UNIVERSITY OF CALIFORNIA
SANTA CRUZ

UNDERSTANDING THE ROLE OF OPTIMIZATION AND LOSS FUNCTION IN DOUBLE DESCENT

A thesis submitted in partial satisfaction
of the requirements for the degree of

MASTER OF SCIENCE

in

COMPUTER SCIENCE AND ENGINEERING

by

Chris Yuhao Liu

June 2023

The Thesis of Chris Yuhao Liu
is approved:

Professor Jeffrey Flanigan, Chair

Professor Yang Liu

Professor Leilani Gilpin

Professor Vaggos (Evangelos) Chatziafratis

Peter Biehl
Vice Provost and Dean of Graduate Studies

Table of Contents

List of Figures

Abstract

Understanding the Role of Optimization and Loss Function in Double Descent

by

Chris Yuhao Liu

Double descent has emerged as a fascinating phenomenon that has been observed across a range of tasks, model architectures, and training paradigms. When double descent occurs, instead of decreasing monotonically, the generalization error initially decreases, then increases as it enters a critical parameterized regime, and finally decreases again. Despite its ubiquity, simple explicit regularization techniques like weight decay and early stopping have been successful in reducing double descent in both theoretical and practical contexts. However, we observe that, in realistic settings, double descent is reduced or does not occur even without any form of explicit regularization. This observation raises a key question: If overfit models do not exhibit the double descent phenomenon in practice, why not?

We identify two key reasons: 1) the use of poor optimizers that struggle to land at a low-loss local minimum even though they obtain zero training error, and 2) the presence of an exponential tail in the shape of the loss function. We further show that, given a sufficient number of iterations, poor optimizers can start to recover the peak. However, exponential-tail loss functions tend to be much more resistant to the peaking behavior even in the long term (when models are extremely overfit). Additionally, we show that

loss functions suffering from the double descent phenomenon (e.g., MSE loss) can be made to exhibit monotonicity, that is no peaking behavior, when they are modified to have an exponential tail.

To validate our findings, we conduct experiments on a wide range of regression and classification loss functions using random feature models and two-layer neural networks trained on realistic datasets. Our results confirm the influence of the two factors identified above on the peaking behavior. These findings offer new insights into the phenomenon of double descent, which is crucial for understanding generalization in machine learning.

For my family, mentors, and friends.

Acknowledgments

I am deeply and sincerely grateful to my advisor, Jeffrey Flanigan, for his unwavering dedication and invaluable mentorship throughout my Master's journey. From the very beginning, he wholeheartedly introduced me to the world of research and has since been a constant source of guidance and support. Words cannot adequately express the extent of my gratitude for his unwavering commitment to my success and tireless efforts in shaping my academic growth, which I hold in the highest regard. I am truly humbled and thankful for the immeasurable wealth of knowledge and expertise he has shared, enriching my understanding and undoubtedly leaving an indelible mark on my future endeavors.

I would also like to extend my heartfelt gratitude to Yang Liu, whose pivotal role in my academic journey cannot be overstated. From teaching my introductory course in machine learning to acquainting me with more advanced concepts during my Master's studies, his expertise and unwavering commitment have profoundly influenced my grasp of the subject matter. Although we did not directly collaborate on the project that eventually became this thesis, our interactions and collaboration have been immensely rewarding. I am sincerely grateful for his guidance, mentorship, and for instilling in me a deep appreciation for the possibilities within this discipline.

I want to express my sincere appreciation for the invaluable discussions, thought-provoking questions, and constructive feedback I have received from my colleagues

at JLab. Brendan King, Changmao Li, Brian Mak, Nilay Patel, Geetanjali Rakshit, Rongwen Zhao, Zekun Zhao, and Chris Toukmaji have all played significant roles in shaping my understanding and broadening my horizons. Their diverse perspectives, expertise, and willingness to engage in intellectual discourse have been instrumental in my academic growth. The collaborative environment at JLab has fostered a sense of camaraderie and a culture of intellectual growth, for which I am truly thankful.

I would also like to extend my sincere appreciation to the members of the thesis reading committee, Yang Liu, Leilani Gilpin, and Vaggos Chatziafratis. Their invaluable contributions, insightful discussions, and constructive feedback have been instrumental in shaping this thesis. I am truly grateful for their generous allocation of time and meticulous evaluation, which have significantly enhanced the quality and depth of this work.

Lastly, I am profoundly grateful to my parents for their unwavering love and support, particularly during the past six years when I have been living in a different country away from home. Despite the physical distance, their constant encouragement, understanding, and belief in my abilities have been a tremendous source of strength throughout this journey. Their dedication and selflessness have empowered me to pursue my dreams with confidence, and I am forever indebted to them for their enduring presence in my life.

Chapter 1

Introduction

The phenomenon of double descent (Belkin et al., 2018), where the generalization error first decreases, increases, and then decreases again as the model size P surpasses the dataset size N, has attracted a lot of attention from the machine learning community due to its contradiction with our traditional understanding. In traditional thinking, as we increase the complexity of a model beyond the size of the dataset it is trained on, overfitting leads to a decline in generalization performance. However, the double descent phenomenon challenges this established belief by demonstrating that after an initial decrease, the generalization error actually increases and then decreases again as the model size surpasses the dataset size. This unexpected behavior calls into question our existing understanding and demands a reevaluation of the true nature of generalization in both under- and over-parameterized regimes. Over the past few years, the double

descent phenomenon has been observed in various models (Nakkiran et al., 2020a; Chang et al., 2020; Sahraee-Ardakan et al., 2021; Fonseca and Guidetti, 2022) and learning paradigms (Nakkiran et al., 2020a; Rice et al., 2020; Hassani and Javanmard, 2022; Dar and Baraniuk, 2020; Cotter et al., 2021). Previous research has contributed to our understanding of the double descent phenomenon in various contexts and from different perspectives, including bias-variance trade-off tools (Yang et al., 2020; d'Ascoli et al., 2020a), VC theory (Lee and Cherkassky, 2022), condition numbers (Kuzborskij et al., 2021; Schaeffer et al., 2023), and aspects of optimization (Kini and Thrampoulidis, 2020; Gamba et al., 2022, 2023). Meanwhile, techniques to mitigate this phenomenon have also been proposed to escape the curse of the critically parameterized models and recover monotonicity in the generalization error. In fact, ℓ_2 regularization , a classic regularization method that penalizes squared magnitudes of the model coefficients, has been shown to be a handy and effective approach for this goal (Mei and Montanari, 2019; Nakkiran et al., 2020a; d'Ascoli et al., 2020a; Lin and Dobriban, 2020; Kan et al., 2020; Qu'etu and Tartaglione, 2023a,b). However, it is worth noting that double descent does not always occur even without any form of explicit regularization, even if models around the interpolation threshold overfit the training data, as is often indicated by near 0 training error. For example, in figures 4, 5, and 19 of Nakkiran et al. (2020a), with a converged model, the double descent peak is almost invisible, and only under a sufficient level of label noise can we see the first ascent of the test error. Similarly, this has also

been observed and discussed in many other works (Yang et al., 2020; d'Ascoli et al., 2020a; Holzmüller, 2020; Maddox et al., 2020; Sa-Couto et al., 2022; Kuzborskij et al., 2021; Zavatone-Veth et al., 2022; Ju et al., 2023).

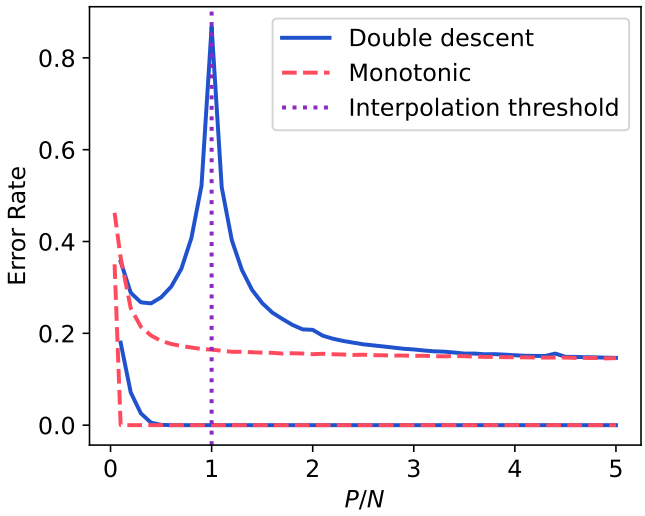


Figure 1.1: An illustration of double descent occurring and not occurring. The x-axis shows the overparameterization ratio, which is the ratio between the model size P and the training dataset size N. The y-axis shows the training and testing error. In both cases, the model converges to 0 training error around the interpolation threshold ($P = N$), but one peaks when N equals P, and the other exhibits the monotonic decreasing error.

In this thesis, the research question is: Given that previous work have attributed double descent to overfitting (Maddox et al., 2020; d'Ascoli et al., 2020b, 2021), if overfit models do not exhibit the double descent phenomenon in practice, why not? We answer the question by conducting a systematic study of what causes or reduces the peak in double descent.[1] We find that all results can be unified and explained by two key

[1]When we say double descent, we always refer to the peak of a double descent curve around the interpolation threshold. We use "double descent" and "peaking phenomenon" interchangeably in this thesis.

elements: optimization and the exponential tail of the loss function.

Firstly, we observe that the double descent does not occur when **using optimizers that struggle to find a low-loss minimum**, even though they may obtain zero training error in Section 5.2. In practice, a poor performance of the optimizer might be due to ill-conditioned features, insufficient iterations, or a small learning rate, all of which can lead to slow convergence of the optimizer and a poor final loss value. Additionally, at the peak of the double descent curve, a peak simultaneously appears in the condition number, which contributes to slow convergence and poor final training loss value when using gradient-based optimizers. We show that by running the optimizer longer, or by choosing conditions and hyperparameters that lead to faster convergence, we can recover double descent. During this process, we strictly control the confounding variables with potential effects on the peak in all experiments, such as label noise, which could exacerbate the curve, and all forms of weight decay and early stopping, which are known to flatten the peak. This ensures the robustness of our results.

Secondly, inspired by the implicit bias of the cross-entropy loss with gradient descent (Soudry et al., 2018), we identify that it is ultimately **the exponential tail in the shape of the loss function** that hinders the peak. We experimentally find that models cannot be made to exhibit double descent if and only if they are trained with a loss that has an exponential tail in Section 5.3. These are exactly the models that converge to a maximum-margin solution (see Section 5.4). Building on the lessons learned from poor

optimizers, we use a fast convergence setting to exclude factors that hinder the emergence of the peak or slow down overfitting. With cross-entropy loss, we almost always see a monotonicly decreasing error curve, even after increasing the number of iterations by a factor of 10. To verify the effect of the exponential tail property, we modify the MSE loss to have an exponential tail, and the resulting error curve is indeed monotonic. On the other hand, by removing the exponential tail property on the cross-entropy loss, either by label smoothing or by modifying the loss, we can bring back the peaking phenomenon (see Section 5.2). Lastly, we further validate the power of exponential-tail loss functions using a toy support-vector dataset introduced in Soudry et al. (2018). We demonstrate that, by fitting a simple linear/logistic regression problem, models with exponential-tail losses all converge to the max-margin solution, whereas those without perfectly separates all data points but overfit.

Based on our results, we believe that classifiers in practice are unlikely to exhibit the peaking phenomenon. This is because 1) inductive biases and hyperparameters are usually chosen carefully using a validation set to prevent overfitting, and 2) in practical classification tasks, MSE loss is rarely used, and the cross-entropy loss already possesses the advantageous property for monotonicity. Both act as "natural" mitigators of double descent. While no existing theoretical framework explains all our results, we believe our results are particularly useful for suggesting avenues for further theoretical study.

Chapter 2

Existing Theoretical Results

In this chapter, we related our findings to existing theoretical results. We first visit the literature of the implicit bias of logistic loss, which is shown to yield a max-margin separation in the separable case and. We then discuss the connection of the double descent in generalization error and condition number.

2.1 The Implicit Bias of Logistic Loss

While we have not provided rigorous explanation on our findings, we point out that our observations on exponential-tail loss functions consistently align with previous theoretical results addressing double descent for logistic regression. For cross-entropy loss and other exponential tail loss functions, our generalization curves have the same monotonic behavior as the asymptotic generalization performance of the binary hard-

margin SVM for separable data (Deng et al., 2019). The authors derive the asymptotics of a SVM classifier and use it to predict the generalization curve given simulation data, which shows that the test error decreases monotonically. Our results consistently align with Kini and Thrampoulidis (2020) in that the peaking phenomenon is almost always invisible under cross-entropy loss. Empirical observations in both Gerace et al. (2020) with hidden manifold model and d'Ascoli et al. (2021) with logistic regression on real data agree with this finding.

Our findings on loss functions are closely related to the implicit bias of cross-entropy and gradient descent (Soudry et al., 2018). The authors prove that GD converges to max-margin classifier when the loss is smooth, monotonically decreasing, and with an exponential tail. We find that exactly the loss functions which satisfy the above properties are the ones that do not exhibit the peaking phenomena. The result from Soudry et al. (2018) shows that GD with these losses converges to a max-margin classifier. This result, when combined with the result from Deng et al. (2019) showing that the test error for the max-margin classifier for seperable data decreases monotonically, hints at why the peaking phenomenon does not occur for these losses. In Section 5.4, we perform a case study similar to the convergence experiment in Soudry et al. (2018) and show that exponential-tailed loss functions all converge to the max-margin separating hyperplane. Nacson et al. (2019) proves the same implicit bias on logistic regression but with SGD using a constant learning rate. Several other work also proved this implicit bias holds for

two-layer neural networks with various smooth and non-smooth ReLU activations (Lyu et al., 2021; Xu et al., 2018; Chatterji et al., 2020, 2021).

Our results also agree with predictions from the theoretical model of Chatterji and Long (2020). The authors used a data model $x = Uq + \tilde{y}\mu$, where U is a unitary matrix with the center of feature cluster being $\mu \in \mathbb{R}^p$. They assume, for a large constant C,

1. the failure probability is $0 \leq \delta \leq 1/C$;
2. we have $n \geq C \log(1/\delta)$ samples;
3. each with $p \geq C \max\{\|\mu\|^2 n, n^2 \log(n/\delta)\}$ dimension;
4. $\|\mu\|^2 \geq C \log(n/\delta)$.

