Journal of Data Science 0 (0), 1–19
??? 2025
DOI: 10.6339/25-JDS1175
Miscellaneous

The Double Descent Behavior in Two Layer Neural Network for Binary Classification

Chathurika S. Abeykoon[1,*], Aleksandr Beknazaryan[2], and Hailin Sang[3]

[1]*Department of Mathematics, 2000 North Pkwy, Rhodes College, Memphis, TN, 38112, United States*
[2]*Department of Mathematical Sciences, University of Cincinnati, Cincinnati, OH 45221, United States*
[3]*Department of Mathematics, University of Mississippi, University, MS, 38677, United States*

Abstract

Recent studies observed a surprising concept on model test error called the double descent phenomenon where the increasing model complexity decreases the test error first and then the error increases and decreases again. To observe this, we work on a two-layer neural network model with a ReLU activation function designed for binary classification under supervised learning. Our aim is to observe and investigate the mathematical theory behind the double descent behavior of model test error for varying model sizes. We quantify the model size by the ration of number of training samples to the dimension of the model. Due to the complexity of the empirical risk minimization procedure, we use the Convex Gaussian MinMax Theorem to find a suitable candidate for the global training loss.

Keywords *generalization error; model complexity; over and under parameterization; ReLU activation; testing error*

1 Introduction

Modern machine learning models are increasingly capable of mimicking human behavior, and are now commonly used in applications such as image and speech recognition, natural language processing, game playing, self-driving cars, and bioinformatics. A key component of these advancements is the neural network, a type of algorithm inspired by the human brain's neural networks. Neural networks have significantly advanced the field of machine learning, enabling more effective decision-making and task execution.

The underlying concepts of these applications use over-parameterized models where the model has more parameters than the data points in the training set. This refers not only to the number of parameters but also to the model's capacity to memorize data, where the number of parameters is one simple measure for it. The presence of more parameters not only makes these models to be complex but also generalizes well with the previously unseen data. The best model refers to the possible lowest "test error" also known as the generalization error, which is a measure of how accurately the algorithm is able to predict outcome values for previously unseen data.

For decades, the "U-shaped curve" explained the conventional wisdom of generalization error with respect to increasing model complexity where the generalization error decreases and increases again due to the bias-variance trade-off scenario. This classical wisdom in statistical

*Corresponding author. Email: abeykoonc@rhodes.edu.

Received October 1, 2024; Accepted March 6, 2025

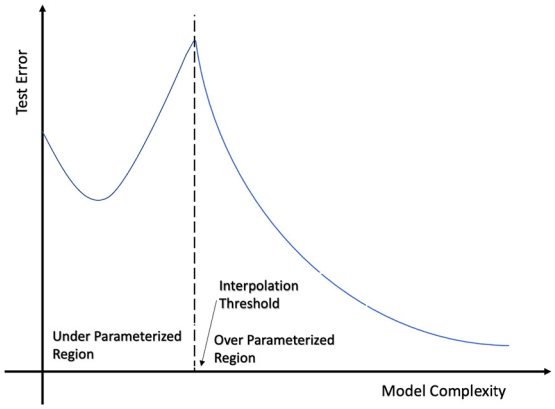

Figure 1: The double descent phenomenon in highly over-parameterized models.

learning focuses on finding the "sweet spot" or the bottom of the U-shaped curve that refers to the lowest possible testing error balancing the under-fitting and over-fitting. This well-established idea focuses on controlling the model complexity to find the best fit. The recent studies by Belkin, Hsu, Ma, and Mandal in Belkin et al. (2019), proposed a surprising behavior on generalization error in a prediction problem called the "Double-Descent" phenomenon where a second descent happens after the classical U-shaped curve for increasing model complexity.

With the double descent behavior, the test error first decreases and then increases tracing the U-shaped curve, and decreases again after the model complexity goes through a certain threshold value. The original concept in Belkin et al. (2019) analyzes the test error with respect to function class sizes or specifically by the number of parameters needed (p) and the number of training samples (n), so that around $p \approx n$, the model enters the modern interpolating regime. This transition threshold ($p \approx n$) is also known as the interpolation threshold which separates the under-parametrized region (the classical regime) and the over-parametrized region (modern interpolating regime). Belkin et al. in Belkin et al. (2019) empirically show the existence of this behavior in neural networks, decision trees and ensemble methods.

Large neural networks have the capacity to perform a second descent in over-parametrized regions due to their increased complexity. This has been demonstrated experimentally for many machine learning architectures like decision trees, two/multi-layer neural networks, and random features while some other studies like Nakkiran et al. (2021), Geiger et al. (2020) and Spigler et al. (2019) explain this phenomenon related to ResNets, CNNs, and Transformers. Hence the empirical results of these models are more successful than the theoretical findings related to double descent.

1.1 Various Forms of Double Descent

Recent literature on double descent behavior studies the sensitivity of test error to the change in different settings/ variables in the model and demonstrates the presence of double descent.

The model-wise double descent is observed when the test error is expressed as a function of varying number of parameters p while the model dimension d and number of samples n are fixed. This is the original double descent concept explained by Belkin et al. in Belkin et al. (2019). The epoch-wise double descent analyzes the test error for varying training time while keeping the model dimension d and the number of samples n fixed. Higher the number of epochs,

the longer the training time is needed. The longer we train, the better the performance. The learning rates, training procedures, noises in the model, and optimization methods may have considerable effect on the double descent curve based on epoch (Advani et al. (2020), Nakkiran (2019) and Nakkiran et al. (2021)). In sample-wise double descent, the number of observations in the training procedure increases in order to observe the double descent behavior while the model dimension and the number of parameters are fixed. This concept is studied experimentally in Nakkiran et al. (2021), with reference to effective model complexity and they observe the peak when the number of samples matches the effective model complexity (Nakkiran (2019), Nakkiran et al. (2020)).

A new approach to explain over-parameterization is to use the ratio between its training sample size and the number of parameters in the model. This ratio is often termed as the "over-parameterization ratio" where it enables us to decide the parameterization in two regions. The sensitivity of the test error for the ratios between n, p (number of parameters), and d are also observed both experimentally and theoretically in modern machine learning literature (Kini and Thrampoulidis (2020), Deng et al. (2022) and Spigler et al. (2019)).

1.2 Our Contribution

Inspired by the fascinating studies on ratio-based double descent behavior, we work on the mathematical and statistical evaluation of a simple machine learning architecture for binary classification problem. Compared to the above studies, our work is new as follows.

We study the double descent of the test error in a binary linear classification problem using the student model: a two-layer neural network equipped with a ReLU activation function. The n training data are generated from a teacher model which assigns each feature vector $\mathbf{x}_i$ a binary label y_i using two Gaussian vectors in d dimension. We would call our work the "ratio-wise double descent" and we quantify the over-parameterization by the ratio $\alpha = n/d$. We study the double descent behavior of the test error by treating it as a function of α when $n, d \to \infty$. The over-parameterization and under-parameterization regions can be easily separated as the following.

- $\alpha < 1$ implies that $n < d$. This defines the over-parameterized region of our model.
- $\alpha > 1$ implies that $d < n$. This defines the under-parameterized region of our model.
- $\alpha = 1$ implies that $n = d$. This point separates the two regions as over and under-parameterized regions.

As we increase α, the model first goes through the over-parameterized region and then passes to the under-parameterized region. We identify this as another difference between the model-wise and ratio-wise double descents because, in the model-wise double descent, the test error switches from under-parameterized to over-parameterized regions when model complexity increases.

We do not perform any training algorithm and instead, we use Convex Gaussian Min-max Theorem (CGMT) to find a theoretical candidate for the local minimum of the training risk. We observe that in the higher dimension, this candidate and global training loss have similar behaviors. With respect to a specific loss function, the final test error when $n, d \to \infty$ is a function of α and we use this to view the double descent phenomena in our model for binary classification. Theorem 1 provide the generalization error formula of the student model in terms of parameters and Theorem 2 give us values of these parameters for minimized empirical loss when $n, d \to \infty$ and they are valid for any margin-based loss function. We present the graphical results based on the square loss function and the two theorems at the end.

2 Visualizing the Double Descent in Binary Classification Models

Before working on the theoretical analysis of the double descent occurrence in a simple binary classification model, below we visualize the double descent concept in the test error of the famous *WisconsinBreastCancer* dataset. This dataset, widely used in classification tasks, contains 30 features such as tumor size and texture, which can be used to predict whether a tumor is malignant or benign. We vary the complexity of the binary classification model by increasing the sample size used to train the model while having the dimension fixed. We observe that the occurrence of a second descent is affected by factors like the regularization strength, the number of hidden nodes in the model, and the number of training epochs.

