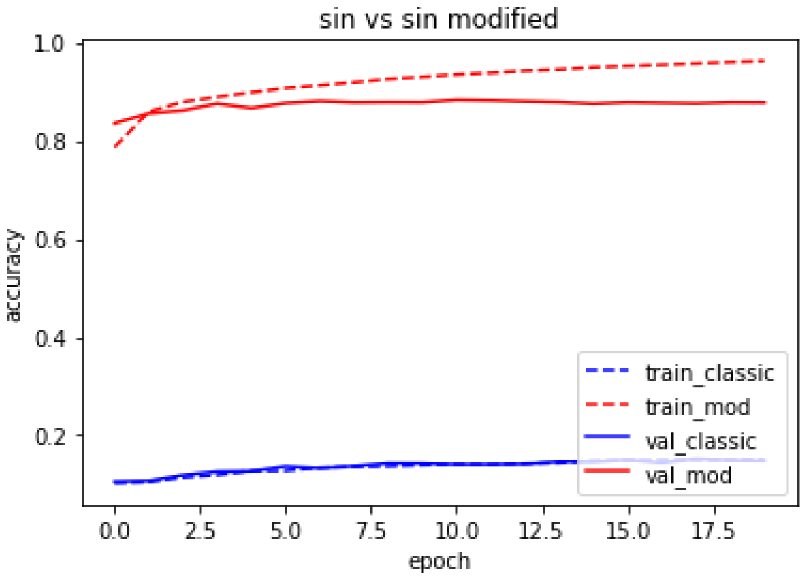


Figure 5.2: Performance of neural network with sin and modified sin as activation functions

This is an interesting result. The function $\sin(x)$ is not a good selection for an activation function in the classical case. However, the transformation $F(x)$, stated here, provides a very good performance. The modified sin beat sin in the two sets. We will use this activation function in some experiments later.

Tanh

Finally, we have a deeper fully connected neural network model with $L = 5$ layers, of 250, 350, 150, 100, and 150 neurons, respectively. The neural network uses 20 epochs, a batch size of 32, and RMSprop as the optimizer algorithm. The red line is for the neural network using $F(x)$ as activation function (in all layers) and the blue line is for the $f(x)$ (classical) case. We adopted the categorical crossentropy as the loss function.

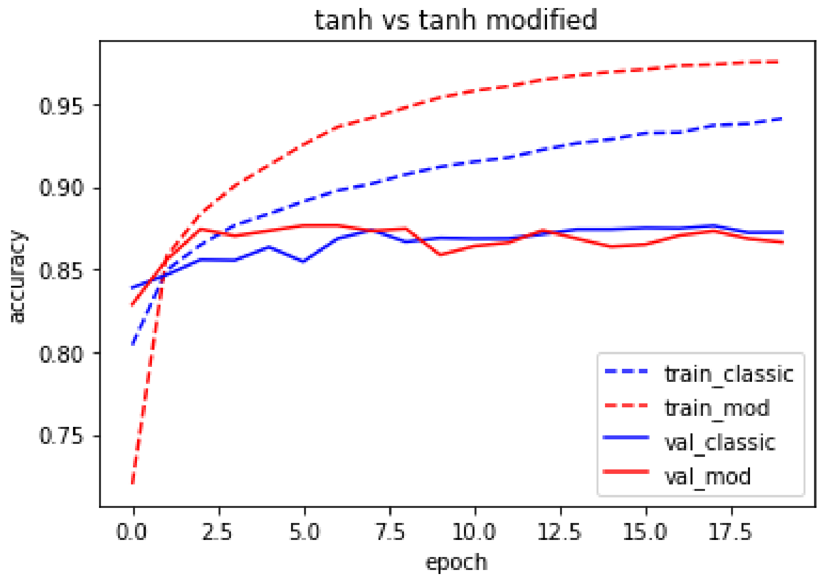


Figure 5.3: Performance of neural network with tanh and modified tanh as activation functions

In this experiment, the performance has the same behavior as in the two previous cases. The red line is better in training and it reached its maximum before the blue line.