The authors prove that, with sub-Gaussian data and the above assumptions and $c > 0$, the misclassification error for a max-margin classifier $\mathbf{W}$ for feature $\mathbf{x}$ and target y in the separable case has the form

$$\mathbb{P}_{(x,y)\sim \mathrm{P}}[\mathrm{sign}(\mathbf{W} \cdot \mathbf{x}) \neq y] \leq \eta + \exp\left(-c\frac{\|\mu\|^4}{P}\right)$$

with probability $1 - \delta$. In other words, for the misclassification error to be lower, the $\frac{\|\mu\|^4}{P}$ term has to grow with P, and needs to satisfy $\|\mu\|^4 = \omega(P)$ for the error to decrease. In the case of random features, the feature dimension P increases, so knowing how faster the class separation $\|\mu\|^4$ grows will tell us how the bound changes. We calculate the quantity $\|\mu\|^4/P$ for the ReLU random features on MNIST. We find that $\|\mu\|^4$ grows

faster than P when the features are normalized, or with a large scale γ for $\mathbf{X}$ or a large initialization scale k with the first-layer weight of a random feature model. In these settings, double descent are prone to occur.

While our focus is not explaining why double descent occurs to regression loss (e.g., MSE), our observations do align with the data-dependent bound for least squares with gradient descent derived in Kuzborskij et al. (2021). Specifically, the error bound heavily depends on the minimum eigenvalue of the features, and the error rate increases as the minimum eigenvalules become smaller. Additionally, they also observed that the condition number is driven by the minimum eigenvalue of the features. However, one distinction is that, we observe that ill-conditioned features (with high condition numbers) are not good indicators of double descent, the optimization setting has to be taken into account. We discuss this in detail in the next section.

2.2 Double Descent in Condition Number

The peaking phenomenon not only exists in the generalization error but also in the condition number of the features. Poggio et al. (2019) were the first to establish such a connection by proving that in a system of n equations and d variables, $Ax = b$, the condition number is at its maximum when $n = d$, resulting in a unique inverse of A. Later, Rangamani et al. (2020) demonstrated that the condition number also regulates the stability of the least squares solution, which is why the error peaks at $P = N$. Our results

also agree with Kuzborskij et al. (2021), which examined the gradient descent solution of least squares and found a negative correlation between the minimum eigenvalue of the data and the condition number. They concluded that the minimum eigenvalues contribute to the generalization error on top of the error arising from optimization.

In Figure 5.1, we find that the condition number is largest at $P = N$ (the peak), which agrees with theoretical results. This makes optimization harder at $P = N$, so optimizers are less likely to converge, which causes the peak of double descent to be reduced or disappear. So there a two antagonistic forces at play when we observe the peak: the condition number grows, which make optimization harder (reducing the peak), but if we optimize better by improving the conditioning, using a better optimizer, or running for more steps, we recover the peak (as shown in Figure 5.3). This finding extends the previous ones because we show the interplay of the condition numbers in features and other elements in optimization.

Chapter 3

Related Work

In this chapter, we discuss work related to ours. We first introduce double descent and argue that double descent is indeed a universal phenomenon and exists in various models and learning paradigms. We then discuss existing popular techniques to mitigate double descent and how our findings on optimization can be used as a potential method to achieve the same goal. Lastly, we discuss possible explanations of the double descent phenomenon.

3.1 The Ubiquity of Double Descent

The double descent phenomenon has been observed and theoretically verified in a wide range of settings, from simple linear models (Belkin et al., 2019; Hastie et al., 2019; Mei and Montanari, 2019) to complex deep neural networks (Belkin et al., 2018;

Nakkiran et al., 2020a) on a wide variety of synthetic datasets and real-world tasks across domains. When double descent occurs, the generalization error initially increases as the model size grows and peaks when the model size P equals the dataset size N. In the overparameterization regime where $P > N$, the generalization error decreases.

The initial study of double descent demonstrated that the non-monotonicity is a universal phenomenon in linear models, ensemble methods, and neural networks, trained with regression loss on both synthetic and real-world datasets (Belkin et al., 2018). While Belkin et al. (2018) first coined the term, some prior work had already observed it in minimum norm linear regression ((Moore-Penrose inverse) (Vallet et al., 1989; Opper et al., 1990) and pseudo-Fisher linear discriminant (on real data) (Duin, 2000), as pointed out by Loog et al. (2020).

Subsequent work also reported double descent in KNN Xing et al. (2019) , generalized linear models (Emami et al., 2020), principal component regressors (Xu and Hsu, 2019; Wu and Xu, 2020), margin-based classifiers (Huang and Yang, 2020), random forest (Ribeiro et al., 2020), Gaussian processes (Hodgkinson et al., 2022), envelope models (Kwon and Zou, 2023), random feature model with leave-one-out loss (Bachmann et al., 2022), and recursive feature machines (Gupta et al., 2023).

The same peaking phenomenon has occurred in deeper neural networks. Nakkiran et al. (2020a) conducted a comprehensive study for double descent in convolutional neural networks and Transformers. The same phenomenon also holds for convolutional

neural network trained as a ridge regressor (Sahraee-Ardakan et al., 2021), siamese neural networks Fonseca and Guidetti (2022), sparse neural networks (Chang et al., 2020).

Double descent also exists across different learning paradigms. This includes adversarial training (Rice et al., 2020; Singla et al., 2021; Hassani and Javanmard, 2022; Ribeiro and Schon, 2022), transfer learning (Dar and Baraniuk, 2020), knowledge distillation Saglietti and Zdeborov'a (2020); Cotter et al. (2021); Qu'etu and Tartaglione (2023b), and meta-learning Ju et al. (2023).

3.2 Double Descent Mitigation

Our work investigates conditions under which the peak of double descent does or does not occur. Related, various techniques have been demonstrated to successfully reduce the peak, including ℓ_2 regularization (Mei and Montanari, 2019; Nakkiran et al., 2020a,b; d'Ascoli et al., 2020b; Lin and Dobriban, 2020; Kan et al., 2020; Qu'etu and Tartaglione, 2023a,b), ensemble methods (Wilson and Izmailov, 2020; d'Ascoli et al., 2020b; Loureiro et al., 2022; Patil et al., 2022a), cross-validation (Patil et al., 2022b), dimensionality reduction (Huang et al., 2020b), input concatenation (Chen et al., 2021), and softplus random features (Dhifallah and Lu, 2020). We highlight that our focus in this thesis is not to propose a technique to mitigate double descent. Instead, we find that the strength of optimizer greatly affects the occurrence of the peaking phenomenon

in practice, and models with poor optimizers do not exhibit the peak. However, this observation might be useful as a simple technique to mitigate double descent in practice, which we leave to future research.

Albeit the universality of the double descent phenomenon, several work claim that double descent is not observed, even without explicit mitigation techniques mentioned above. These settings include self-adaptive training (Huang et al., 2020a, 2021), level of supervision (Dar et al., 2020; Luzi et al., 2021), random forest models (Buschjäger and Morik, 2021), two-layer neural network with certain initialization of the first layer weight (Ba et al., 2020), and special activation functions (Wang and Bento, 2022; Singla et al., 2021). Our results indicate that these negative results could occur because of poor optimization, conditions that lead to poor optimization, or loss functions which don't strongly exhibit the peaking phenomena.

While we do not know if our results contradict any of the results in the above settings, we emphasize the importance of carefully examining the effect of optimization in producing double descent. This is because optimization poses a generic influence on the peaking phenomenon, in almost all settings mentioned above. Therefore, it is crucial to identify whether the absence of double descent is due to a poor optimization procedure or other factors. For example, in Appendix B, we provide additional empirical evidence showing that softplus activation, which is shown to mitigate double descent in Dhifallah and Lu (2020), exhibits double descent as long as a better optimizer is used.

3.3 Unified Explanation Of Double Descent

Although there exist theoretical results explaining double descent curves in certain settings with particular assumptions (Deng et al., 2019; Kini and Thrampoulidis, 2020; Yang et al., 2020; Kuzborskij et al., 2021; Liu et al., 2021; Lee and Cherkassky, 2022; Singh et al., 2022; Ba et al., 2020; Gamba et al., 2022, 2023), we are not aware of any prior work that theoretically or experimentally presents a unified view of when double descent occurs or does not occur across various settings, especially realistic ones. The closest would be Deng et al. (2019) and Kini and Thrampoulidis (2020), which proved and empirically demonstrated that a max-margin classifier obtained from training logistic regression via GD exhibits monotonicity. Kuzborskij et al. (2021) is also relevant in that their derived upper bound on the generalization error captures the peaking behavior. Our work is a first step towards filling this gap by presenting a unified finding of when double descent occurs or doesn't occur across a wide range of settings.

Chapter 4

Experimental Setup

In this chapter, we briefly introduce the notations and the problem setup for a supervised classification setting. We leave a detailed formulation of the model, data, and learning algorithm in Appendix A. We consider a model $f(\cdot; \mathbf{W})$ and N i.i.d. training samples $\mathcal{D} = \{\mathbf{x}_i, y_i\}_{i=1}^N \sim p(X, Y)$, where the feature $\mathbf{x}_i \in \mathbb{R}^d$, and the learning procedure L is fixed. Depending on the setup, we apply normalization $\frac{\mathbf{X}-\mu}{s} \cdot \gamma$ to the feature matrix $\mathbf{X}$, where μ, s are the mean and standard deivation, and γ is a scaling factor. We consider both random feature models (Rahimi and Recht, 2007) and two-layer neural network, and initialize the first and second layer weights as $\mathbf{W}_0 \sim \mathcal{N}\left(0, k/\sqrt{D}\right)$ and $\mathbf{W}_1 \sim \mathcal{N}\left(0, 1/\sqrt{P}\right)$, respectively. k is used to control the standard deviation of the first-layer random matrix.

We primarily focus on the MSE loss and the cross-entropy loss, because they have

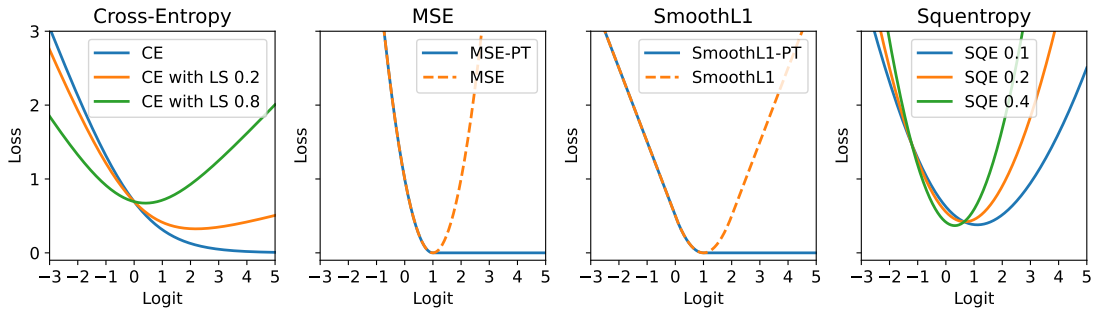


Figure 4.1: An illustration of the shape of cross-entropy (with varying label smoothing), MSE, smooth L1, squentropy for a binary classification problem, assuming the target class label is 1 and and the prediction is the x-axis. For the squentropy loss, it is infeasible to plot the MSE part as it only considers the incorrect labels, so we plot it by assuming the target class label is 0 for the incorrect class. For the same reason, only positive thresholded MSE and smooth L1 are shown.

been heavily studied in the literature (Belkin et al., 2018; Hastie et al., 2019; Mei and Montanari, 2019; Deng et al., 2019; Kini and Thrampoulidis, 2020; Emami et al., 2020; Dhifallah and Lu, 2020). We also examine some variants to ensure the robustness of our findings, including the ℓ_1 variants (i.e., smooth L1 loss), cross-entropy with label smoothing (Szegedy et al., 2016), and the newly proposed squentropy loss (Hui et al., 2023). For MSE loss, we follow the previous implementation and do not include the softmax at the network output Belkin et al. (2018); Hui and Belkin (2020). We modify the squentropy loss to include a weighting mechanism on the cross-entropy and MSE term, represented as $w \cdot \mathcal{L}_{\text{MSE}} + (1 - w) \cdot \mathcal{L}_{\text{CE}}$. Note that the MSE part is calculated over the incorrect classes only. We also employ a similar version that uses the original MSE objective (for all classes).

Thresholded MSE Loss. We introduce a thresholded version of the standard MSE loss, whose justification is given below in the next paragraph, with the form

$$\mathcal{L}_{\text{MSE-T}}(\hat{y}, y) = \sum_{n=1}^{N} \sum_{c=0}^{C-1} \mathbb{1}[\mathcal{C}] \cdot (\hat{y}_{i,c} - y_{i,c})^2.$$

Given the correct class c^* and incorrect class(es) $\bar{c}$, the $\mathbb{1}[\mathcal{C}]$ term is defined as an indicator function with conditions shown as follows:

$$\mathcal{C} = \begin{cases} \hat{y}_{i,c^*} > 1 & \text{positive threshold (MSE-PT)} \\ \hat{y}_{i,\bar{c}} < 1 & \text{negative threshold (MSE-NT)} \\ \hat{y}_{i,c^*} > 1 \vee \hat{y}_{i,\bar{c}} < 1 & \text{positive and negative threshold (MSE-PNT)} \end{cases}$$

When the predicted value $\hat{y}_{i,c}$ falls within the range of $[0, 1]$, the original MSE objective is retrieved. The abbreviations PT, NT, and PNT are used throughout the rest of the thesis to refer to the three variants. Note that removing the $\mathbb{1}[\mathcal{C}]$ recovers the original MSE loss. We show the shape of each loss function and their variants in Figure 4.1.

Obtaining or removing an exponential tail. We now provide justifications for our choice of loss functions. To make parabola-shaped loss functions almost-everywhere smooth, monotonically decreasing, and an exponential tail as in Soudry et al. (2018), we use a thresholding technique that sets loss value from certain classes to zero based on the predicted values to obtain an exponential tail. Specifically, with MSE-PT, setting the loss of the correct class prediction to 0 when the prediction is > 1 removes the right side

of the parabola, as shown in Figure 4.1. Note that MSE-PNT has a stronger exponential-property than MSE-PT and MSE-NT. This is because we are enforcing this property for all positive and negative classes. If only positive (or negative) thresholding is applied, only the loss calculated on the positive (or negative) classes will be exponential-tailed. It is worth noting that this thresholding technique is applicable to any regression loss function (including the ℓ_1 counterpart and the MSE term in the squentropy loss), as the modification relies solely on the predicted values and targets. Therefore, we adopt the smooth L1 loss and squentropy loss to test the effectiveness of the exponential tail. If enforcing the exponential tail eliminates double descent, does removing it bring back the peaking phenomenon? To test this hypotheis, we remove the exponential tail on cross-entropy by utilizing an existing technique – label smoothing (Szegedy et al., 2016). This modification reduces the model's confidence in its predictions, thus forming an ascent the right of the loss shape, as shown in the first subfigure of Figure 4.1.