The following figures illustrate the behavior of test error as a function of α where $\alpha = n/d$ for 200 epochs in a two-layer binary classification neural network. The model applies ReLU activation function and it consists of one hidden layer with 10 neurons. The output layer has 2 neurons equipped with a sigmoid activation function for binary classification probabilities. It uses Adam optimizer for training and binary cross-entropy as the loss function. For R code, see supplementary material 2 S.1.

Note that similar double descent behaviors using real datasets CIFAR-10 and MINST have been demonstrated for different types of models in the works D'Ascoli et al. (2020) and Lee and Cherkassky (2024).

After observing the phenomenon of double descent in above binary classification model, next we aim to provide a theoretical foundation for this behavior by analyzing it within the context of a simple two-layer neural network architecture. This approach helps us understand how over-parameterization, under-fitting, and overfitting interact in neural networks, and offers insight into the mechanisms behind the observed empirical results.

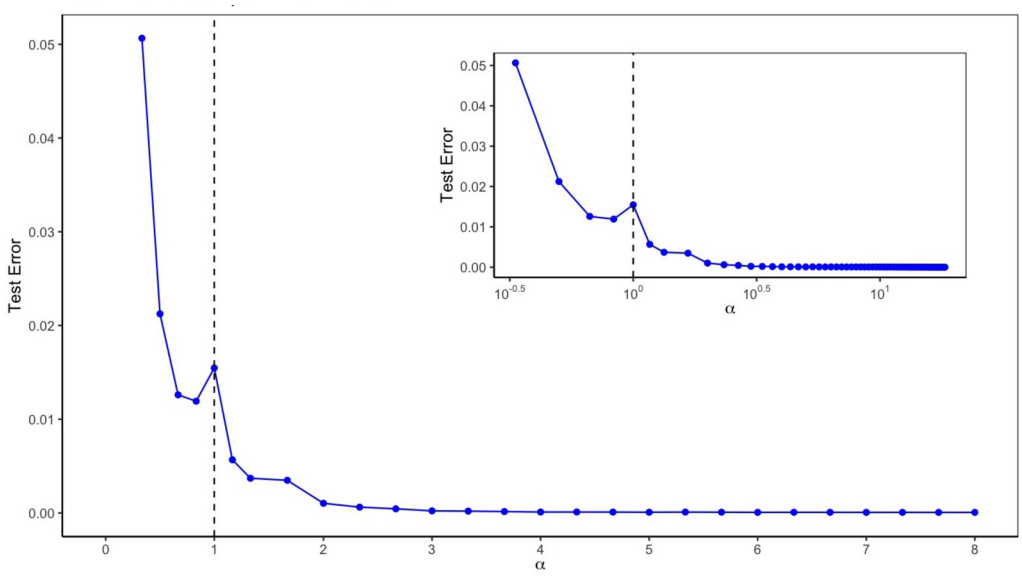

Figure 2: Test error showing the double descent behavior when a two-layer ReLu model is used with very low l_2 regularization $\lambda = 10^{-6}$ for binary classification in Wisconsin breast cancer dataset. The test error decreases first and shows a slight increase around $\alpha = 1$ and decreases again in the under-parameterized region.

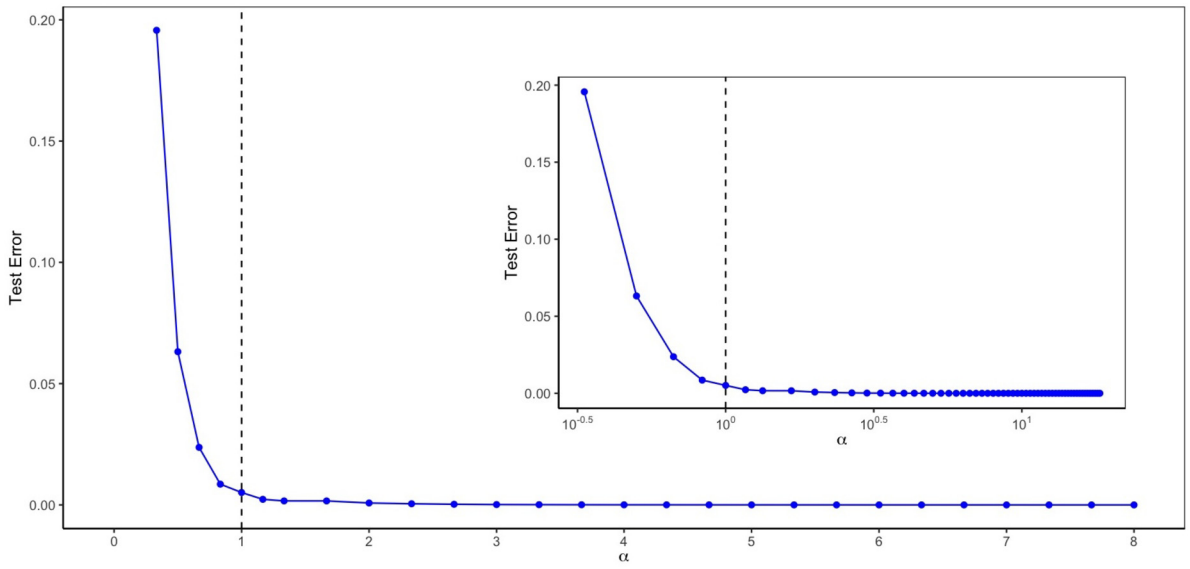

Figure 3: Test error not showing the double descent behavior when a two-layer ReLu model is used with high l_2 regularization $\lambda = 0.1$ for binary classification in Wisconsin breast cancer dataset. The test error decreases monotonically in both over and under parameterized regions.

3 Two Layer Neural Network with ReLU Activation Function

3.1 Problem Setup

We consider a two layer neural network for binary classification under supervised learning. For a given input vector $\mathbf{x} \in \mathbb{R}^d$, the single output unit consists of a label $y \in \{-1, 1\}$. The rectified linear unit or ReLU ($\sigma(z) = \max(0, z)$) is used as the activation function in the hidden layer. This function is computationally simple and efficient and it is heavily used as the default activation function in deep neural networks.

3.2 Student Model

The student model works as follows. The input layer loads the data into the neural network and it consists of d nodes. Then the first layer with a single neuron computes a function of inputs from $\mathbb{R}^d$ to $\mathbb{R}$ as $\mathbf{x}_i^T \boldsymbol{\beta}$ and sends it through the ReLU activation function. We use a bias term b to make the results more general.

$$f(\mathbf{x}_i) = \sigma\left(\frac{\mathbf{x}_i^T \boldsymbol{\beta}}{\sqrt{d}} + b\right). \tag{1}$$

Here, $\boldsymbol{\beta} \in \mathbb{R}^d$ is the weight vector, $\mathbf{x}_i \in \mathbb{R}^d$ is the ith feature vector for $i = 1, .., n$, then $\mathbf{x}_i^T \boldsymbol{\beta} = \sum_{j=1}^d x_{ij}^T \beta_j = x_{i1}\beta_1 + \cdots + x_{id}\beta_d, \in \mathbb{R}^d$. The jth feature for the ith training example is denoted as x_{ij}. The feature matrix denotes all the training cases along with their features as $\mathbf{X} \in \mathbb{R}^{d \times n}$. Training the student model is done using a dataset with n data points as $\mathcal{D} = \{(\mathbf{x}_i, y_i) | \mathbf{x}_i \in \mathbb{R}^d, y_i \in \{1, -1\}, 1 \leqslant i \leqslant n\}$.

In the second layer which is the output layer consisting of a single node classifies the input $\mathbf{x}_i$ into two labels as follows for all $i \in [n]$.

$$\hat{y}_i = \begin{cases} 1 & \text{if } f(\mathbf{x}_i) > 0 \\ -1 & \text{if } f(\mathbf{x}_i) \leqslant 0. \end{cases}$$

3.3 Data Generation – Teacher Model

We study supervised binary classification under the following data distribution for feature vector $\mathbf{x}_i \in \mathbb{R}^d$ using the class labels $y_i \in \{-1, 1\}$. We incorporate two independent Gaussian vectors as $\boldsymbol{\eta}$ and $\boldsymbol{\epsilon}_i$ in $\mathbb{R}^d$ which have components from $\mathcal{N}(0, 1)$. In particular, for each $i \in [n]$, a given data point $\mathbf{x}_i$ is generated as,

$$\mathbf{x}_i = \frac{\boldsymbol{\eta}}{\sqrt{d}} y_i + \boldsymbol{\epsilon}_i. \tag{2}$$

Each data point relates to one of the two class labels $\{-1, 1\}$ with probabilities ρ_{-1} and ρ_1 such that $\rho_{-1} + \rho_1 = 1$. We define two Gaussian clusters located at $\frac{\boldsymbol{\eta}}{\sqrt{d}}$ and $-\frac{\boldsymbol{\eta}}{\sqrt{d}}$. Since our data set consists of n data points, we expect to have $n\rho_1$ and $n\rho_{-1}$ points respectively in two clusters. Under this setting the feature vector $\mathbf{x}_i \in \mathbb{R}^d$ is a Gaussian vector with mean $\mathbf{0}$.