Remark. *Let us compare the performance of our previous networks. In the following table we have the average of* 30 *simulations (runnign over* ***Google Colab****) of maximum validation accuracy of the networks as well as the epoch in which this accuracy is achieved.*

Metrics	*tanh*	*tanh mod*	*sigmoid*	*sigmoid mod*	*sin*	*sin mod*
Validation Accuracy	*0.87*	*0.87*	*0.88*	*0.87*	*0.13*	*0.88*
Epoch (argmax)	*14*	*6*	*16*	*11*	*15*	*11*
Time (seconds)	*157.18*	*170.55*	*130.60*	*138.51*	*170.95*	*180.25*

We can see that our approach achieves the maximum 1.7 *times faster than the classical case. The maximum is approximately equal for tanh and sigmoid and better for sin. Note that the sigmoid and sin cases have the same architecture, so we can compere them. The modified sin achieves the same accuracy as the sigmoid case five epochs earlier. Therefore, we have found an activation function with a very good performance:*

$$[\sin(x) - 0.606531x][\frac{1}{\sqrt{0.0644529}}]$$

5.3.3 Experiments: Fashion MNIST, F bounded

Let us consider the same classification problem as for F not bounded. Here we present the results for two bounded functions that satisfy the assumptions of Theorem 5.2.3. First, we make the following claim:

Proposition. *Every bounded real analytic function f such that its derivative f' is an odd function and*

$$\mathbb{E}[f(x)] = 0,$$
$$\mathbb{E}[f^2(x)] = 1,$$

satisfies the assumptions of Theorem 5.2.3.

Proof. Since f' is an odd function, the following integral is equal to zero:

$$\mathbb{E}[f'(x)] = \int f'(x) \frac{e^{-x^2/2}}{\sqrt{2\pi}} dx = 0.$$

This completes the proof. □

Cos

Let us consider the following function, which satisfies the hypotheses of the previous proposition:

$$\frac{1}{\sqrt{0.199788}}(\cos(x) - \frac{1}{\sqrt{e}}).$$

We choose this function because $\frac{d\cos(x)}{dx} = \sin(x)$ is an odd function. So, with a simple correction, such as that in the beginning of this section, we can get this activation function. In the following figure, we present the performance of a fully connected neural network model with $L = 2$ layers, of 250 and 350 neurons, respectively. The neural network uses 40 epochs, a batch size of 32, and RMSprop as the optimizer algorithm. The red line is for the neural network using $F(x) = \frac{1}{\sqrt{0.199788}}(\cos(x) - \frac{1}{\sqrt{e}})$ as activation function (in all layers) and the blue line is for the $f(x) = \cos(x)$ (classical) case. The categorical crossentropy was adopted as the loss function.

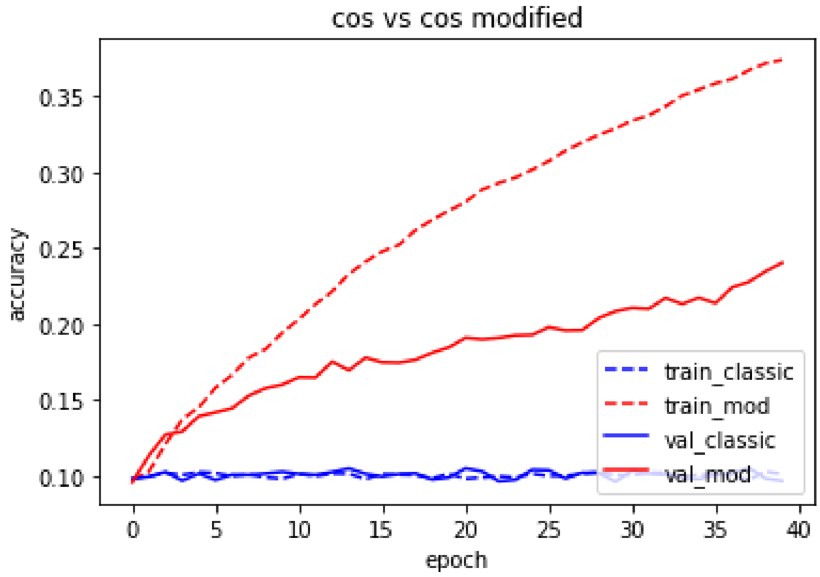


Figure 5.4: Performance of neural network with cos and modified cos as activation functions

Although this function satisfies all the requirements to be a *good* activation function (in the sense of Theorem 5.2.3), it does not have a really good training performance, likewise with the validation performance. However, the modified function is better than the classical one.

Gaussian

This is an interesting case. The red line represents a neural network using

$$F(x) = \Big(\frac{3\sqrt{5}}{3-\sqrt{5}}\Big)^{1/2}\Big(e^{-x^2} - \frac{1}{\sqrt{3}}\Big),$$

as activation functions in all its layers. Note that F satisfies the hypotheses of the previous proposition. The blue line is for $f(x) = e^{-x^2}$, a kind of Gaussian density (classical case). We use a fully connected neural network model with $L = 2$ layers, of 250 and 350 neurons, respectively. The neural network uses 25 epochs, a batch size of 32, and RMSprop as the optimizer algorithm. We adopted the categorical crossentropy as the loss function.