Chapter 5

Experiments

Now we present empirical evidence from three aspects. First, we show that poor optimizers due to ill-conditioned feature matrices do not exhibit the peaking phenomenon. Second, we demonstrate that the same finding also holds for optimizers with slow convergence. Lastly, we examine the impact of the exponential tail in loss functions on the double descent peak. We present evidence on random feature models trained on MNIST (LeCun et al., 2010) in the main text of the thesis and include additional results for random feature models and two-layer neural networks trained on Fashion-MNIST (Xiao et al., 2017) and CIFAR-10 (Krizhevsky, 2009) in Appendix B.

Throughout this chapter, we make the following observations:

- **Observation 1**: The height of the double descent peak negatively correlates with the condition number of the feature matrix (i.e., the worse (higher) the condition

number, the higher the double descent peak), and the peak in the condition number always appears, regardless of whether double descet occurs.

- **Observation 2**: A poor optimization that results in slow convergence reduces or removes the peaking phenomenon; a better optimization restores the double descent peak.

- **Observation 3**: Loss functions with exponential tail almost never exhibit the peaking phenomenon, and giving a parabola-shape loss function the exponential tail property also eliminates the peak. We also study the convergence of exponential tail loss functions and see that the exponential-tail is what helps them converge to a max-margin solution.

5.1 Poor Optimizer with Ill-Conditioned Features

It is well-known from optimization theory that better conditioning leads to faster convergence for gradient-based optimizers. We now show that 1) the height of the double descent peak negatively correlates with the condition number of the feature matrix and 2) the presence of the peak in condition number does not necessarily predict double descent in generalization error, but it is the height of the peak that makes the difference. As shown in Figure 5.1, the condition number curves of the random features exhibit peaks at $P = N$, confirming the findings of previous studies (Poggio et al., 2019; Rangamani

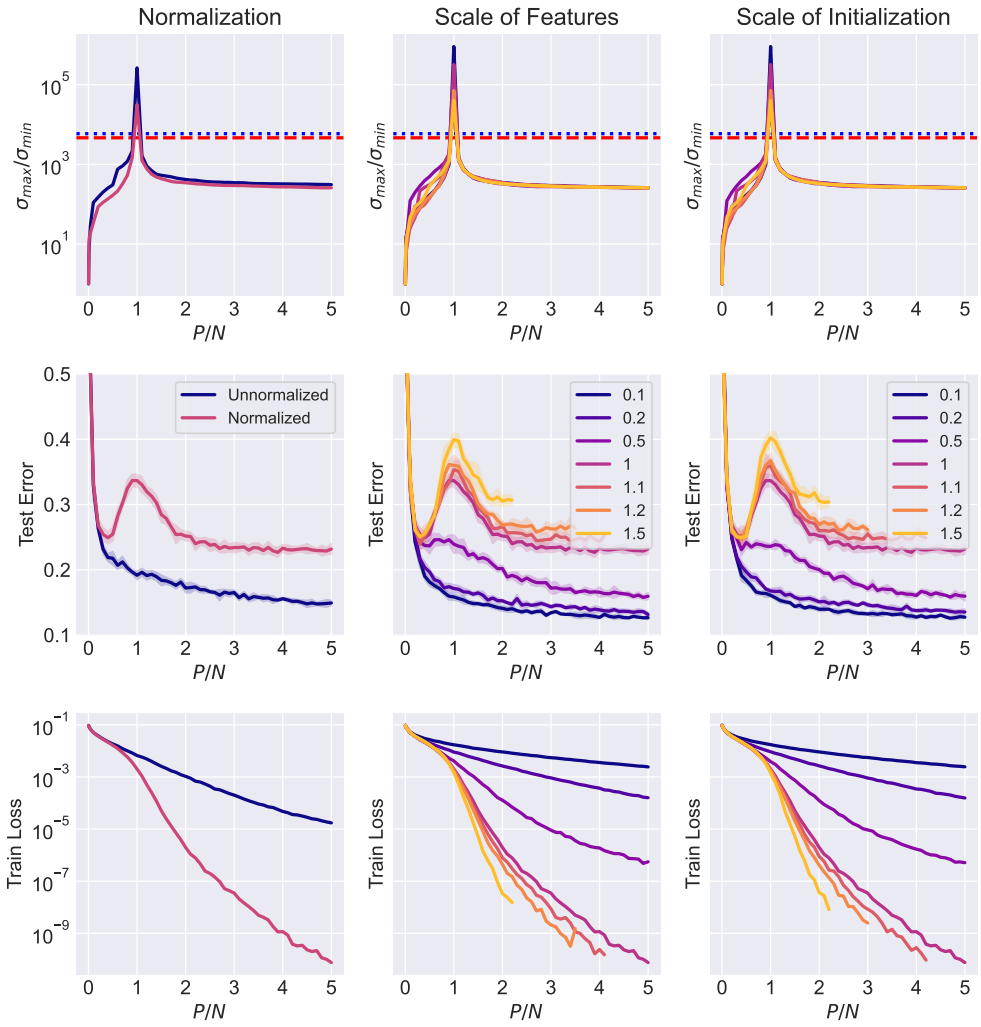


Figure 5.1: Condition number ($\sigma_{max}/\sigma_{min}$), test error, and training loss with and without normalization and varying the scale of features and the scale of initialization of the first-layer of a RFM with ReLU features and MSE loss. We find that double descent occurs when the initial setup has a lower condition number which allows for better optimization of the train loss. **Top**: The red and blue lines corresponds to the condition numbers of the original normalized and unnormalized features. Normalizing the data, increasing the scale of the features and random weights yield better condition numbers at $P = N$. **Middle**: Double descent does not occur in unnormalized, small scale of input, and small scale of initialization of weights. **Bottom**: A higher double descent peak corresponds to a lower training loss at $P = N$.

et al., 2020; Huang and Yang, 2020; Kuzborskij et al., 2021; Chen and Schaeffer, 2021). The dashed and dotted lines represent the condition numbers of the unprojected original features $\mathbf{X}'$ and $\mathbf{X}$, respectively. Notably, the condition number always peaks at $P = N$, as predicted by theoretical results. However, applying normalization and scaling makes them differ greatly at $P = N$.

We also see that the peak disappears when the feature matrix is unnormalized or when a small γ or k is used to scale $\mathbf{X}'$ and $\overline{\mathbf{W}}_0$. Note that these three factors have similar effect because, for random feature models, they all change the input features to the linear classifier (the last layer). Furthermore, we visualize the training loss and observe that the setting where double descent occurs has a training loss much smaller than one in which double descent does not occur. The ill-conditioned features makes optimization harder at $P = N$, which makes optimizers less likely to converge and causes the peak of double descent to be reduced or disappear. We can further infer that 1) a single condition number curve might not be sufficient to predict double descent, and 2) the curve-difference consistently indicates that double descent tends to occur in better conditioned matrices, with better optimization confirmed by the lower training loss.

While the peaking phenomenon does occur in condition numbers, there a two antagonistic forces at play when we observe the peak: the condition number grows, which make optimization harder and reduce the peak, but if we optimize better by improving the optimization setup by using a better optimizer or running for more steps,

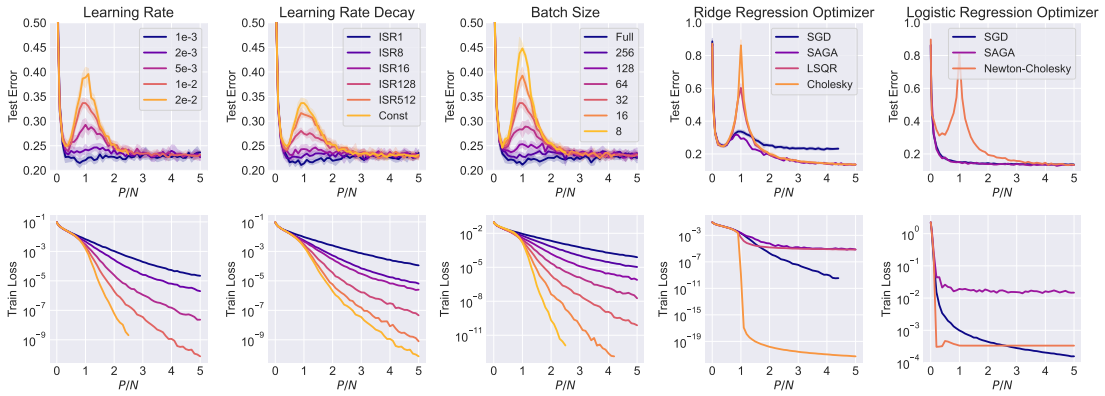


Figure 5.2: Test error and training loss curves by varying learning rate, batch size, and optimization algorithm. Models in the first four columns use MSE loss, and the last column use the logistic loss. Poor optimizers are in darker colors. Double descent occurs when a sufficiently large learning rate with infrequent decay, a small enough batch size, or a better optimizer are used. Similar to previous observations, a higher double descent peak typically corresponds to lower training loss.

we recover the peak (as shown in Figure 5.3).

5.2 Poor Optimizers with Slow Convergence

Our results suggest that poor optimization resulting in slow convergence often reduces or removes the peaking phenomenon, and better optimizers restore the peaking phenomenon. We initially observed this pattern in learning rate, batch size, and the optimization algorithm. However, all three factors can be considered as affecting the convergence rate of the optimizer on the provided loss function, where a slow convergence in loss corresponds to a less prominent peak.

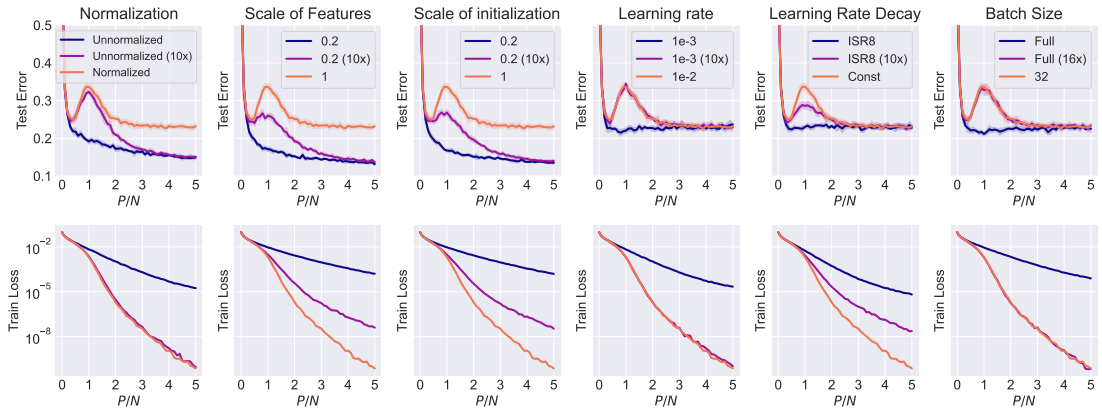


Figure 5.3: Classification error for a linear classifier trained with MSE loss. We report test error and training loss curves of poor optimizers, poor optimizers with 10x more iterations (labeled with "10x"), and better optimizers (in orange). The generalization error curves of poor optimizers approach that of the better optimzier case when the number of iterations is scaled by a factor of 10.

Optimization algorithm. We demonstrate that a better optimization algorithm that is able to find a lower low-loss minimum at $P = N$ induces a higher peak in generalization error. We select numerical solvers, Cholesky and QR (Paige and Saunders, 1982), for ridge regression, and Newton-Cholesky for logistic regression, because they obtain solutions with much lower loss than SGD. We also chose SAGA (Defazio et al., 2014) as an alternative gradient-based algorithm that (empirically) converges slower than SGD with momentum for comparison. A small regularization constant 1e-8 is used for numerical stability. In the last two columns of Figure 5.2, the height of the peak negatively correlates with the training loss at $P = N$. For curves with Cholesky decomposition, the error rate approaches random-guessing in both ridge regression and logistic regression.

Similar observation can be made on SAGA and SGD, where the double descent peak occurs mildly with a slight increase in test error.

Learning rate. We see that a higher peaks occurs when a large and constant learning rate is used. In the first column of Figure 5.2, when the overparameterization ratio $P/N > 3$, using a smaller learning rate has almost no impact on the test error, but it completely eliminates the peak without any form of explicit regularization. The same result holds for learning rate decay. In our experiments, we adopt an inverse square root schedule that is similar to the one used in Nakkiran et al. (2020a), but we multiply the initial learning rate by the factor $\frac{1}{\sqrt{\lfloor t/l \rfloor + 1}}$, where t is the current iteration and l is an interval parameter. By controlling the interval l, we modify the frequency of decay during the training trajectory. We observe that more frequent decay has a similar effect to a small constant learning rate, and decaying every iteration removes the peak entirely. Both imply that the emergence of double descent requires maintaining a stable and large enough learning rate during training, which corresponds to a better optimizer that lands at a lower loss minimum.