3.4 Loss and Risk

Our study is valid for margin based convex loss functions, such as square loss function. We evaluate the classification performance of the network f by the **empirical risk** $\hat{R}_n(\boldsymbol{\beta})$ subject to a margin based loss function $l : \mathbb{R} \longrightarrow \mathbb{R}$ in binary classification. To overcome overfitting from too large weights for unseen data, we add l_2 regularization term to the empirical risk.

$$\hat{R}_n(\boldsymbol{\beta}) = \sum_{i=1}^{n} l\left(y_i \sigma\left(\frac{\mathbf{x}_i^T \boldsymbol{\beta}}{\sqrt{d}} + b\right)\right) + \frac{\lambda}{2} ||\boldsymbol{\beta}||_2^2. \tag{3}$$

The non-negative tuning parameter λ is used to control the balance between overfitting and under-fitting. Most popular margin based loss functions are, square loss: $l(x) = \frac{1}{2}(1 - x)^2$, hinge loss: $l(x) = \max\{0, 1 - x\}$ and logistic loss: $l(x) = \log(1 + e^{-x})$.

3.5 Asymptotic Setting and Fixed Quantities

We are interested in high dimensional setting where n and $d \longrightarrow \infty$ while preserving the ratio $\alpha = n/d$ fixed. Furthermore we will observe changes that are possible to happen with large sample sizes and higher dimensions as well as the ratios in between them. To achieve an accurate analytical formula for generalization error, we choose two non-negative quantities r and s having the following interpretations. We require each β_j to be bounded for all $1 \leqslant j \leqslant d$ and define,

$$r = \frac{1}{d} ||\boldsymbol{\beta}||_2^2 \quad \text{and} \quad s = \frac{1}{d} \boldsymbol{\beta}^T \boldsymbol{\eta}. \tag{4}$$

Later on we will restrict our attention to the domain $s^2 \leqslant r$ (see Section 5).

4 Generalization

We consider a new sample data $(\mathbf{x}_N, y_N)$ and we quantify the test error (deterministic) of the model using $\hat{\boldsymbol{\beta}} = \boldsymbol{\beta}(\mathbf{X}, \mathbf{y})$ where $(\mathbf{X}, \mathbf{y})$ is our original training sample. Predicted value of the new data is decided using the classification rule $\hat{f}(\mathbf{x}_N) = sgn(\sigma(\frac{\mathbf{x}_N^T \hat{\boldsymbol{\beta}}}{\sqrt{d}} + b))$ where $sgn(z) = 1$ if $z > 0$

and $sgn(z) = -1$ if $z \leqslant 0$ for any $z \in \mathbb{R}$. Let $\boldsymbol{\epsilon}_N$ have the components taken from $\mathcal{N}(0,1)$ pairing with the teacher model introduced in (2). The generalization error is defined as the expectation of getting a misclassified output which is calculated using the indicator function.

$$R(\hat{\boldsymbol{\beta}}) = \mathbb{E}\left[\mathbb{1}(\hat{f}(\mathbf{x}_N) \neq y_N)\right].$$

Theorem 1. *The test/generalization error of the two layer neural network model defined in (1) under the fixed quantities r and s introduced in (4) is given by,*

$$R(\hat{\boldsymbol{\beta}}) = 1 - \rho_1 \Phi\left(\frac{s+b}{\sqrt{r}}\right) - \rho_{-1}\Phi\left(\frac{s-b}{\sqrt{r}}\right), \tag{5}$$

where Φ is the cumulative distribution function of standard normal distribution, b is the bias term and ρ_{-1} and ρ_1 relate to the probabilities of getting 1 or -1 in classification.

Proof. Observe that, $R(\hat{\boldsymbol{\beta}}) = \mathbb{E}\left[\mathbb{1}(\hat{f}(\mathbf{x}_N) \neq y_N)\right] = \mathbb{P}\left[\hat{f}(\mathbf{x}_N) \neq y_N\right]$. As y_N takes the values -1 and 1 with probabilities ρ_{-1} and ρ_1, respectively, we have

$$\begin{aligned}
R(\hat{\boldsymbol{\beta}}) &= \mathbb{P}\left[\operatorname{sgn}\left(\sigma\left(\frac{\mathbf{x}_N^T\hat{\boldsymbol{\beta}}}{\sqrt{d}} + b\right)\right) \neq y_N\right] \\
&= \rho_1 \mathbb{P}\left[\operatorname{sgn}\left(\sigma\left(\frac{\mathbf{x}_N^T\hat{\boldsymbol{\beta}}}{\sqrt{d}} + b\right)\right) \neq 1\right] + \rho_{-1}\mathbb{P}\left[\operatorname{sgn}\left(\sigma\left(\frac{\mathbf{x}_N^T\hat{\boldsymbol{\beta}}}{\sqrt{d}} + b\right)\right) \neq -1\right] \\
&= \rho_1 \mathbb{P}\left[\frac{\mathbf{x}_N^T\hat{\boldsymbol{\beta}}}{\sqrt{d}} + b \leqslant 0\right] + \rho_{-1}\mathbb{P}\left[\frac{\mathbf{x}_N^T\hat{\boldsymbol{\beta}}}{\sqrt{d}} + b > 0\right].
\end{aligned}$$

Next, use the teacher model (2) to simplify the last line. Also notice that $\boldsymbol{\epsilon}_N^T\hat{\boldsymbol{\beta}}$ follows Gaussian distribution with mean 0 and variance rd. Then using the definitions given in 4, we have the following.

$$\begin{aligned}
R(\hat{\boldsymbol{\beta}}) &= \rho_1 \mathbb{P}\left[\left(\frac{\boldsymbol{\eta}}{\sqrt{d}} + \boldsymbol{\epsilon}_N\right)^T \hat{\boldsymbol{\beta}} + b\sqrt{d} \leqslant 0\right] + \rho_{-1}\mathbb{P}\left[\left(\frac{-\boldsymbol{\eta}}{\sqrt{d}} + \boldsymbol{\epsilon}_N\right)^T \hat{\boldsymbol{\beta}} + b\sqrt{d} > 0\right] \\
&= \rho_1 \mathbb{P}\left[\boldsymbol{\epsilon}_N^T\hat{\boldsymbol{\beta}} \leqslant -\sqrt{d}(s+b)\right] + \rho_{-1}\mathbb{P}\left[\boldsymbol{\epsilon}_N^T\hat{\boldsymbol{\beta}} \geqslant \sqrt{d}(s-b)\right] \\
&= \rho_1 \Phi\left(\frac{-(s+b)}{\sqrt{r}}\right) + \rho_{-1}\left(1 - \Phi\left(\frac{s-b}{\sqrt{r}}\right)\right).
\end{aligned}$$

□

We see that the generalization error depends on the values of s and r along with the probabilities ρ_{-1} and ρ_1. Moreover, when the bias term $b = 0$, the generalization error depends on s and r only. In our model we assume that the ratio $\alpha = n/d$ is fixed as $n, d \to \infty$, and in Theorem 2 we find the asymptotic values r^*, s^* and b^* of r, s and b, respectively, as $n, d \to \infty$.

5 Regularized Empirical Risk

In this theoretical work, we do not follow any training algorithms as in decision trees, support vector machines, and logistic regression (Bhavsar and Ganatra (2012), Mahesh et al. (2020),

Bonaccorso (2018)) or iterative optimization procedures like gradient descent and stochastic gradient descent (Amir et al. (2021), Hutter et al. (2019)). During the procedure, we feed the training data generated from the teacher model (2) to the student model (1). We expect to have minimal empirical loss and we solve the empirical risk minimization as an optimization problem. We use Legendre transformation and the Convex Gaussian Min-max Theorem (CGMT) Thrampoulidis et al. (2014) to find a theoretical lower bound for the local training loss and avoid computing the exact local loss.