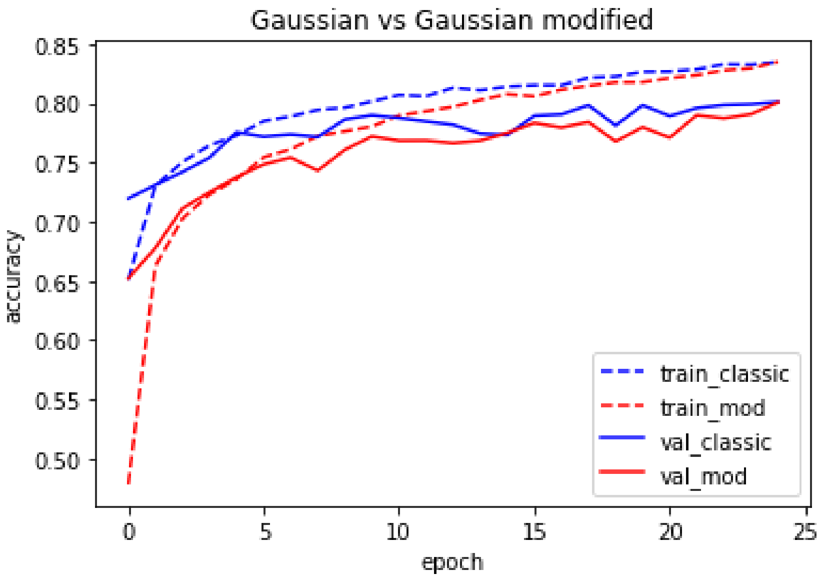


Figure 5.5: Performance of neural network with Gaussian and modified Gaussian as activation functions

In this case, we have a function that satisfies all the assumptions of Theorem 5.2.3. Nevertheless, its training performance is better in the classical case.

Remark. *These two examples provide us the insight that in some cases the assumptions of Theorem 5.2.3 are not enough to have a good performance in deep neural networks. There are many ways to try to explain this, such as by the optimizer algorithms or the loss functions, but we leave this for future research.*

Bibliography

[1] C. Louart, Z. Liao and R. Couillet (2018). A random matrix approach to neural networks. *Ann. Appl. Probab.* **28**, no. 2, 1190–1248, doi:10.1214/17-AAP1328.

[2] J. Pennington and P. Worah (2017). Nonlinear random matrix theory for deep learning. In: *Annual Advances in Neural Information Processing Systems 30: Proceedings of the 2017 Conference*, 2637–2646.

[3] L. Benigni and S. Péché (2019). Eigenvalue distribution of nonlinear models of random matrices. *ArXiv* abs/1904.03090

[4] S. Konishi and G. Kitagawa (1996). Generalized Information Criteria in Model Selection. *Biometrika* **83**, no. 4, 875–90.

[5] G. Gene, M. Heath, and G. Wahba (1979). Generalized cross-validation as a method for choosing a good ridge parameter. *Technometrics* **21**, no. 2, 215–223.

[6] J. Pennington and Y. Bahri (2017). Geometry of Neural Network Loss Surfaces via Random Matrix Theory. In: *Proceedings of the 34th International Conference on Machine Learning* **70**, 2798–2806.

[7] C. Louart and R. Couillet (2018). A Random Matrix and Concentration Inequalities framework for Neural Networks Analysis. In: *IEEE International Conference on Acoustics, Speech and Signal Processing*, 4214-4218.

[8] J. Pennington, S. Schoenholz and S. Ganguli (2017). Resurrecting the sigmoid in deep learning through dynamical isometry: Theory and practice. In: *Advances in Neural Information Processing Systems*.

[9] V. A. Marchenko and L. A. Pastur (1967). Distribution of eigenvalues for some sets of random matrices. *Math. USSR, Sb*. **1**, 457–483.

[10] N. El Karoui (2009). Concentration of measure and spectra of random matrices: Applications to correlation matrices, elliptical distributions and beyond. *Ann. Appl. Probab.* **19**, 2362–2405.

[11] J. W. Silverstein and Z. D. Bai (1995). On the empirical distribution of eigenvalues of a class of large dimensional random matrices. *Journal of Multivariate analysis*, **54**, no. 2, 175–192.

[12] R. Couillet and M. Debbah (2011). *Random Matrix Methods for Wireless Communications*. Cambridge University Press.

[13] T. Tao (2012). *Topics in Random Matrix Theory*, American Mathematical Society, Providence, RI, USA.

[14] M. A. Nielsen (2015). *Neural Networks and Deep Learning*. Determination Press, http://neuralnetworksanddeeplearning.com/index.html.

[15] S. Ioffe and C. Szegedy (2015). Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift. In: *Proceedings of the 34th International Conference on Machine Learning*.