Batch size. We observe that double descent occurs when the batch size is small, or, equivalently, when the number of gradient updates increases. In Figure 5.2, for batch sizes of 500 (full-batch) and 256, the peaking phenomenon disappears on the generalization curve. It is worth noting that modifying the batch size changes the gradient

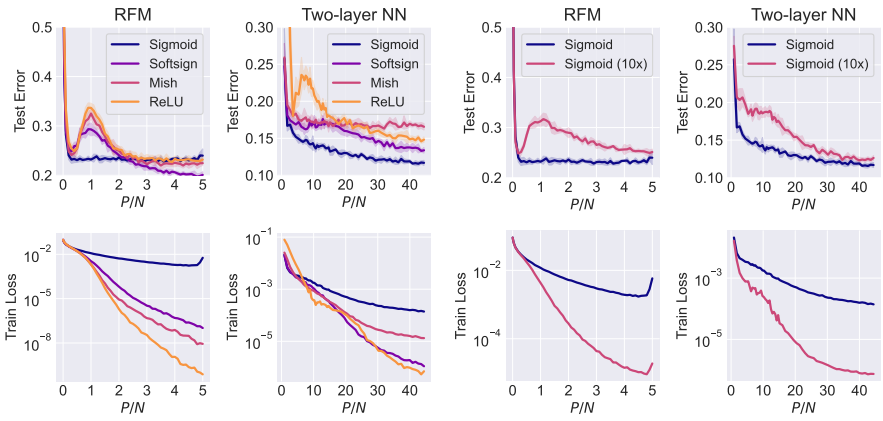


Figure 5.4: **Left**: Test error and training loss curves for RFMs trained with MSE loss and four different activation functions. Models with activation functions that converge a higher training loss tend to avoid double descent. The sigmoid activation does not exhibit double descent in both RFM and two-layer neural network given a fixed optimizer. **Right**: Test error and training loss curves for RFMs trained with MSE loss and sigmoid activation. By scaling the number of iterations by a factor of 10, double descent is recovered. This matches our findings on the impact of poor optimizers on double descent.

estimation, as a smaller batch size introduces more noise in the gradient, but with more gradient updates performed by the optimizer.

Activation functions. Previous work shows that activation functions reduce the peaking phenomenon (Dhifallah and Lu, 2020), but we show that, by simply increasing the number of iterations, we can recover the peak. We observe the same phenomenon on a wide variety of activation functions. However, for many of them, training becomes extremely unstable for larger P, so we show four representative ones below. In the left figure of Figure 5.4, we employ ReLU, mish (Misra, 2020), softsign, and sigmoid nonlinearities. Mish and softsign are used as variants of ReLU and sigmoid, respectively. For ReLU and sigmoid, they show consistent behavior for both RFM and two-layer neural

networks, even though activation functions operate differently on these two models (i.e., one with the input and the other with the embeddings). However, for mish and softsign, double descent is only observed in RFMs. In the right figure of Figure 5.4, we show that the absence of double descent on sigmoid follows the same pattern as our poor optimizer experiments. The sigmoid activation hinders the optimization and results in an illusion of monotonicity. We can recover the peaking phenomenon by scaling the number of iterations by a factor of 10.

Double descent occurs for better minimum. Lastly, we show that optimizers that converge to a lower training loss recover the peaking phenomenon. We increase the number of iterations by a factor of 10 so that the training loss of a poor optimizer setting is aligned with or approaches the default (a much lower one). For batch size, we make sure that the number of gradient updates matches that in a mini-batch case. Given a full-batch setting, we calculate the extended number of iterations by $\mathcal{T}_{\text{new}} = \lceil N/b \rceil \cdot \mathcal{T}$, which is equal to the number of gradient updates in a mini-batch setting with batch size b and $\mathcal{T}$ iterations. In Figure 5.3, we show that the double descent peak is recovered for all factors, with some of them exactly overlapped with curves produced by a better optimization setup (i.e., learning rate and batch size). This suggests that the strength of the optimization is a simple but strong indicator of why double descent is not observed in some realistic setting. In practical settings, even for critically parameterized models, people use regularization techniques or early stopping to obtain the best generalization

performance on the validation data. This process often prevents models reaching 0 training error. Even though no explicit regularization is applied, we usually do not continue training for a large number of epochs when the model has already overfit the training set. In our experiments, to achieve a better optimizer in Figure 5.3, models are usually trained 200-400 times longer after converging to 0 training error, which is not realistic for deep and large models used in practice. This also implies that, the rate of emergence of the double descent phenomenon largely depends on the specific optimization setup, where poor optimizers that converge slow do not exhibit the peaking behavior.

5.3 Loss Function With an Exponential-Tail Mitigates Double Descent

We now present evidence showing that 1) loss functions with exponential tail almost never exhibit double descent (even with a good optimizer) and 2) modifying parabola-shape losses (e.g., MSE and L1) to satisfy this property also entirely eliminates the peak. To avoid the illusion of poor optimization, we ensure for all loss functions, a good optimizer with large enough learning rate and small enough batch size is used. In column 1 Figure 5.5, we see that double descent does not occur when the regular cross-entropy loss is used. This observation is consistent with the results of Nakkiran et al. (2020a)

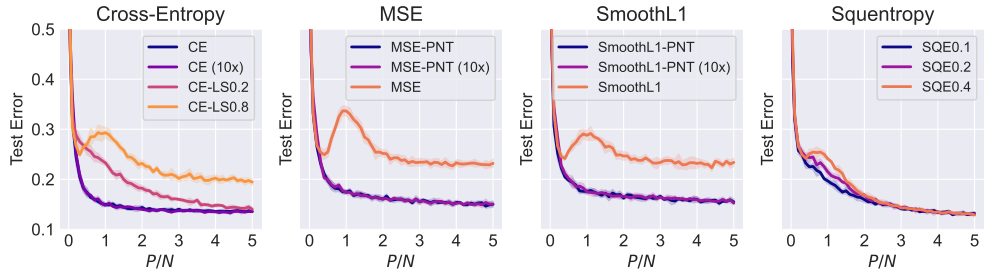


Figure 5.5: Test error curves for different types of loss functions. Double descent does not occur in loss functions with an exponential tail (CE, MSE-PNT, SmoothL1-PNT), and starts to emerge when the exponential tail is removed (CE with label smoothing and squentropy). Unlike poor optimizers, for those with an exponential tail, they robustly show a monotonic decreasing test error.

and Yang et al. (2020), where the double descent peak becomes much less pronounced or disappears when no or less label noise is introduced for cross-entropy loss. However, by incorporating label smoothing, the peak starts to emerge. For regression losses like MSE and L1 in their original form, a prominent peak is shown. To give MSE and L1 the exponential-tail property, we adapt the thresholding technique introduced in Chapter 4 by masking losses larger than 1 for the correct class and losses smaller than 0 for the incorrect classes. Even after increasing the number of iterations by a factor of 10, the monotonicity for cross-entropy, MSE, and L1 losses all hold.

Interestingly, we can increase the height of the peak by removing the exponential tail. In column 4, by reweighting the CE and MSE terms in squentropy, as we gradually increase the weight for the MSE loss and decreasing the weight for cross-entropy. Note that by multiply the MSE loss by a constant essentially changes the shape of the parabola, and a larger constant makes the parabola shape steeper, which is similar to removing

the exponential-tail property. Therefore, a larger weight on MSE corresponds to a more pronounced peak. Similarly, by increasing the smooth constant for cross-entropy, we relax the exponential-tail property by penalizing confidence predictions, which makes the curve go up on the right and creates a parabola-like shape shown in Figure 4.1.

5.4 Exponential-Tail Loss Functions Converge to Max-Margin Solutions

In this section, we experimentally verify that loss functions with an exponential tail tend to converge to max-margin solution in our setup. This is theoretically predicted in Soudry et al. (2018). Thus we experimentally verify that the exponential-tail loss functions that do not exhibit double descent also converge to a max-margin hyperplane. These loss functions are not limited to the logistic loss (as proved in Soudry et al. (2018)) and also include variants of the thresholded MSE loss introduced in Chapter 4. We employ the synthetic dataset in (Soudry et al., 2018), which is used to show a logistic regression model trained via SGD converges to a max-margin solution during training. Following their setting, we use four support vectors $\mathbf{x}_1 = (0.5, 1.5), \mathbf{x}_2 = (1.5, 0.5), \mathbf{x}_3 = -\mathbf{x}_1, \mathbf{x}_4 = -\mathbf{x}_2$ and assign positive label to $\mathbf{x}_1, \mathbf{x}_2$ and negative label to $\mathbf{x}_3, \mathbf{x}_4$. We randomly generated 8 other data points for each class and train the a linear or logistic regression with SGD using learning rate $\eta = 1/\sigma^2_{\max}(\mathbf{X})$ and without

momentum.

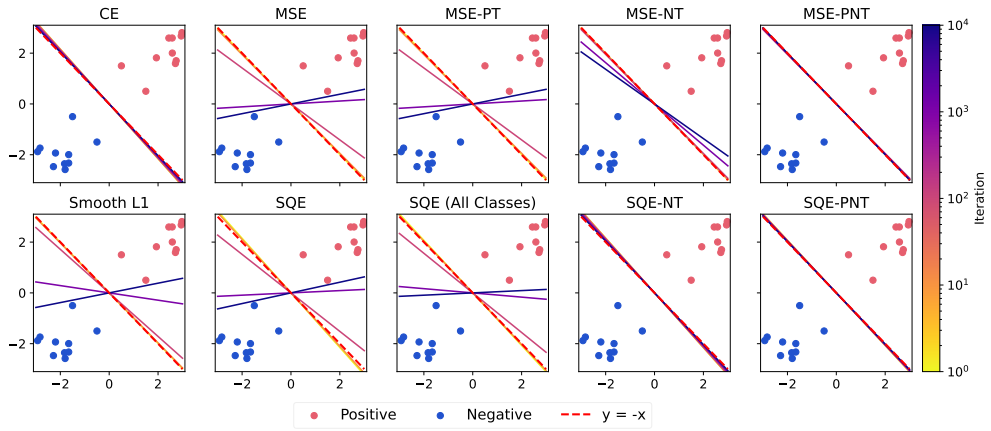


Figure 5.6: A synthetic dataset with four support vectors and 16 other randomly points generated based on the correponding (correct) class region. The known max-margin separating hyperplane is the red dashed line. CE, MSE-PNT, SQE-NT, and SQE-PNT all have exponential tails for all classes, and are exactly the loss functions that we find do not exhibit double descent (see Figure 5.5 and Appendix B.2). MSE-PT and MSE-NT only has an exponential tail for the positive and negative classes, respectively. Solutions at the 1st, 10th, 100th, 1000th, and the 10000th iteration are shown for each loss function. The double-descent-resistant (exponential-tailed) loss functions all converge to the max-margin solution, whereas others diverge.

In Figure 5.6, for each loss function, we show five solutions during training, and the last solution is marked by the darkest color. For CE and MSE-PNT, both of them converge to the max-margin hyperplane at the last iteration. SQE-NT and SQE-PNT are variants of the squentropy loss (Hui et al., 2023). SQE-NT uses the original implementation by including only the incorrect classes in its MSE term, but also thresholds its negative predictions smaller than 0. SQE-PNT uses MSE-PNT as its MSE term. Both SQE variants also converge to the optimal hyperplane. For these losses, whether its convergence is a max-margin separating hyperplane seems to be consistent

with if double descent occurs (see Appendix B.2). Notably, by examing the convergence of MSE-PT, MSE-NT, and MSE-PNT, we see that MSE-PT suffers from overfitting in the same way as the regular MSE, whereas it is less severe for MSE-NT. Interestingly, it also aligns with the peaking behavior, where MSE-PT still suffer from the peak, and MSE-NT only shows a tiny peak after a large number of iterations (see Figures B.8 and B.10). This also suggests that the degree of thresholding is related to the strength of the exponential tail, which matters in terms of double descent and generalization.

5.5 Growth of Feature Separation With Respect to Feature Dimension

We experimentally verify the result discussed in Section 2.1 that the separation of the feature mean $\|\mu\|^4$ grows faster than the feature dimension P under some normalization and scaling conditions. As we observe in Section 5.1, it is exactly these conditions that result in ill-conditioned features and are prone to double descent under non-exponential-tail losses. In Theorem 4. of Chatterji and Long (2020), the authors used a data model $x = Uq + \tilde{y}\mu$, where U is a unitary matrix with the center of feature cluster being $\mu \in \mathbb{R}^p$. They assume, for a large constant C, 1) the failure probability is $0 \leq \delta \leq 1/C$, 2) we have $n \geq C\log(1/\delta)$ samples, 3) each with $p \geq C\max\{\|\mu\|^2 n, n^2\log(n/\delta)\}$ dimension, and 4) $\|\mu\|^2 \geq C\log(n/\delta)$. The authors prove that, with sub-Gaussian data

and the above assumptions and $c > 0$, the misclassification error for a max-margin classifier $\mathbf{W}$ for feature $\mathbf{x}$ and target y in the separable case trained with logistic loss has the following form

$$\mathbb{P}_{(x,y)\sim \mathrm{P}}[\mathrm{sign}(\mathbf{W} \cdot \mathbf{x}) \neq y] \leq \eta + \exp\left(-c\frac{\|\mu\|^4}{P}\right),$$

with probability $1 - \delta$. In other words, for the misclassification error to be lower, the $\frac{\|\mu\|^4}{P}$ term has to grow with P, and needs to satisfy $\|\mu\|^4 = \omega(P)$ for the error to decrease. In the case of random features, the feature dimension P increases, so knowing how faster the class separation $\|\mu\|^4$ grows will tell us how the bound changes. We empirically measure the quantity $\frac{\|\mu\|^4}{P}$ for both normalized and unnormalized features, and the largest and the smallest γ and k, similar to Figure 5.1, on MNIST dataset between digit 0 and all other classes. In Figure 5.7, we observe that $\|\mu\|^4$ grows faster than P when the features are normalized, or use a large scale γ or a large initialization scale k. While this result is empirically verified, further research on why it applies to these conditions is needed. Interestingly, as we discuss in Section 2.1 and observe in Section 5.1, it is exactly these conditions that lead to ill-conditioned features and are prone to double descent under non-exponential-tail losses.