Our goal is to minimize the empirical risk in (3) to achieve an analytical formula for training loss subject to asymptotic settings in high dimension. First we define the concept of "local training loss" $L_\lambda(r, s)$ using the fixed values r, s and regularization parameter λ:

$$\begin{aligned} L_\lambda(r,s) := \min_{\boldsymbol{\beta}} \quad & \frac{1}{d}\sum_{i=1}^{n} l\left(y_i\sigma\left(\frac{\mathbf{x}_i^T\boldsymbol{\beta}}{\sqrt{d}}+b\right)\right)+\frac{\lambda}{2d}||\boldsymbol{\beta}||_2^2, \\ \text{subjectto} \quad & r=\frac{1}{d}||\boldsymbol{\beta}||_2^2, \\ & s=\frac{1}{d}\boldsymbol{\beta}^T\boldsymbol{\eta}. \end{aligned} \tag{6}$$

It is easy to see that l_2 norm of standard Gaussian random vector is approximately equal to the square root of its dimension when the dimension is large enough. Hence by Cauchy-Schwartz inequality, we have

$$|s| = \frac{|\boldsymbol{\beta}^T\boldsymbol{\eta}|}{d} \leqslant \frac{||\boldsymbol{\beta}||_2}{\sqrt{d}}\frac{||\boldsymbol{\eta}||_2}{\sqrt{d}} \approx \frac{\sqrt{rd}}{\sqrt{d}}\frac{\sqrt{d}}{\sqrt{d}} = \sqrt{r}.$$

Then $s^2 \leqslant r$ holds in high dimensional settings and we define the "global training loss" with the constraint $s^2 \leqslant r$ as below.

$$L_\lambda^* := \min_{s^2\leqslant r} \quad L_\lambda(r,s). \tag{7}$$

Our next steps include finding a deterministic function for local training loss while keeping the ratio $\alpha = n/d$ fixed.

5.1 Regularized Empirical Risk Minimization Procedure

If we omit the constraints, the minimization problem in (6) can be written as

$$\begin{aligned} L_\lambda(r,s) &= \min_{\boldsymbol{\beta}} \quad \left\{\frac{1}{d}\sum_{i=1}^{n} l\left(y_i\sigma\left(\frac{\mathbf{x}_i^T\boldsymbol{\beta}}{\sqrt{d}}+b\right)\right)+\frac{\lambda}{2d}||\boldsymbol{\beta}||_2^2\right\} \\ &= \min_{\boldsymbol{\beta}} \quad \left\{\frac{1}{d}\sum_{i=1}^{n} l\left(\frac{y_i}{2\sqrt{d}}\left(\mathbf{x}_i^T\boldsymbol{\beta}+b\sqrt{d}+|\mathbf{x}_i^T\boldsymbol{\beta}+b\sqrt{d}|\right)\right)+\frac{\lambda}{2d}||\boldsymbol{\beta}||_2^2\right\}. \end{aligned} \tag{8}$$

In the second line above we have used the idea that the ReLU function can be represented as $\sigma(z) = (z + |z|)/2$. Solving this minimization problem is complicated and the derivative of the absolute value term does not exist when it is zero. Hence we use the Convex Gaussian Min-max Theorem (CGMT) (Thrampoulidis et al. (2014), see also Thrampoulidis et al. (2015)) to handle this minimization problem. For this purpose, we should rewrite the minimization problem on $\boldsymbol{\beta}$ as a combination of a maximization problem on a new variable ($\mathbf{u} \in \mathbb{R}^n$) followed by the original minimization. This is a min-max problem since we deal with both minimization and maximization on two different variables.

To convert the original problem (8) to a min-max problem, we utilize Legendre transformation for the convex loss function $l(\cdot)$ and rewrite it as a maximization problem (see supplementary material 1 Section S.1 for more details).

Let $\tilde{l}(\cdot)$ be the Legendre transformation of the convex loss function $l(\cdot)$. Then we use Legendre transformation for each i, $1 \leqslant i \leqslant n$ and rewrite $L_\lambda(r, s)$ as the following.

$$L_\lambda(r,s) = \min_{\boldsymbol{\beta}} \ \frac{1}{d}\sum_{i=1}^{n} \max_{u_i} \ \left\{ \frac{u_i y_i}{2\sqrt{d}} \left(\mathbf{x}_i^T \boldsymbol{\beta} + b\sqrt{d} + |\mathbf{x}_i^T \boldsymbol{\beta} + b\sqrt{d}| \right) - \tilde{l}(u_i) \right\} + \frac{\lambda r}{2}.$$

The sum of the maximums with respect to each u_i is the same as the maximum of the sum with respect to $\mathbf{u}$. Together with the teacher model in (2) and constraints in (4), we have

$$L_\lambda(r,s) = \frac{\lambda r}{2} + \min_{\boldsymbol{\beta}} \ \max_{\mathbf{u}} \ \frac{1}{d}\sum_{i=1}^{n} \left\{ \frac{u_i s}{2} - \tilde{l}(u_i) + \frac{u_i y_i b}{2} + \frac{u_i y_i}{2\sqrt{d}} |\mathbf{x}_i^T \boldsymbol{\beta} + b\sqrt{d}| + \frac{u_i y_i \boldsymbol{\epsilon}_i^T \boldsymbol{\beta}}{2\sqrt{d}} \right\}.$$

Next we use a standard normal vector $\boldsymbol{\zeta}_i^T$ to substitute the standard normal vector $y_i \boldsymbol{\epsilon}_i^T$.

$$L_\lambda(r,s) = \frac{\lambda r}{2} + \min_{\boldsymbol{\beta}} \ \max_{\mathbf{u}} \ \frac{1}{d}\sum_{i=1}^{n} \left\{ \frac{u_i (s + y_i b)}{2} - \tilde{l}(u_i) + \frac{u_i y_i}{2\sqrt{d}} |\mathbf{x}_i^T \boldsymbol{\beta} + b\sqrt{d}| + \frac{u_i \boldsymbol{\zeta}_i^T \boldsymbol{\beta}}{2\sqrt{d}} \right\}. \tag{9}$$

According to CGMT, we shall call (9) as the primary optimization problem (PO) derived from the original minimization problem in (8). Next, considering the convexity of $\tilde{l}(u_i)$ and the d dimensional Gaussian random vector $\boldsymbol{\zeta}$, we define the auxiliary optimization problem (AO) which is denoted by $\tilde{L}_\lambda(r, s)$. For $\boldsymbol{\beta} \in \mathbb{R}^d$ and $\mathbf{u} \in \mathbb{R}^n$, let $\mathbf{g} \sim \mathcal{N}(0, I_n)$ and $\mathbf{h} \sim \mathcal{N}(0, I_d)$ be two Gaussian vectors in $\mathbb{R}^n$ and $\mathbb{R}^d$ respectively. Then $\tilde{L}_\lambda(r, s)$ is defined as

$$\tilde{L}_\lambda(r,s) = \frac{\lambda r}{2} + \min_{\boldsymbol{\beta}} \ \max_{\mathbf{u}} \left\{ \psi(\boldsymbol{\beta}, \mathbf{u}) + \frac{\sqrt{r}}{2d} \sum_{i=1}^{n} u_i g_i + \frac{||\mathbf{u}||_2 \mathbf{h}^T \boldsymbol{\beta}}{2d\sqrt{d}} \right\}, \tag{10}$$

where

$$\psi(\boldsymbol{\beta}, \mathbf{u}) := \frac{1}{d}\sum_{i=1}^{n} \left(\frac{u_i (s + y_i b)}{2} - \tilde{l}(u_i) + \frac{u_i y_i}{2\sqrt{d}} |\mathbf{x}_i^T \boldsymbol{\beta} + b\sqrt{d}| \right). \tag{11}$$

According to the CGMT theorem, for any constant $c \in \mathbb{R}$, we have

$$\mathbb{P}(L_\lambda(r,s) < c) \leqslant 2\mathbb{P}(\tilde{L}_\lambda(r,s) < c). \tag{12}$$

Then, $\tilde{L}_\lambda(r,s) \geqslant \dfrac{\lambda r}{2} + \min\limits_{\boldsymbol{\beta}} \ \max\limits_{\mathbf{u}, u_i y_i > 0} \left\{ \psi(\boldsymbol{\beta}, \mathbf{u}) + \dfrac{\sqrt{r}}{2d} \sum\limits_{i=1}^{n} u_i g_i + \dfrac{||\mathbf{u}||_2 \mathbf{h}^T \boldsymbol{\beta}}{2d\sqrt{d}} \right\}.$

Using the convexity of the loss function and following the standard application procedures (see Mignacco et al. (2020) and page 5 of supplementary reading Mignacco et al. (2020)) of the CGMT theorem we get,

$$\mathbb{P}(\tilde{L}_\lambda(r,s) < c) \leqslant \mathbb{P}(\omega_\lambda^{(d)}(r,s) < c). \tag{13}$$

Now, combining the inequalities in (12) and (13) yields

$$\mathbb{P}(L_\lambda(r,s) < c) \leqslant 2\mathbb{P}(\omega_\lambda^{(d)}(r,s) < c). \tag{14}$$

Next we solve the maximization problem on Legendre transformation variable $\mathbf{u} \in \mathbb{R}^n$ in $\omega_\lambda^{(d)}(r,s)$. We use the Lagrange multiplier method for this maximization as it is a simple optimization problem with one constraint. Let p^* to be the optimal solution of the maximization problem

$$p^* := \max_{\substack{\mathbf{u}, \\ u_i y_i > 0}} \left\{ \frac{1}{d} \sum_{i=1}^{n} \left(\frac{u_i}{2}(s + y_i b + \sqrt{r} g_i) - \tilde{l}(u_i) \right) + \left(\frac{s\boldsymbol{\eta}^T \mathbf{h}}{d} - \sqrt{r - s^2} \right) \frac{||\mathbf{u}||_2}{2\sqrt{d}} \right\}.$$