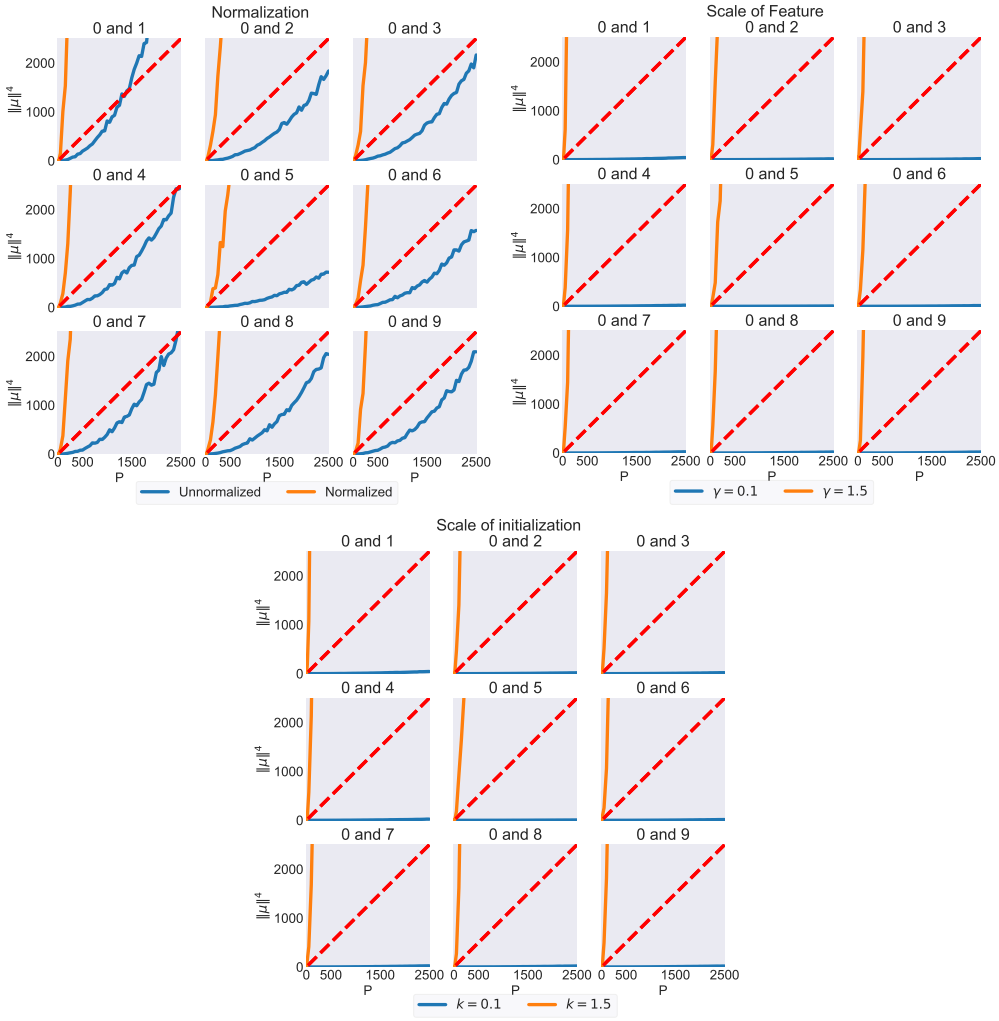


Figure 5.7: The feature seperation distance on MNIST dataset for class 0 and all other classes with 1) unnormalized and normalized features, 2) a small and a large scale of features, and 3) a small and a large scale of initialization for the random matrix in RFM.

Chapter 6

Conclusion

In this thesis, we investigate why the double descent peaking phenomenon sometimes does not occur for overfit models in practice. Our findings highlights two key factors: the use of poor optimizers and the presence of an exponential tail in the loss function shape. We see poor optimizers that struggles to find low-loss local minima despite zero training error, can be double-descent-free. By employing inductive bias and hyperparameters with faster convergence or a longer training time, we successfully recover the peaking behavior. We also observe that loss functions with an exponential tail are found to resist the double descent, even in extreme overfitting scenarios. Modifying loss functions to incorporate an exponential tail restored monotonicity in the error curve, while removing it reinstated double descent. Based on the results, we believe that classifiers in practice is unlikely to suffer severely from the peaking phenomenon. On one hand, inductive biases

are usually chosen meticulously using a validation set to prevent overfitting. Explicit regularization techniques also serve as another means of addressing this issue. On the other hand, in practical classification tasks, MSE loss is rarely used, and the widely adopted cross-entropy loss already possesses the advantageous property for monotonicity. The careful selection of inductive biases and hyperparameters, along with the prevalent usage of exponential-tail loss functions, contributes to a "natural" mitigation of double descent. Based on our results, we believe that classifiers in practice is unlikely to suffer severely from the peaking phenomenon. Additionally, taken together, our results point towards areas of further theoretical study.

Future work While we have included convincing results on how poor optimization and exponential-tail loss functions reduce the double descent peak, providing rigorous theoretical justifications would be a next step to consolidating our findings. Specifically, one direction is to derive bounds similar to Kuzborskij et al. (2021) that capture the peaking phenomenon and, at the same time, account for different elements that affect the strength of the optimization, with minimal assumptions. Another direction is to study the exponential-tail loss functions and why they reduce the peak in the critically-parameterized regime. To our knowledge, proofs in existing work focusing on the max-margin convergence of logistic-loss-like functions (Soudry et al., 2018; Ji and Telgarsky, 2019; Nacson et al., 2019) are model-size agnostic. This potentially suggests why the error rate can decrease for all model sizes around the interpolation threshold,

which visually reduces the whole peak, but a formal proof based on the parameterization ratio would be ideal. Other interesting direction include studying the rate of emergence of the double descent peak. As we observe in Figure 5.3, a poor optimization setting requires approximately 10 times more iterations to exhibit double descent. A characterization of how fast the peaking phenomenon occurs would also be useful to identify the double descent behavior and avoid it in practice.

Appendix A

Experimental Setup and Hyperparameters

In this chapter, we described the detailed setup of dataset, models, and training.

A.1 Datasets

We perform all experiments on MNIST (LeCun et al., 2010), Fashion-MNIST (Xiao et al., 2017), and CIFAR-10 (Krizhevsky, 2009). We select small subsets of size N from the full training set as our training data and evaluate the generalization error using the complete test set. We trained models for RFMs with a random feature size of up to $P/N = 5$, while for two-layer NNs, we utilize models up to $P/N = 5 \cdot C$ parameters, where C is fixed at 10 in our setting. For two-layer NNs, the interpolation threshold for

the MSE loss is positioned at $N \cdot C$ instead of N, as depicted in Belkin et al. (2018). By default, we normalize the image pixel features to follow a normal distribution by applying the transformation $\frac{\mathbf{X}-\mu}{s} \cdot \gamma$, where μ, s, γ are the mean, standard deviation, and the scaling factor, respectively.

A.2 Models

Random feature models. Here we give the experimental details of the random feature models. For random feature models, we use a static first-layer weights $\overline{\mathbf{W}}_0 \in \mathbb{R}^{d \times P}$ and trainable second-layer weights $\mathbf{W}_1 \in \mathbb{R}^{P \times C}$, where P and C represent the projected feature dimension and the number of classes, respectively. We initialize both weight matrices from $\overline{\mathbf{W}}_0 \sim \mathcal{N}\left(0, \frac{k_0}{\sqrt{D}}\right)$ and $\mathbf{W}_1 \sim \mathcal{N}\left(0, \frac{k_1}{\sqrt{P}}\right)$, where k_0 is a scaling factor for the standard deviation of the weight matrix and D represents the input feature dimension. k_1 is always 1 in all experiments. Bias terms are always set to 0, and the ReLU nonlinearity is used by default.

Two-layer neural networks Our setup for two-layer neural networks resembles that for a random feature model. The only difference is that the first layer weights are trained.

A.3 Training

We now describe the training setup and how to produce double descent. Consistent with Belkin et al. (2018) and Nakkiran et al. (2020a), we employ the same set of hyperparameters for all model sizes and trained them using SGD with a fixed number of epochs and constant step size. The default hyperparameters are selected based on two training error constraints: 1) the largest model has to attain 0 training error within the first $1/10$ iterations, and 2) (at least) all models with $P \geq N$ has to converge to 0 training error before the final iteration. These constraints, derived from empirical observations, are found to be fast in convergence and effective in generating the double descent phenomenon for both RFM and two-layer NN trained with SGD on MNIST and Fashion-MNIST. After some exploration, we use SGD with Nesterov momentum 0.95, a mini-batch size of 32, and a constant step size of 1e-2 for 1000 epochs. For two-layer NNs, we increase the step size to 5e-2 and the number of epochs to 1500. By default, we utilize the standard MSE loss. When a specific hyperparameter or loss function is studied, all other parameters follow the default ones. All experiments are repeated at least five times.

Appendix B

Additional Experiments

Here we present the same figures as in the main text of the thesis, but on additional datasets (Fashion-MNIST and CIFAR-10) and two-layer neural networks. We show that our findings in the main text of the thesis generalize to these datasets and models.

B.1 Poor Optimization

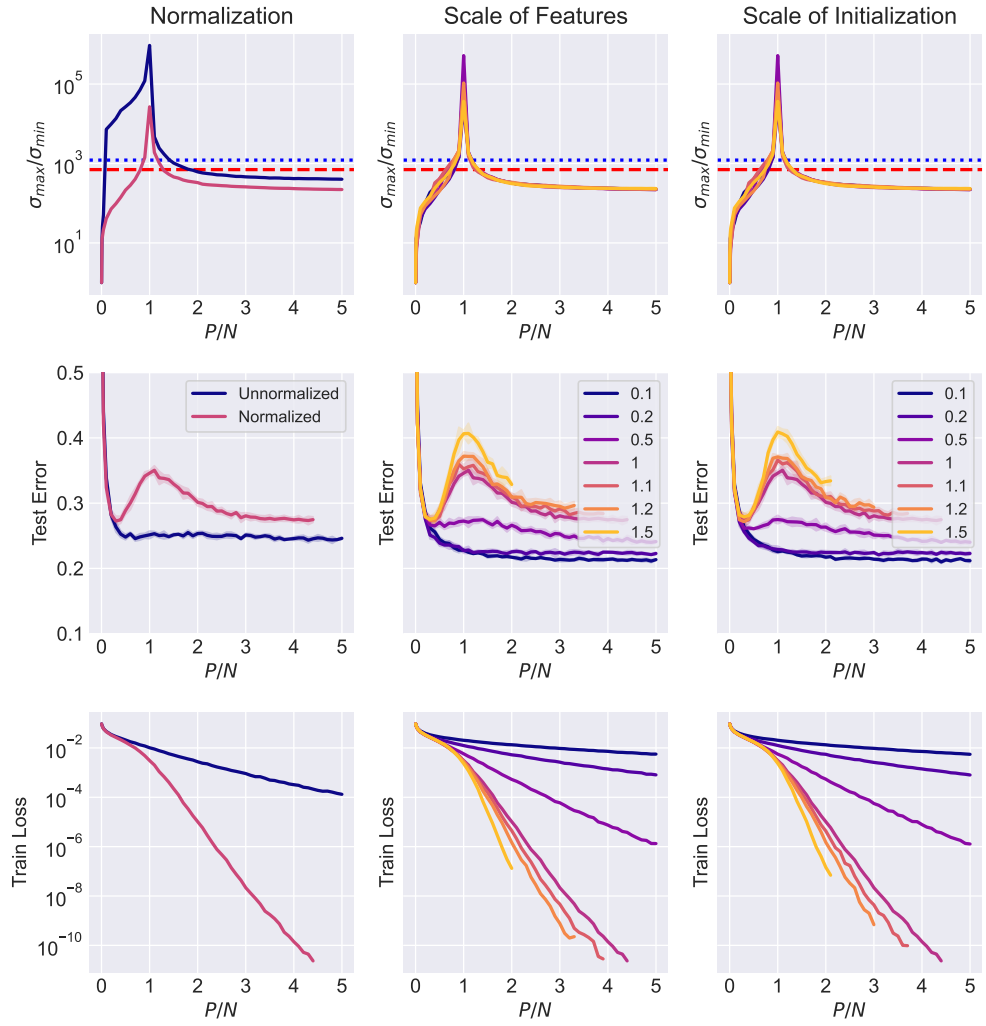


Figure B.1: Additional results to support Figure 5.1. Condition number ($\sigma_{\max}/\sigma_{\min}$), test error, and training loss of RFMs 1) with and without normalization, varying 2) the scale of features and 3) the scale of initialization of the first layer random feature model on Fashion-MNIST and CIFAR-10. The peaking phenomenon becomes less prominent as the features becomes worse-conditioned, with which the training loss is higher.