Let τ_i be the non-negative Lagrange multipliers associated with $u_i y_i > 0$ for $1 \leqslant i \leqslant n$. Then the Lagrangian function can be written as

$$L(\mathbf{u}, \boldsymbol{\tau}) = \frac{1}{d} \sum_{i=1}^{n} \left(\frac{u_i}{2}(s + y_i b + \sqrt{r} g_i) - \tilde{l}(u_i) \right) + \left(\frac{s\boldsymbol{\eta}^T \mathbf{h}}{d} - \sqrt{r - s^2} \right) \frac{||\mathbf{u}||_2}{2\sqrt{d}} + \frac{1}{d} \sum_{i=1}^{n} \tau_i u_i y_i.$$

Then, $p^* = \max_{\tau_i \geqslant 0, \mathbf{u}} L(\mathbf{u}, \boldsymbol{\tau})$. Let the optimal values of $(\mathbf{u}, \boldsymbol{\tau})$ be $(\mathbf{u}^*, \boldsymbol{\tau}^*) = \arg\max_{\tau_i \geqslant 0, \mathbf{u}} L(\mathbf{u}, \boldsymbol{\tau})$. Then following Theorem 18.5 in Simon et al. (1994) with above $L(\mathbf{u}, \boldsymbol{\tau})$, we have

(a) $\dfrac{\partial L(\mathbf{u}^*, \boldsymbol{\tau}^*)}{\partial u_i^*} = \frac{1}{d}\left(\frac{1}{2}(s + y_i b + \sqrt{r} g_i) - \tilde{l}'(u_i^*) \right) + \left(\frac{s\boldsymbol{\eta}^T \mathbf{h}}{d} - \sqrt{r - s^2} \right) \frac{u_i^*}{2\sqrt{d}||\mathbf{u}^*||_2} + \frac{\tau_i^*}{d} y_i = 0,$

(b) $\tau_i^*(u_i^* y_i) = 0,$

(c) $\tau_i^* \geqslant 0$

(d) $u_i^* y_i > 0, \quad \text{for all} \quad 1 \leqslant i \leqslant n.$

From (b) and (d) we have, $\tau_i^* = 0$ for $1 \leqslant i \leqslant n$ and this agrees with (c). Along with this, we rewrite (a) as,

$$\left(\sqrt{r - s^2} - \frac{s\boldsymbol{\eta}^T \mathbf{h}}{d} \right) \frac{\sqrt{d}}{2||\mathbf{u}^*||_2} u_i^* + \tilde{l}'(u_i^*) = \frac{1}{2}(s + y_i b + \sqrt{r} g_i). \tag{15}$$

This creates a system of equations on $u_i^*, i = 1, \ldots, n$. Recall that $\mathbf{g} = (g_1, \ldots, g_n)$ is a random vector with i.i.d. standard normal entries and $y_i = \pm 1$ is independent of g_i. Then we have the solution to the maximization problem.

$$p^* = \frac{1}{d} \sum_{i=1}^{n} \left(\frac{u_i^*}{2}(s + y_i b + \sqrt{r} g_i) - \tilde{l}(u_i^*) \right) + \left(\frac{s\boldsymbol{\eta}^T \mathbf{h}}{d} - \sqrt{r - s^2} \right) \frac{||\mathbf{u}^*||_2}{2\sqrt{d}}.$$

We plug p^* back into $\omega_\lambda^{(d)}(r,s)$ and simplify it using the relationship in (15).

$$\omega_\lambda^{(d)}(r,s) = \frac{\lambda r}{2} + \frac{1}{d} \sum_{i=1}^{n} (u_i^* \tilde{l}'(u_i^*) - \tilde{l}(u_i^*)). \tag{16}$$

This $\omega_\lambda^{(d)}(r,s)$ in (16) is a lower bound for the AO problem. This quantity depends on $\mathbf{u}^*$ which satisfies (15).

Scalar Change of Variables by the Substitution $v_i = \tilde{l}'(u_i^*)$

Here we introduce a scalar change of variables by $v_i = \tilde{l}'(u_i^*)$. Properties of Legendre transformation help to verify the following two identities and the detailed steps are shown in supplementary material 1 Section S.2.

$$\begin{aligned} u_i^* &= l'(v_i) \\ u_i^* \tilde{l}'(u_i^*) - \tilde{l}(u_i^*) &= l(v_i). \end{aligned} \tag{17}$$

For each $i \in [n]$, $v_i \in \mathbb{R}$ and we denote them by vector $\mathbf{v} \in \mathbb{R}^n$. Moreover by $l'(\mathbf{v})$, we mean the element-wise derivative on the vector $\mathbf{v}$. The new vector $\mathbf{v}$ changes the expression in (15) to the following.

$$\Big(\sqrt{r-s^2} - \frac{s\boldsymbol{\eta}^T\mathbf{h}}{d}\Big)\frac{\sqrt{d}}{2||l'(\mathbf{v})||_2}\, l'(v_i) + v_i = \frac{1}{2}(s + y_i b + \sqrt{r}g_i). \tag{18}$$

For easy reference we denote, $\gamma = \Big(\sqrt{r-s^2} - \frac{s\boldsymbol{\eta}^T\mathbf{h}}{d}\Big)\frac{\sqrt{d}}{2||l'(\mathbf{v})||_2}$. (19)

Hence (18) can be re-written as, $\gamma l'(v_i) + v_i = \frac{1}{2}(s + y_i b + \sqrt{r}g_i)$. (20)

Applying the second relation in (17) to $\omega_\lambda^d(r,s)$ in (16) we get the following expression for $\omega_\lambda^d(r,s)$.

$$\omega_\lambda^{(d)}(r,s) = \frac{\lambda r}{2} + \frac{\alpha}{n}\sum_{i=1}^{n} l(v_i). \tag{21}$$

Now the calculations only require a margin based loss function and future calculations does not require finding the convex conjugates of the loss functions. Using the CGMT, we can show that $\omega_\lambda^{(d)}(r,s)$ can be used as a candidate to observe the asymptotic behavior of the global training loss (see supplementary material 1 Section S.3 for more details). Since we already have an expression for $\omega_\lambda^{(d)}(r,s)$ in (21), first we minimize it to find $\omega_\lambda^*(r,s)$ and finally consider the high dimensional behavior by sending $n, d \to \infty$.

Minimizing $\omega_\lambda^{(d)}(r,s)$ to Find r, s and b

As explained in Theorem 1, the generalization error depends on the fixed values of r, s and b. Through the training procedure of the model, we find the fixed quantities r, s and b that correspond to the minimum possible training error of the model. Hence we work on the minimization of $\omega_\lambda^{(d)}(r,s)$ in (21) with constraint $s^2 \leqslant r$ to find $\omega_\lambda^*(r,s)$, which can be done by setting the derivatives of $\omega_\lambda^{(d)}(r,s)$ with respect to r, s and b to zero. We use the relationship introduced in (20) and the following relationship derived from (19) to find the partial derivatives of v and γ as needed.

$$4\alpha\gamma^2||l'(\mathbf{v})||_2^2 = n\Big(\sqrt{r-s^2} - \frac{s\boldsymbol{\eta}^T\mathbf{h}}{d}\Big)^2. \tag{22}$$

More details on calculating the following derivatives are explained in the supplementary material 1 Section S.5. The derivative of (21) with respect to r is given by,

$$\frac{\alpha}{\sqrt{r}}\frac{1}{n}\sum_{i=1}^{n} g_i(w_i - 2v_i) = -4\lambda\gamma + 1 - \frac{s\boldsymbol{\eta}^T\mathbf{h}}{d\sqrt{r-s^2}}, \tag{23}$$

where $w_i = (s + y_i b + \sqrt{r}g_i)$ and w_i follows a normal distribution with mean $s + y_i b$ and standard deviation $\sqrt{r}$ conditioned on y_i for each $1 \leqslant i \leqslant n$. The derivative of (21) with respect to s is given by,

$$-\alpha\frac{1}{n}\sum_{i=1}^{n}(w_i - 2v_i) = s + G, \tag{24}$$

where $G = \sqrt{r-s^2}\frac{\boldsymbol{\eta}^T\mathbf{h}}{d} - \frac{s^2\boldsymbol{\eta}^T\mathbf{h}}{d\sqrt{r-s^2}} - s\left(\frac{\boldsymbol{\eta}^T\mathbf{h}}{d}\right)^2$. The derivative of (21) with respect to b is given by,

$$\sum_{i=1}^{n} y_i(w_i - 2v_i) = 0. \tag{25}$$

For fixed n and d, solving the equations (23), (24) and (25) may give us the optimal r, s and b values which satisfy $\omega_\lambda^*(r,s)$. However, these three equations cannot be solved in general. Hence we select a margin-based convex loss function and solve the equations. Since γ is involved here, we use the relationship in (20) as $2v_i + 2\gamma l'(v_i) = w_i$ when needed.

Since we are interested in high dimensional behavior of r, s, and b, we look at the limiting behavior of the three equations when $n, d \to \infty$. Let r^*, s^*, b^* and γ^* be the values achieved by r, s, b and γ when $n, d \to \infty$.

Finding the Asymptotics of r, s, b and γ

We consider the limits of both sides of (23).

$$\lim_{n,d\to\infty}\left(\frac{\alpha}{\sqrt{r}}\frac{1}{n}\sum_{i=1}^{n} g_i(w_i - 2v_i)\right) = \lim_{n,d\to\infty}\left(-4\lambda\gamma + 1 - \frac{s\boldsymbol{\eta}^T\mathbf{h}}{d\sqrt{r-s^2}}\right).$$

After simplifying both sides it shows the following asymptotic relationship. Notice that the last term in right side goes to zero since $\frac{\boldsymbol{\eta}^T\mathbf{h}}{d} \to 0$ by law of large numbers when $d \to \infty$.

$$\frac{\alpha}{\sqrt{r^*}}\lim_{n,d\to\infty}\left(\frac{1}{n}\sum_{i=1}^{n} g_i(w_i - 2v_i)\right) = -4\lambda\gamma^* + 1. \tag{26}$$

Next, we consider the limits of both sides of (24).

$$-\alpha\lim_{n,d\to\infty}\left(\frac{1}{n}\sum_{i=1}^{n}(w_i - 2v_i)\right) = s^*. \tag{27}$$

Notice that when $n, d \to \infty$, the term $G = \sqrt{r-s^2}\frac{\boldsymbol{\eta}^T\mathbf{h}}{d} - \frac{s^2\boldsymbol{\eta}^T\mathbf{h}}{d\sqrt{r-s^2}} - s\left(\frac{\boldsymbol{\eta}^T\mathbf{h}}{d}\right)^2$ also goes to zero since $\frac{\boldsymbol{\eta}^T\mathbf{h}}{d} \to 0$ by law of large numbers. By (25), we have

$$\lim_{n,d\to\infty}\left(\frac{1}{n}\sum_{i=1}^{n} y_i(w_i - 2v_i)\right) = 0. \tag{28}$$

Since γ is involved in above expressions as function of $\mathbf{v}$, we consider the high dimensional behavior of (22). First we rewrite it using the expression (S14).

$$\alpha\frac{1}{n}\sum_{i=1}^{n}(w_i - 2v_i)^2 = \left(\sqrt{r-s^2} - \frac{s\boldsymbol{\eta}^T\mathbf{h}}{d}\right)^2.$$

Then the limits give

$$\alpha\lim_{n,d\to\infty}\left(\frac{1}{n}\sum_{i=1}^{n}(w_i - 2v_i)^2\right) = r^* - (s^*)^2. \tag{29}$$

For a specific loss function, we can solve the above four equations (26), (27), (28) and (29) to find r^*, s^*, γ^* and b^*. In summary, we have the following main result of the paper.

Theorem 2. *Let* $f(\mathbf{x}_i) = \sigma(\frac{\mathbf{x}_i^T\boldsymbol{\beta}}{\sqrt{d}} + b)$ *be a two-layer neural network with ReLU activation function. We minimize the empirical risk (3). Denote* $r = \frac{1}{d}||\boldsymbol{\beta}||_2^2$ *and* $s = \frac{1}{d}\boldsymbol{\beta}^T\boldsymbol{\eta}$. *Assume that data are generated from the teacher model in (2), the values of* α *and regularization parameter* λ *are known, and* l *is the square loss function. Let* $w_i \sim \mathcal{N}(s + y_i b, r)$ *conditional on* $y_i = \pm 1$ *and* $g_i \sim \mathcal{N}(0,1)$ *for all* $i = 1, \ldots, n$. $v_i \in \mathbb{R}$ *satisfies the following relationship for any* $\gamma > 0$.

$$\gamma l'(v_i) + v_i = \frac{1}{2}w_i. \tag{30}$$

For fixed ratio α, *the quantities* r, s, b *and* γ *converge to fixed quantities* r^*, s^*, b^* *and* γ^* *as* $n, d \to \infty$. *These fixed quantities are given by the solutions of the following equations.*

$$\frac{\alpha}{\sqrt{r^*}} \lim_{n,d\to\infty} \left(\frac{1}{n}\sum_{i=1}^{n} g_i(w_i - 2v_i)\right) = -4\lambda\gamma^* + 1, \tag{31}$$

$$-\alpha \lim_{n,d\to\infty} \left(\frac{1}{n}\sum_{i=1}^{n} (w_i - 2v_i)\right) = s^*, \tag{32}$$

$$\lim_{n,d\to\infty} \left(\frac{1}{n}\sum_{i=1}^{n} y_i(w_i - 2v_i)\right) = 0, \tag{33}$$

$$\alpha \lim_{n,d\to\infty} \left(\frac{1}{n}\sum_{i=1}^{n} (w_i - 2v_i)^2\right) = r^* - (s^*)^2. \tag{34}$$

These quantities r^*, s^* *and* b^* *are used to calculate the limit of generalization error of* $f(\mathbf{x}_i)$ *in Theorem 1, for a fixed* α *when* $n, d \to \infty$.

6 Theoretical Generalization Error Curves for Square Loss

Here we apply Theorem 1 and Theorem 2 with square loss and analyze the theoretical generalization error as a function of α for different values of λ and ρ_1. The calculation and the R code for the below work is presented in the supplementary material 1 Section S.6 and supplementary material 2 Section S.1 respectively. The regularization is used to balance over-fitting and under-fitting in the model. Our strategy prevents us from having $\lambda = 0$ hence we use negligible λ values to mean no regularization and vice-versa. Also, observe that the number of parameters in the model is same as the model dimension ($p = d$).

Figure 4 shows the test error as a function of $\alpha = n/d$ in the model (1) under the square loss function. Here $\rho_1 = \rho_{-1} = 0.5$. Thus, the two classification groups have the same probability and we take the regularization parameter to be very small: $\lambda = 10^{-5}$. As we can see, the test error first decreases, then increases again and reaches its peak when $\alpha = 1$, and later when $\alpha > 1$, the test error steadily keeps decreasing.

- When $\alpha < 1$, i.e., when $n < d$, we have the over-parameterized region of the model in (1). The test risk decreases first and then rises up again tracing a U-shaped curve. The first local minimum occurs before $\alpha = 1$ and we identify this minimum as the sweet-spot.
- Next, the increasing test error reaches its peak when $\alpha = 1$, that is, when $n = d$. Hence, maximum test error occurs when $n = d$ and we identify this point as the interpolation threshold.

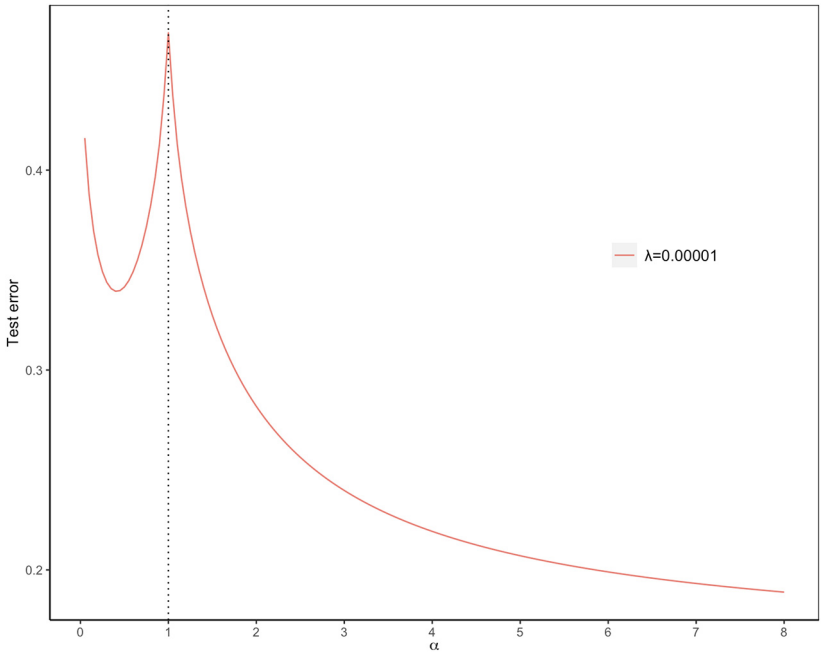

Figure 4: Test error of model (1) with square loss as a function of α with low regularization ($\lambda = 10^{-5}$) and with $\rho_1 = 0.5$.