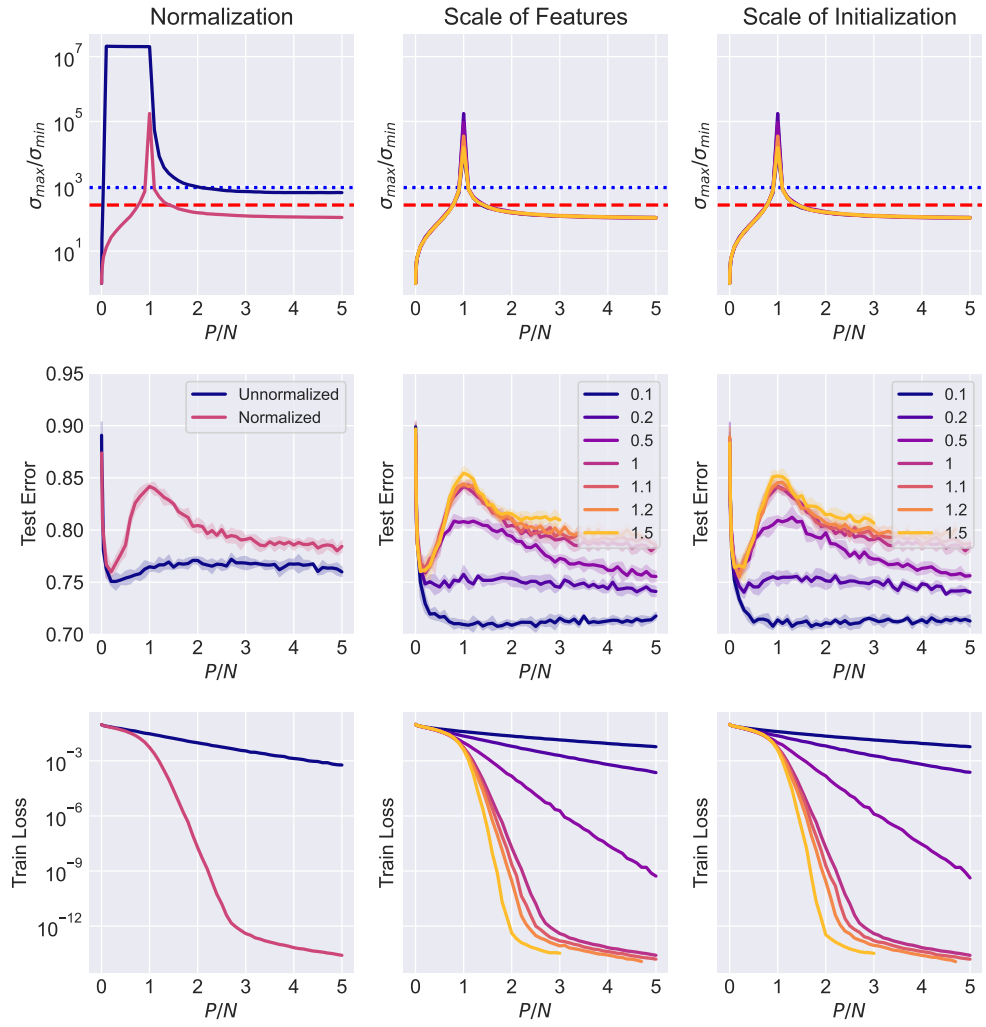
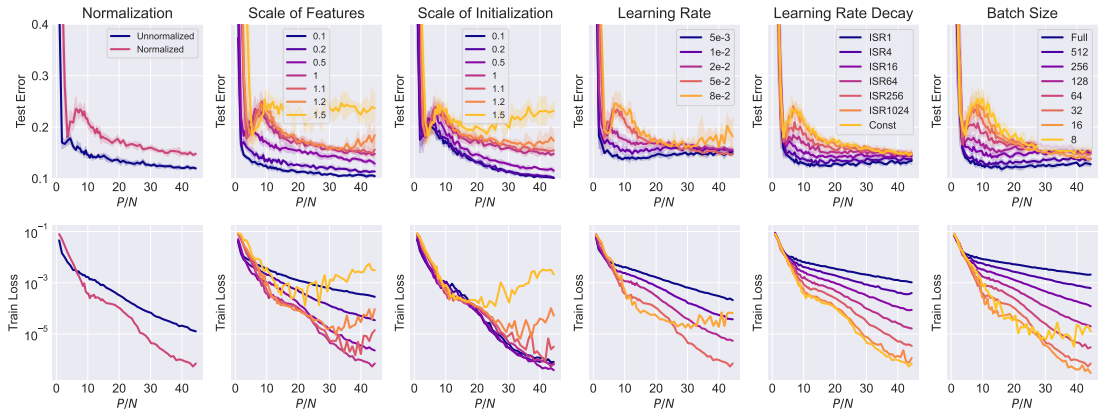
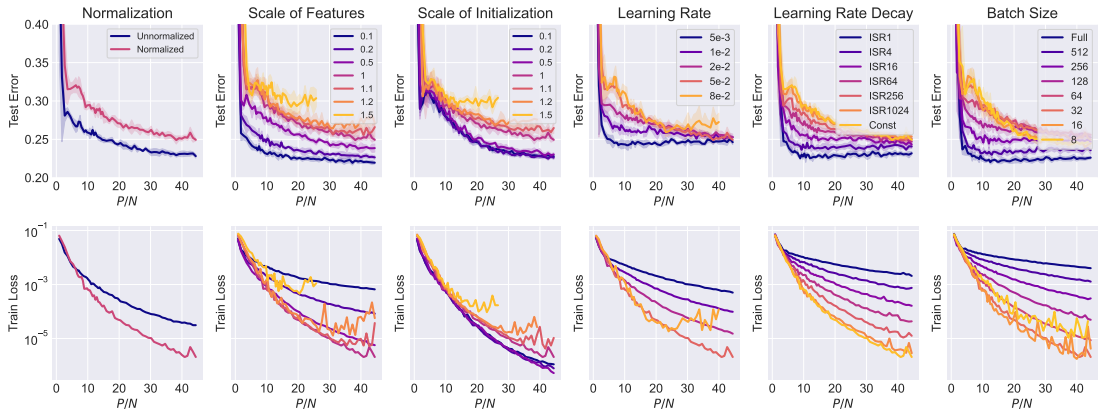


Figure B.2: Additional results to support Figure 5.1. Condition number ($\sigma_{\max}/\sigma_{\min}$), test error, and training loss of two-layer neural networks 1) with and without normalization, varying 2) the scale of features and 3) the scale of initialization of the first layer random feature model on Fashion-MNIST and CIFAR-10. The peaking phenomenon becomes less prominent as the features becomes worse-conditioned, with which the training loss is higher.

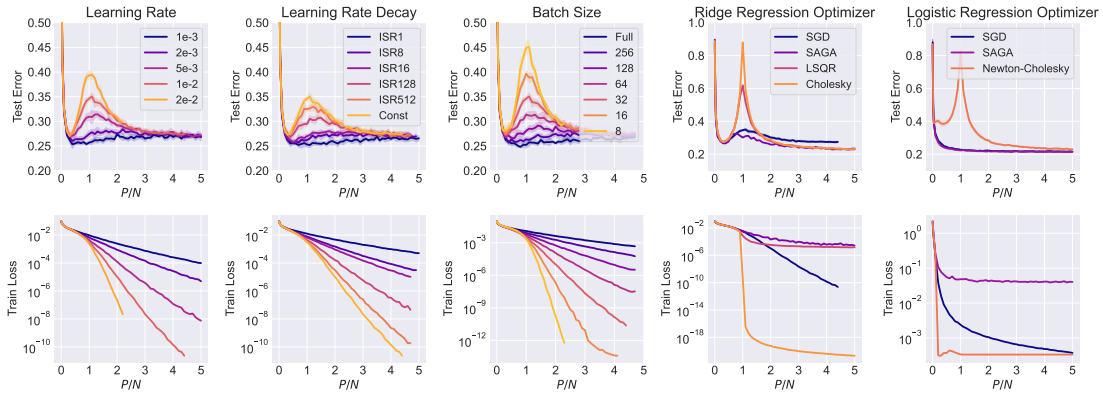


(a) Two-layer neural network on MNIST

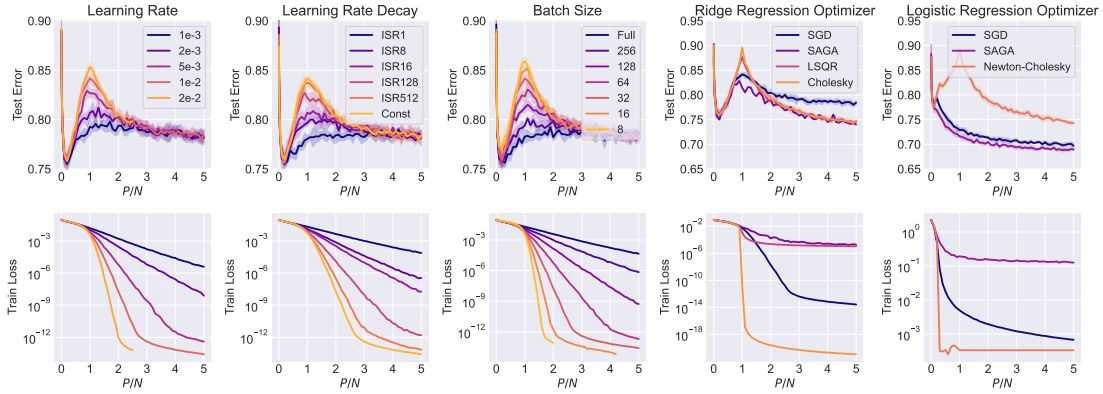


(b) Two-layer neural network on Fashion-MNIST

Figure B.3: Additional results to support Figures 5.1 and 5.2. Test error and training loss curves by (using) normalization, varying scale of features or initialization, learning rate, batch size, and optimization algorithm of a two-layer neural network on Fashion-MNIST and CIFAR-10. The peaking phenomenon becomes less prominent as the features becomes worse-conditioned or directly using a poor optimizer.

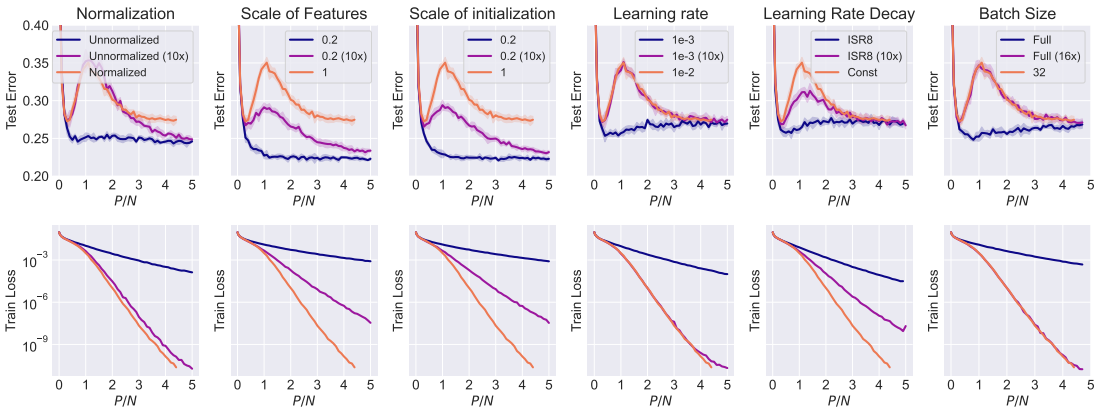


(a) RFM on Fashion-MNIST

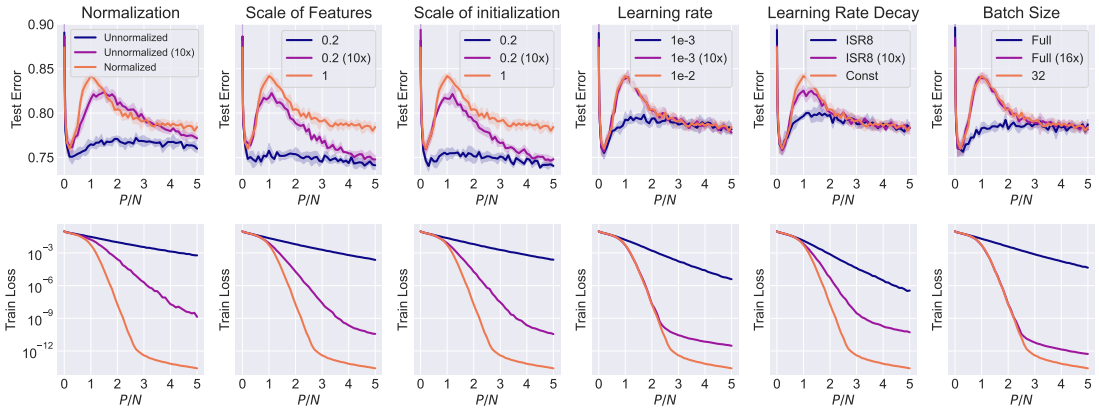


(b) RFM on CIFAR-10

Figure B.4: Additional results to support Figure 5.2. Test error and training loss curves by varying learning rate, batch size, and optimization algorithm of a random feature model on Fashion-MNIST and CIFAR-10. The peaks are reduced on poor optimizers with too small learning rate, too frequent learning rate decay, too large batch size, and poor optimization algorithms.

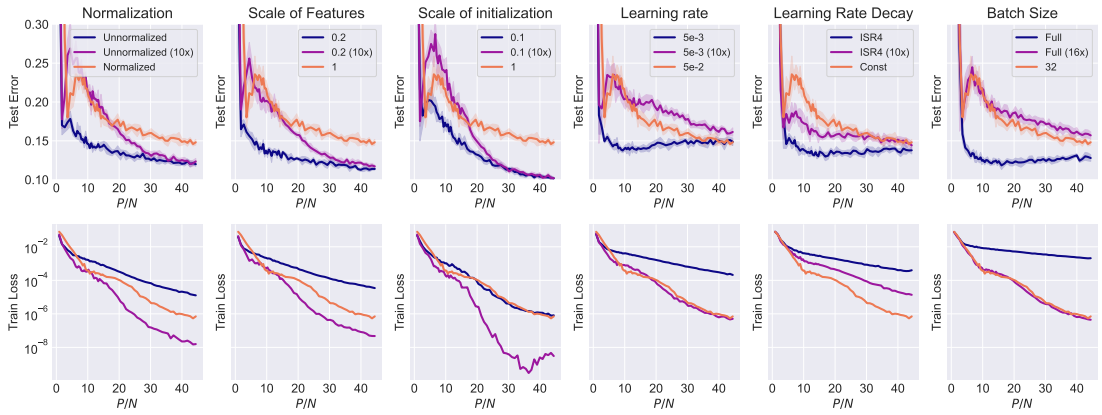


(a) RFM on Fashion-MNIST

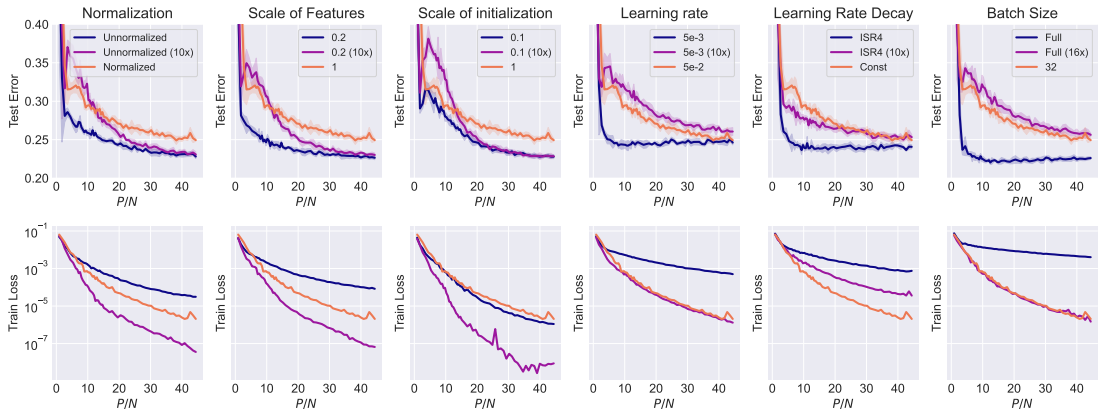


(b) RFM on CIFAR-10

Figure B.5: Additional results to support Figure 5.3. Test error and training loss curves of poor optimizers with 10x iterations of RFMs on Fashion-MNIST and CIFAR-10. We are able to recover the peaking phenomenon in all cases by scaling the number of iterations by a factor of 10.