- Finally, when $\alpha > 1$, i.e., when $n > d$, the model enters the under-parameterized region and the test error decreases monotonically for increasing α values. With more data the model overfits resulting in a lower test error and the best performance of the model is achieved in this region agreeing with the classic idea "more data is always better".

So it is clear that we can jump from over-parameterized region to under-parameterized region by increasing α while observing the peak in between. The curve manifests the classical U-shaped curve in the over-parametrized region and the long plateau in the under-parametrized region. This is a noticeable difference between the double descent behavior observed as a function of model capacity (n and d are fixed, numbers of parameters are varying) and this ratio-based double descent behavior (the test error is a function of $\alpha = n/d$, with n and d going to infinity).

The over-parameterized region has its own local minima that corresponds to a better-performing model, and in under-parameterized region we have the flexibility to pick the best model since the test error is decreasing monotonically. For this specific binary classification model, the best model comes from the under-parameterized region and the test error of that model is comparatively smaller than any model coming from the over-parameterized region. We do not work on training of the model using iterative procedures like gradient descent and, therefore, we cannot comment on the position of global minima, as it depends on the composition of training data.

6.1 Effect of Regularization on Double Descent Behavior

Figure 5 illustrates how the increasing regularization can smooth out the peak in the generalization error curve for the case $\rho_1 = \rho_{-1} = 0.5$. We did not allow λ to be exactly zero for the numerical stability of the algorithm we followed. Peak is clearly visible with lower regularization

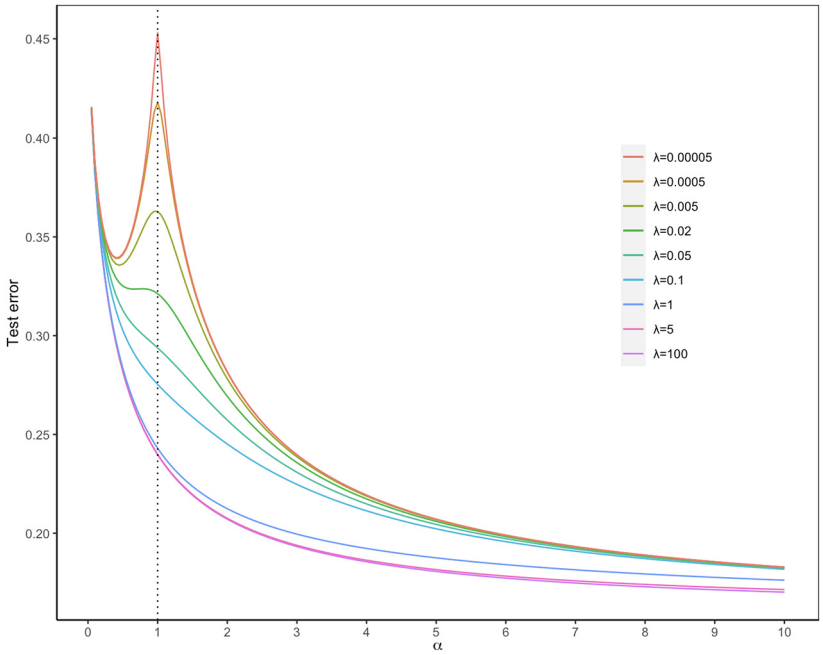

Figure 5: Test error of model (1) for square loss as a function of α with $\rho_1 = 0.5$ and with varying regularization.

like $\lambda = 0.005$ and with further increase in regularization, the peak gets smoothed out and test error decreases monotonically.

Moreover, this depicts the significance of having regularization in classification tasks to achieve better results. Similar studies done in Nakkiran et al. (2020) confirm that most under-regularized linear regression models observe this type of double descent curve. In this setting, when $\rho_1 = 0.5$, it seems that the higher λ values, the lower test errors. However, it does not improve much if λ is too large. We suggest to use $\lambda = 5$ with equal class sizes.

6.2 Different ρ_1 Values with Weak and Strong Regularization

Test error curves from the previous part correspond to equal cluster sizes with $\rho_1 = 0.5$. Now we study the double descent behavior for uneven cluster sizes.

As illustrated in Figure 6 below, when $\rho_1 = 0.7$, then, for lower regularization values, the test error starts at 0.3 and then goes down monotonically, and the double descent behavior can be clearly observed. But when $\lambda \geqslant 1$, the test error keeps unchanged until a specific α value is reached and then starts decreasing monotonically as α gets bigger and bigger. Even though the higher regularization values acted similar on cases with $\rho_1 = 0.5$, when the cluster sizes are uneven, too much regularization does not yield favorable results.

Figure 7 shows the same curves in Figure 6 for higher α values to observe the downward trend for highly regularized models. We can see that even when $\lambda = 100$, the test error starts decreasing when $\alpha \approx 60$. It is clear that the higher the regularization is, the higher values of α are needed to achieve a considerably lower test error, as the rate of decrease is very low. In summary, we shall have some suitable regularization ($0.1 \leqslant \lambda \leqslant 1$) to have best performance.

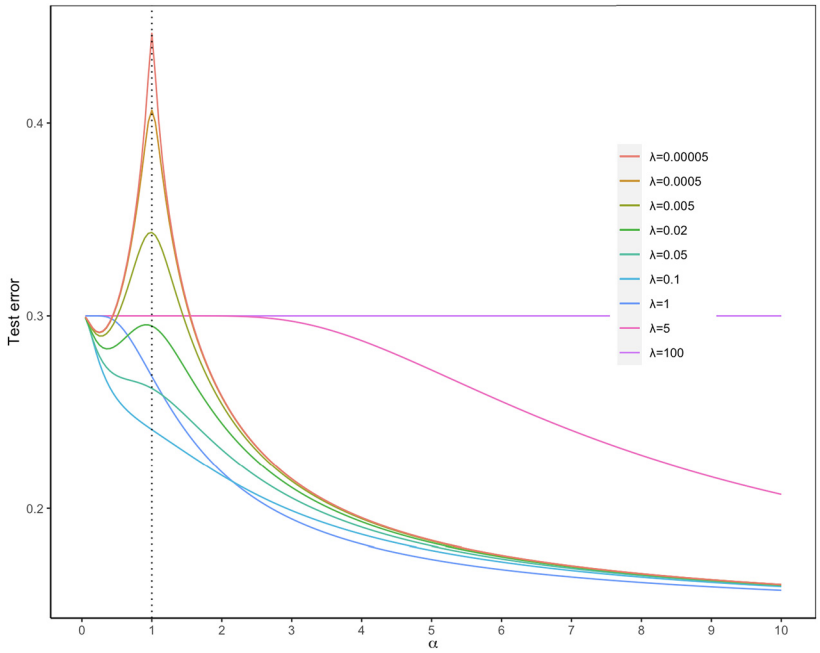

Figure 6: Test error of model (1) for square loss as a function of α with $\rho_1 = 0.7$, and with varying regularization values.

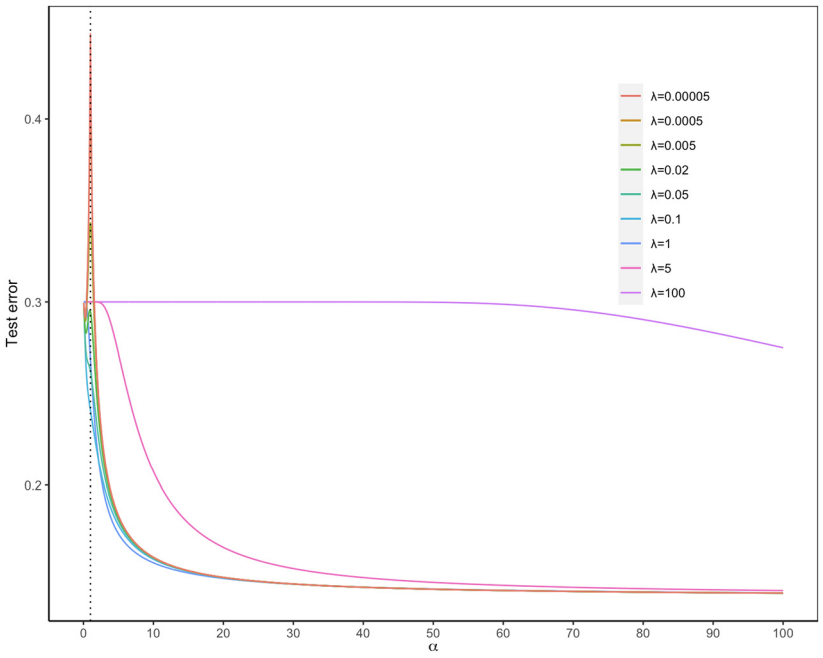

Figure 7: Test error of model (1) for square loss as a function of higher α values with $\rho_1 = 0.7$ and with varying regularization values.