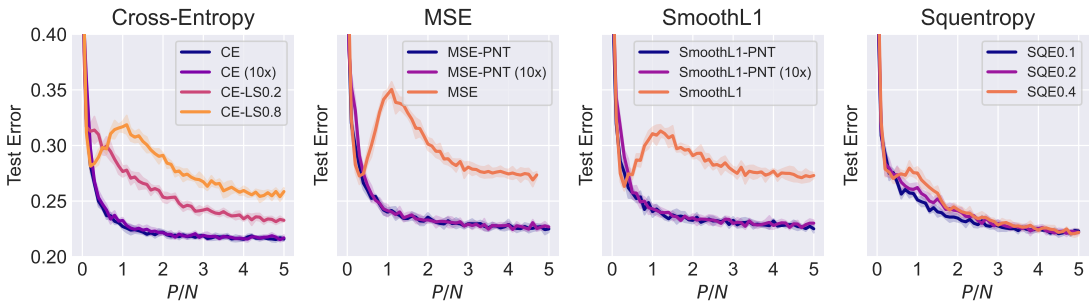
(a) Two-layer neural network on MNIST



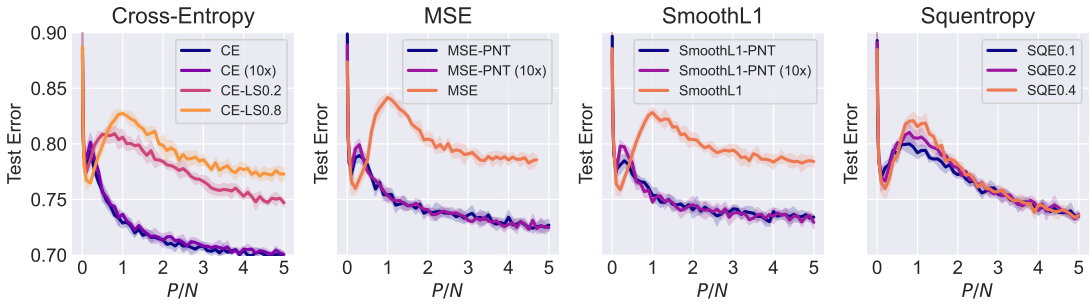
(b) Two-layer neural network on Fashion-MNIST

Figure B.6: Additional results to support Figure 5.3. Test error and training loss curves of poor optimizers with 10x iterations of two-layer neural networks on Fashion-MNIST and CIFAR-10. We are able to recover the peaking phenomenon in all cases by scaling the number of iterations by a factor of 10.

B.2 Loss Functions

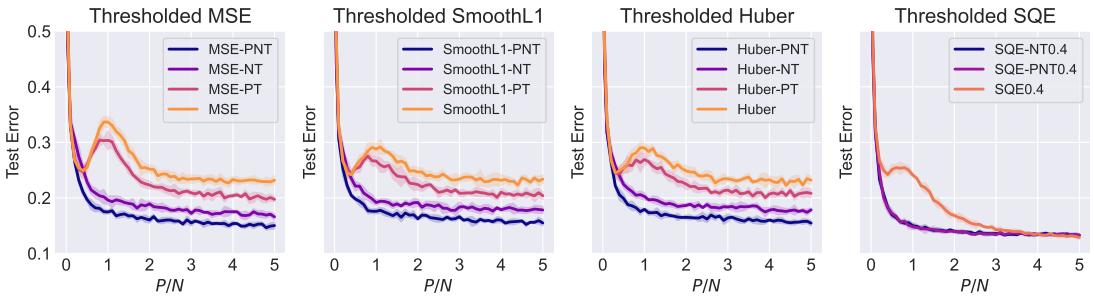


(a) RFM on Fashion-MNIST

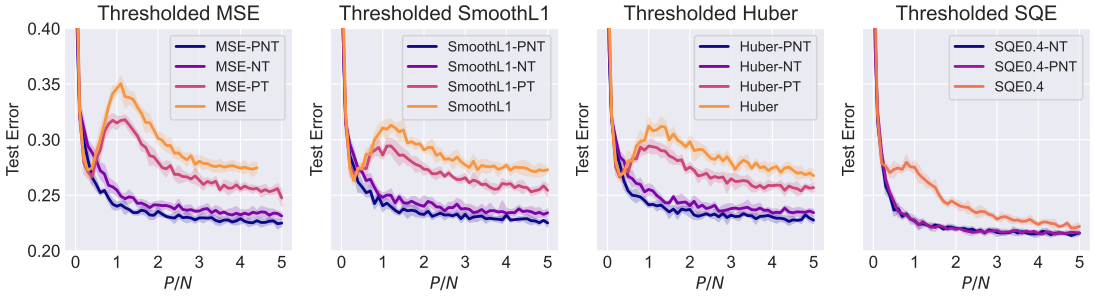


(b) RFM on CIFAR-10

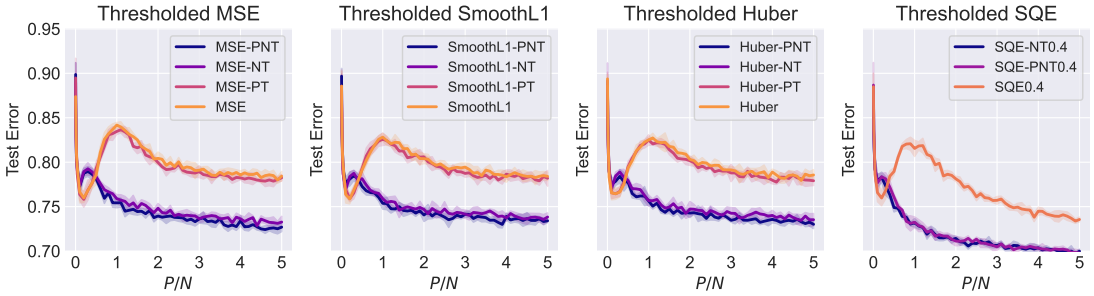
Figure B.7: Additional results to support Figure 5.5. Test error curves for RFMs trained with loss functions with exponential-tail for extended iterations on Fashion-MNIST and CIFAR-10. Loss functions with exponential tail do not exhibit the peak or only show a slight peak.



(a) RFM on MNIST

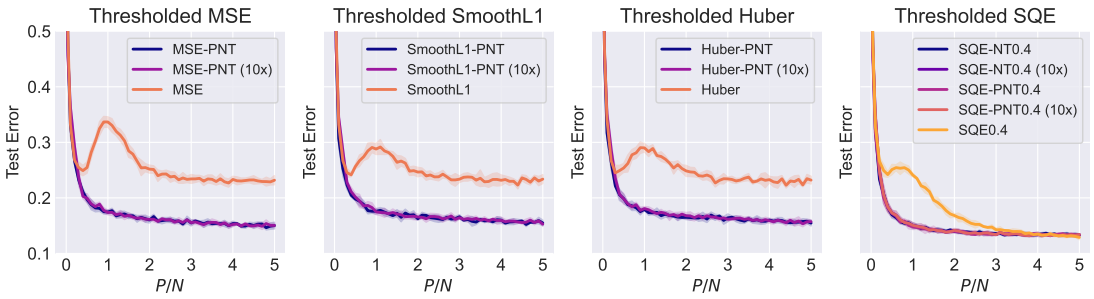


(b) RFM on Fashion-MNIST

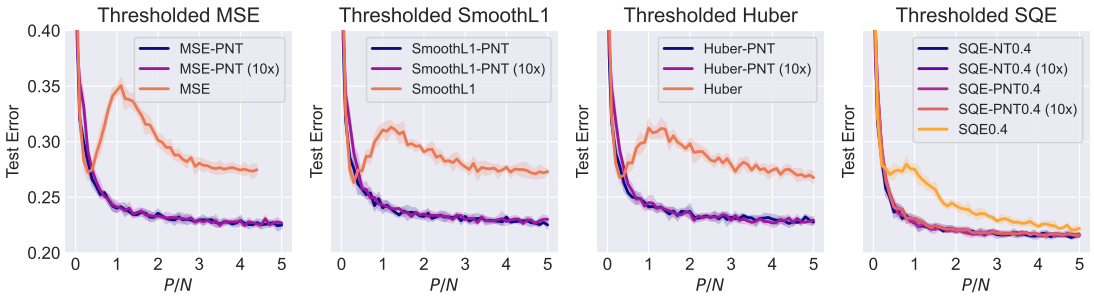


(c) RFM on CIFAR-10

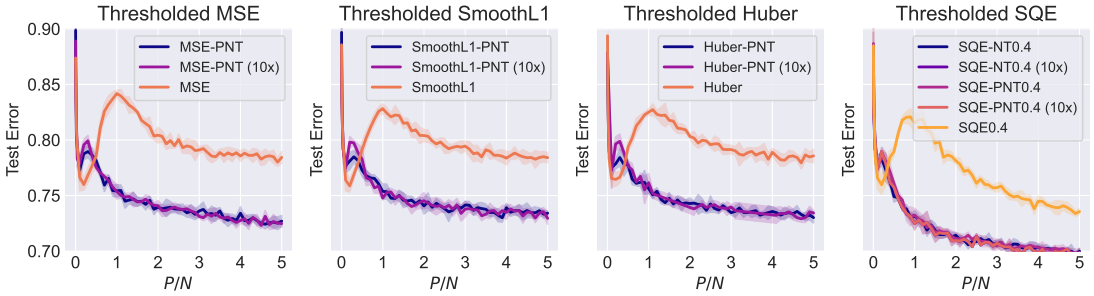
Figure B.8: Additional results to support Figure 5.5. Test error curves for RFMs trained with loss functions and different thresholding directions on MNIST, Fashion-MNIST, and CIFAR-10. Double descent is mitigated in losses with negative threshold or positive and negative threshold (i.e., MSE, smooth L1, huber. For squentropy, monotnicity is restored when the MSE term has an exponential tail.



(a) RFM on MNIST

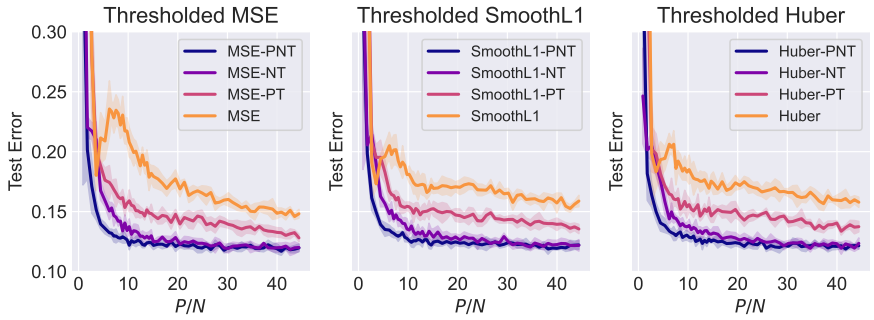


(b) RFM on Fashion-MNIST

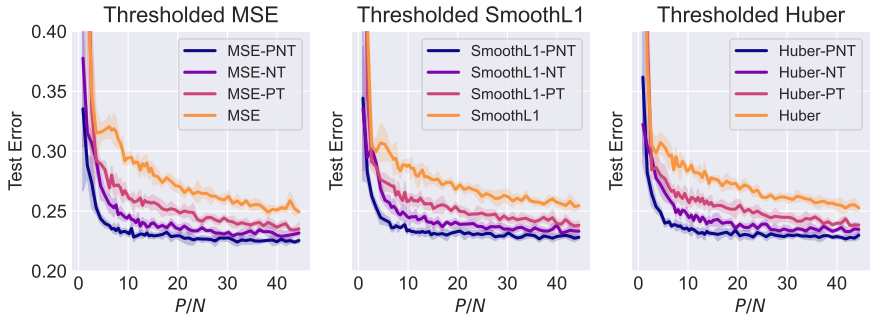


(c) RFM on CIFAR-10

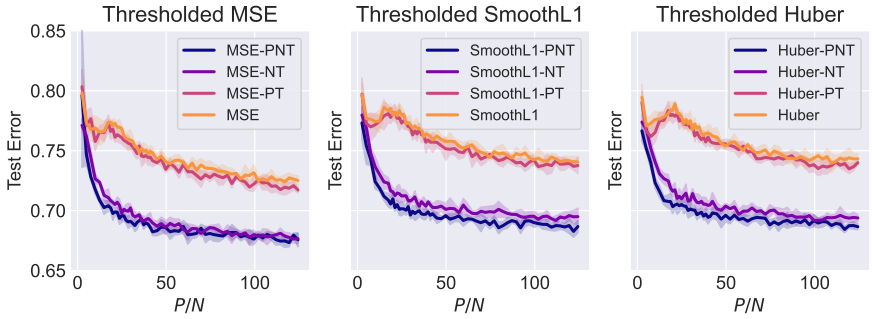
Figure B.9: Additional results to support Figure 5.5. Test error curves for RFMs trained with loss functions with exponential-tail for extended iterations on MNIST, Fashion-MNIST, and CIFAR-10. The monotonicity is robust under the exponential-tail property, even with a better optimzier with 10x iterations.