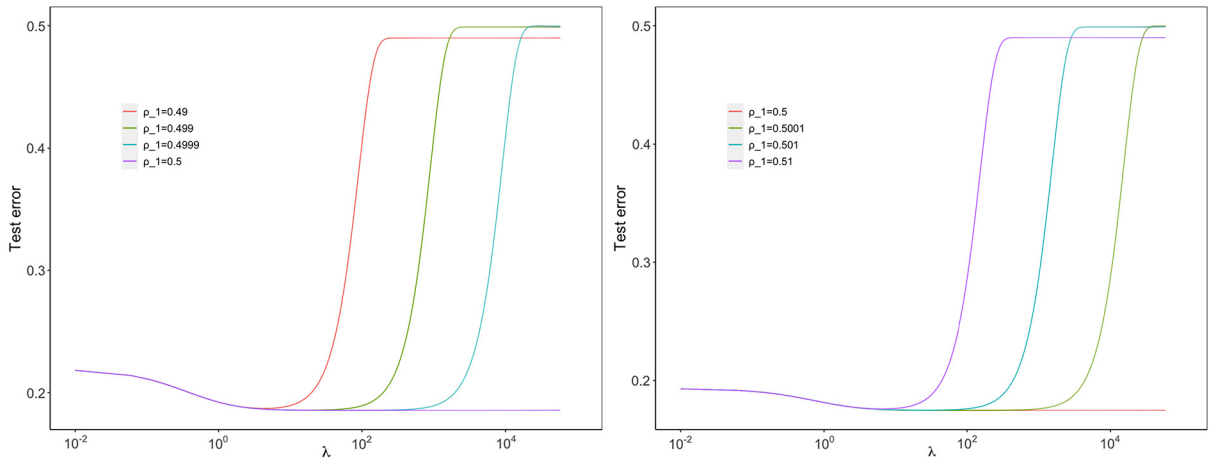

Figure 8: Test error of model (1) as a function of regularization. We fix the values $\alpha = 4$ (left) and $\alpha = 7$ (right) and consider different values of ρ_1 that are getting closer to 0.5.

6.3 Test Error of Different Cluster Sizes for Increasing Regularization when α Is Fixed

Here we fix α and view the generalization error as a function of λ. In Figure 5 we noticed that when $\rho_1 = 0.5$, the generalization error is not sensitive to higher regularization. Hence we consider the cluster sizes close to 0.5 to observe the change in the test error when regularization increases. According to Figure 8, for each ρ_1 close to 0.5, the generalization error decreases and increases again to reach a plateau. The minimum possible test error is achieved at some finite regularization value (λ^*) and when $\lambda > \lambda^*$ test error increases again. But when ρ_1 is exactly 0.5, the test error keeps steady at a low level after a certain λ^*.

7 Conclusion

In this paper we work on ratio-wise double descent behavior for a two layer neural network classification model. The test error is a function of the ratio α between the sample size n and the model dimension d and we consider asymptotics of the test error when $n, d \to \infty$. We derive the exact theoretical test error of the model in Theorem 1. Next, we perform empirical risk minimization procedure to find the unknown quantities in the test error formula. An upper bound on the local training loss is used as the candidate to observe the behavior of the asymptotics of unknown quantities. We outline these findings in Theorem 2.

Utilizing the results from Theorems 1 and 2 and using the square loss function, we plot the test error curve as a function of α and observe the double descent behavior when the regularization is very low. The curve's peak happens when the sample size equals the model dimension. We also notice that when the regularization increases, the peak of the curve disappears and the test error decreases monotonically. When the cluster sizes are equal, the effect of strong regularization is not significant, while with uneven cluster sizes, under strong regularization, the test error is steady at first, and then it starts decreasing as a higher value of α is reached.

We confirm the existence of the double descent phenomenon in the test error for two-layer neural network model with low level of regularization. In this case, our theoretical results confirm that when the test error is analyzed ratio-wise, the best performance of the model is achieved

after the peak of the test error in the under-parameterized region. We then analyze the effect of regularization on the double descent curve. We suggest to use a suitable level of regularization in the empirical risk to have ideal test error for different ratios α, no matter whether the cluster sizes are even or not. With this optimal test error, the double descent phenomenon disappears. Instead, the test error decreases monotonically and it is consistent with the classical idea that more data is always better.

Here we used l_2 regularization to support the fact that optimal regularization can mitigate the double descent in binary classification models. In future research projects we plan to investigate the double descent phenomenon with other regularization methods like Lasso or elastic-net, which is a combination of both l_2 and Lasso.

Supplementary Material

We have included two supplementary files where Supplementary material 1 contains detailed calculations, theorems and proofs and Supplementary material 2 contains the R/RStudio codes used to draw the curves presented in the paper.

Acknowledgments

The authors thank the editor, associate editor, and referees for their constructive comments which has led to significant improvement of this paper.

Funding

The research of Hailin Sang is partially supported by the Simons Foundation Grant 586789, USA.

References

Advani MS, Saxe AM, Sompolinsky H (2020). High-dimensional dynamics of generalization error in neural networks. *Neural Networks*, 132: 428–446. https://doi.org/10.1016/j.neunet.2020.08.022

Amir I, Koren T, Livni R (2021). Sgd generalizes better than gd (and regularization doesn't help). In: *Conference on Learning Theory* (M Belkin, S Kpotufe, eds.), 63–92. PMLR.

Belkin M, Hsu D, Ma S, Mandal S (2019). Reconciling modern machine-learning practice and the classical bias–variance trade-off. *Proceedings of the National Academy of Sciences*, 116(32): 15849–15854.

Bhavsar H, Ganatra A (2012). A comparative study of training algorithms for supervised machine learning. *International Journal of Soft Computing and Engineering*, 2(4): 2231–2307.

Bonaccorso G (2018). *Machine Learning Algorithms: Popular Algorithms for Data Science and Machine Learning*. Packt Publishing Ltd.

D'Ascoli S, Refinetti M, Biroli G, Krzakala F (2020). Double trouble in double descent: Bias and variance(s) in the lazy regime. In: *Proceedings of the 37th International Conference on Machine Learning* (HD III, A Singh, eds.), volume 119 of Proceedings of Machine Learning Research, 2280–2290. PMLR.

Deng Z, Kammoun A, Thrampoulidis C (2022). A model of double descent for high-dimensional binary linear classification. *Information and Inference*, 11(2): 435–495.

Geiger M, Jacot A, Spigler S, Gabriel F, Sagun L, d'Ascoli S, et al. (2020). Scaling description of generalization with number of parameters in deep learning. *Journal of Statistical Mechanics: Theory and Experiment*, 2020(2): 023401.

Hutter F, Kotthoff L, Vanschoren J (2019). *Automated Machine Learning: Methods, Systems, Challenges*. Springer Nature.

Kini GR, Thrampoulidis C (2020). Analytic study of double descent in binary classification: The impact of loss. In: *2020 IEEE International Symposium on Information Theory (ISIT)*, 2527–2532. IEEE.

Lee EH, Cherkassky V (2024). Understanding double descent using vc-theoretical framework. *IEEE Transactions on Neural Networks and Learning Systems*, 169: 242–256.

Mahesh B, et al. (2020). Machine learning algorithms-a review. *International Journal of Science and Research*, 9(1): 381–386.

Mignacco F, Krzakala F, Lu Y, Urbani P, Zdeborova L (2020). The role of regularization in classification of high-dimensional noisy Gaussian mixture. In: *Proceedings of the 37th International Conference on Machine Learning* (HD III, A Singh, eds.), volume 119 of Proceedings of Machine Learning Research, 6874–6883. PMLR.

Nakkiran P (2019). More data can hurt for linear regression: Sample-wise double descent. arXiv preprint: https://arxiv.org/abs/1912.07242.

Nakkiran P, Kaplun G, Bansal Y, Yang T, Barak B, Sutskever I (2021). Deep double descent: Where bigger models and more data hurt. *Journal of Statistical Mechanics: Theory and Experiment*, 2021(12): 124003.

Nakkiran P, Venkat P, Kakade S, Ma T (2020). Optimal regularization can mitigate double descent. arXiv preprint: https://arxiv.org/abs/2003.01897.

Simon CP, Blume L, et al. (1994). *Mathematics for Economists*, volume 7. Norton, New York.

Spigler S, Geiger M, d'Ascoli S, Sagun L, Biroli G, Wyart M (2019). A jamming transition from under-to over-parametrization affects generalization in deep learning. *Journal of Physics A: Mathematical and Theoretical*, 52(47): 474001.

Thrampoulidis C, Oymak S, Hassibi B (2014). The gaussian min-max theorem in the presence of convexity. arXiv preprint: https://arxiv.org/abs/1408.4837.

Thrampoulidis C, Oymak S, Hassibi B (2015). Regularized linear regression: A precise analysis of the estimation error. In: *Conference on Learning Theory* (P Grünwald, E Hazan, S Kale, eds.), volume 40, 1683–1709. PMLR.