(a) Two-layer neural network on MNIST

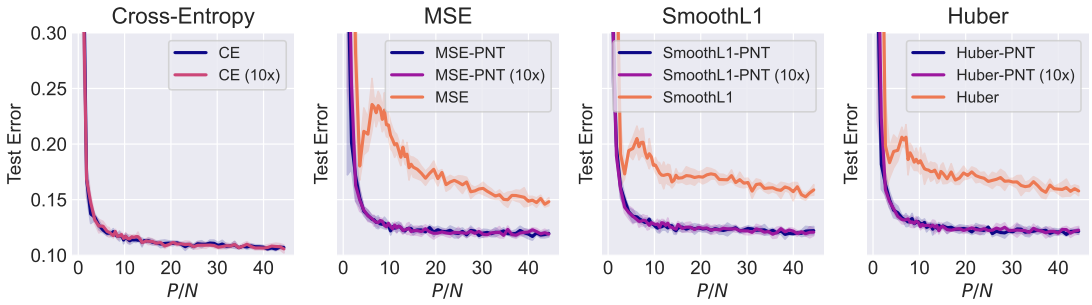


(b) Two-layer neural network on Fashion-MNIST

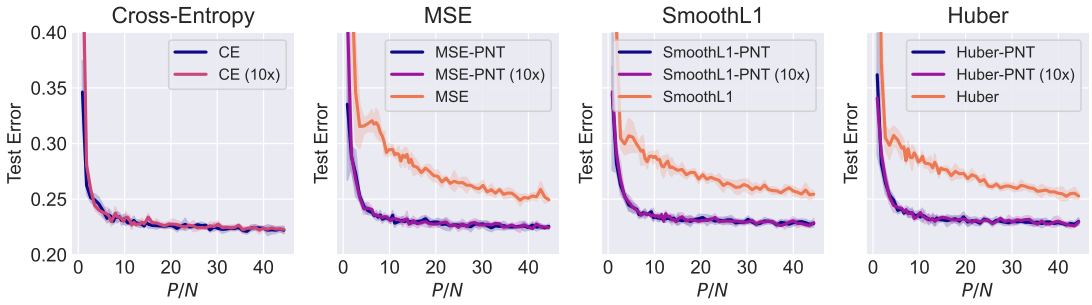


(c) Two-layer neural network on CIFAR-10

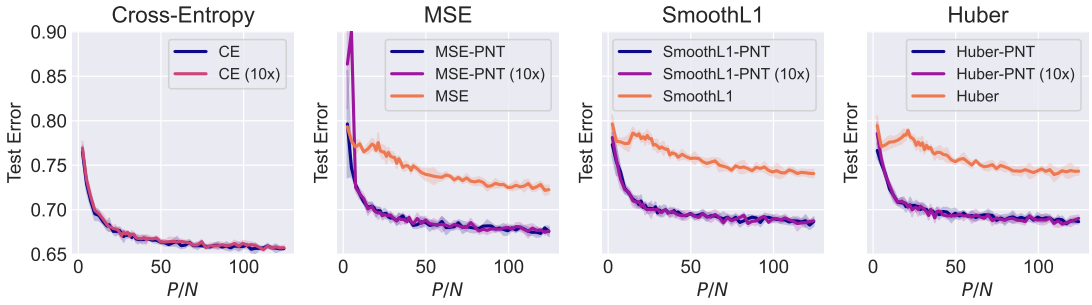
Figure B.10: Additional results to support Figure 5.5. Test error curves for two-layer neural networks trained with loss functions and different thresholding directions on MNIST, Fashion-MNIST, and CIFAR-10. Thresholds in the negative or the positive and negative directions are required to mitigate the peak.



(a) Two-layer neural network on MNIST



(b) Two-layer neural network on Fashion-MNIST



(c) Two-layer neural network on CIFAR-10

Figure B.11: Additional results to support Figure 5.5. Test error curves for two-layer neural networks trained with loss functions with exponential-tail for extended iterations on MNIST, Fashion-MNIST, and CIFAR-10. Loss functions with exponential tail do not exhibit the peak, even with a better optimzier with 10x iterations.

Bibliography

Ba, J., Erdogdu, M. A., Suzuki, T., Wu, D., and Zhang, T. (2020). Generalization of two-layer neural networks: An asymptotic viewpoint. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net.

Bachmann, G., Hofmann, T., and Lucchi, A. (2022). Generalization through the lens of leave-one-out error. *ArXiv preprint*, abs/2203.03443.

Belkin, M., Hsu, D. J., Ma, S., and Mandal, S. (2018). Reconciling modern machine-learning practice and the classical bias-variance trade-off. *Proceedings of the National Academy of Sciences*, 116:15849–15854.

Belkin, M., Hsu, D. J., and Xu, J. (2019). Two models of double descent for weak features. *SIAM J. Math. Data Sci.*, 2:1167–1180.

Buschjäger, S. and Morik, K. (2021). There is no double-descent in random forests. *ArXiv preprint*, abs/2111.04409.

Chang, X., Li, Y., Oymak, S., and Thrampoulidis, C. (2020). Provable benefits of overparameterization in model compression: From double descent to pruning neural networks. In *AAAI Conference on Artificial Intelligence*.

Chatterji, N. S. and Long, P. M. (2020). Finite-sample analysis of interpolating linear classifiers in the overparameterized regime. *ArXiv preprint*, abs/2004.12019.

Chatterji, N. S., Long, P. M., and Bartlett, P. L. (2020). When does gradient descent with logistic loss find interpolating two-layer networks? *J. Mach. Learn. Res.*, 22:159:1–159:48.

Chatterji, N. S., Long, P. M., and Bartlett, P. L. (2021). When does gradient descent with logistic loss interpolate using deep networks with smoothed relu activations? In *Annual Conference Computational Learning Theory*.

Chen, J., Wang, Q., and Kyrillidis, A. (2021). Mitigating deep double descent by concatenating inputs. *Proceedings of the 30th ACM International Conference on Information & Knowledge Management*.

Chen, Z. and Schaeffer, H. (2021). Conditioning of random feature matrices: Double descent and generalization error. *ArXiv preprint*, abs/2110.11477.

Cotter, A., Menon, A. K., Narasimhan, H., Rawat, A. S., Reddi, S. J., and Zhou, Y. (2021). Distilling double descent. *ArXiv preprint*, abs/2102.06849.

Dar, Y. and Baraniuk, R. (2020). Double double descent: On generalization errors in transfer learning between linear regression tasks. *ArXiv preprint*, abs/2006.07002.

Dar, Y., Mayer, P., Luzi, L., and Baraniuk, R. G. (2020). Subspace fitting meets regression: The effects of supervision and orthonormality constraints on double descent of generalization errors. In *Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 13-18 July 2020, Virtual Event*, volume 119 of *Proceedings of Machine Learning Research*, pages 2366–2375. PMLR.

d'Ascoli, S., Gabri'e, M., Sagun, L., and Biroli, G. (2021). On the interplay between data structure and loss function in classification problems. In *Neural Information Processing Systems*.

d'Ascoli, S., Refinetti, M., Biroli, G., and Krzakala, F. (2020a). Double trouble in double descent : Bias and variance(s) in the lazy regime. In *International Conference on Machine Learning*.

d'Ascoli, S., Sagun, L., and Biroli, G. (2020b). Triple descent and the two kinds of overfitting: where and why do they appear? *Journal of Statistical Mechanics: Theory and Experiment*, 2021.

Defazio, A., Bach, F. R., and Lacoste-Julien, S. (2014). SAGA: A fast incremental gradient method with support for non-strongly convex composite objectives. In Ghahramani, Z., Welling, M., Cortes, C., Lawrence, N. D., and Weinberger, K. Q., editors, *Advances in Neural Information Processing Systems 27: Annual Conference on Neural Information Processing Systems 2014, December 8-13 2014, Montreal, Quebec, Canada*, pages 1646–1654.

Deng, Z., Kammoun, A., and Thrampoulidis, C. (2019). A model of double descent for high-dimensional binary linear classification. *ArXiv preprint*, abs/1911.05822.

Dhifallah, O. and Lu, Y. M. (2020). A precise performance analysis of learning with random features. *ArXiv preprint*, abs/2008.11904.

Duin, R. P. W. (2000). Classifiers in almost empty spaces. *Proceedings 15th International Conference on Pattern Recognition. ICPR-2000*, 2:1–7 vol.2.

Emami, M., Sahraee-Ardakan, M., Pandit, P., Rangan, S., and Fletcher, A. K. (2020). Generalization error of generalized linear models in high dimensions. In *Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 13-18 July 2020, Virtual Event*, volume 119 of *Proceedings of Machine Learning Research*, pages 2892–2901. PMLR.

Fonseca, N. and Guidetti, V. (2022). Similarity and generalization: From noise to corruption. *ArXiv preprint*, abs/2201.12803.

Gamba, M., Azizpour, H., and Bjorkman, M. (2023). On the lipschitz constant of deep networks and double descent. *ArXiv preprint*, abs/2301.12309.

Gamba, M., Englesson, E., Bjorkman, M., and Azizpour, H. (2022). Deep double descent via smooth interpolation. *ArXiv preprint*, abs/2209.10080.

Gerace, F., Loureiro, B., Krzakala, F., Mézard, M., and Zdeborová, L. (2020). Generalisation error in learning with random features and the hidden manifold model. In *Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 13-18 July 2020, Virtual Event*, volume 119 of *Proceedings of Machine Learning Research*, pages 3452–3462. PMLR.

Gupta, A., Mishra, R., Luu, W., and Bouassami, M. (2023). On feature scaling of recursive feature machines. *ArXiv preprint*, abs/2303.15745.

Hassani, H. and Javanmard, A. (2022). The curse of overparametrization in adversarial training: Precise analysis of robust generalization for random features regression. *ArXiv preprint*, abs/2201.05149.

Hastie, T. J., Montanari, A., Rosset, S., and Tibshirani, R. J. (2019). Surprises in high-dimensional ridgeless least squares interpolation. *Annals of statistics*, 50 2:949–986.

Hodgkinson, L., van der Heide, C., Roosta, F., and Mahoney, M. W. (2022). Monotonicity and double descent in uncertainty estimation with gaussian processes. *ArXiv preprint*, abs/2210.07612.

Holzmüller, D. (2020). On the universality of the double descent peak in ridgeless regression.

Huang, H. and Yang, Q. (2020). Large scale analysis of generalization error in learning using margin based classification methods. *Journal of Statistical Mechanics: Theory and Experiment*, 2020.

Huang, L., Zhang, C., and Zhang, H. (2020a). Self-adaptive training: beyond empirical risk minimization. In Larochelle, H., Ranzato, M., Hadsell, R., Balcan, M., and Lin, H., editors, *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*.

Huang, L., Zhang, C., and Zhang, H. (2021). Self-adaptive training: Bridging the supervised and self-supervised learning. *IEEE transactions on pattern analysis and machine intelligence*, PP.

Huang, N. T., Hogg, D. W., and Villar, S. (2020b). Dimensionality reduction, regularization, and generalization in overparameterized regressions. *SIAM J. Math. Data Sci.*, 4:126–152.

Hui, L. and Belkin, M. (2020). Evaluation of neural architectures trained with square loss vs cross-entropy in classification tasks. *ArXiv preprint*, abs/2006.07322.

Hui, L., Belkin, M., and Wright, S. (2023). Cut your losses with squentropy. *ArXiv preprint*, abs/2302.03952.

Ji, Z. and Telgarsky, M. (2019). The implicit bias of gradient descent on nonseparable data. In Beygelzimer, A. and Hsu, D., editors, *Conference on Learning Theory, COLT 2019, 25-28 June 2019, Phoenix, AZ, USA*, volume 99 of *Proceedings of Machine Learning Research*, pages 1772–1798. PMLR.

Ju, P., Liang, Y., and Shroff, N. B. (2023). Theoretical characterization of the generalization performance of overfitted meta-learning. *ArXiv preprint*, abs/2304.04312.

Kan, K. K., Nagy, J. G., and Ruthotto, L. (2020). Avoiding the double descent phenomenon of random feature models using hybrid regularization. *ArXiv preprint*, abs/2012.06667.

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

Krizhevsky, A. (2009). Learning multiple layers of features from tiny images. Technical report.

Kuzborskij, I., Szepesvari, C., Rivasplata, O., Rannen-Triki, A., and Pascanu, R. (2021). On the role of optimization in double descent: A least squares study. In *Neural Information Processing Systems*.

Kwon, O.-R. and Zou, H. (2023). Multivariate regression via enhanced response envelope: Envelope regularization and double descent.

LeCun, Y., Cortes, C., and Burges, C. (2010). Mnist handwritten digit database. *ATT Labs [Online]. Available: http://yann.lecun.com/exdb/mnist*, 2.

Lee, E. H. and Cherkassky, V. (2022). Vc theoretical explanation of double descent. *ArXiv preprint*, abs/2205.15549.

Lin, L. and Dobriban, E. (2020). What causes the test error? going beyond bias-variance via anova. *J. Mach. Learn. Res.*, 22:155:1–155:82.

Liu, F., Suykens, J. A. K., and Cevher, V. (2021). On the double descent of random features models trained with sgd. *ArXiv preprint*, abs/2110.06910.

Loog, M., Viering, T. J., Mey, A., Krijthe, J. H., and Tax, D. M. J. (2020). A brief prehistory of double descent. *Proceedings of the National Academy of Sciences*, 117:10625–10626.

Loureiro, B., Gerbelot, C., Refinetti, M., Sicuro, G., and Krzakala, F. (2022). Fluctuations, bias, variance & ensemble of learners: Exact asymptotics for convex losses in high-dimension. *ArXiv preprint*, abs/2201.13383.

Luzi, L., Dar, Y., and Baraniuk, R. (2021). Double descent and other interpolation phenomena in gans. *ArXiv preprint*, abs/2106.04003.

Lyu, K., Li, Z., Wang, R., and Arora, S. (2021). Gradient descent on two-layer nets: Margin maximization and simplicity bias. In *Neural Information Processing Systems*.

Maddox, W. J., Benton, G. W., and Wilson, A. G. (2020). Rethinking parameter counting in deep models: Effective dimensionality revisited. *ArXiv preprint*, abs/2003.02139.

Mei, S. and Montanari, A. (2019). The generalization error of random features regression: Precise asymptotics and the double descent curve. *Communications on Pure and Applied Mathematics*, 75.

Misra, D. (2020). Mish: A self regularized non-monotonic activation function. In *British Machine Vision Conference*.

Nacson, M. S., Srebro, N., and Soudry, D. (2019). Stochastic gradient descent on separable data: Exact convergence with a fixed learning rate. In Chaudhuri, K. and Sugiyama, M., editors, *The 22nd International Conference on Artificial Intelligence and Statistics, AISTATS 2019, 16-18 April 2019, Naha, Okinawa, Japan*, volume 89 of *Proceedings of Machine Learning Research*, pages 3051–3059. PMLR.

Nakkiran, P., Kaplun, G., Bansal, Y., Yang, T., Barak, B., and Sutskever, I. (2020a). Deep double descent: Where bigger models and more data hurt. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net.

Nakkiran, P., Venkat, P., Kakade, S. M., and Ma, T. (2020b). Optimal regularization can mitigate double descent. *ArXiv preprint*, abs/2003.01897.

Opper, M., Kinzel, W., Kleinz, J., and Nehl, R. (1990). On the ability of the optimal perceptron to generalise. *Journal of Physics A*, 23.

Paige, C. C. and Saunders, M. A. (1982). Lsqr: An algorithm for sparse linear equations and sparse least squares. *ACM Trans. Math. Softw.*, 8:43–71.

Patil, P. V., Du, J.-H., and Kuchibhotla, A. K. (2022a). Bagging in overparameterized learning: Risk characterization and risk monotonization.

Patil, P. V., Kuchibhotla, A. K., Wei, Y., and Rinaldo, A. (2022b). Mitigating multiple descents: A model-agnostic framework for risk monotonization. *ArXiv preprint*, abs/2205.12937.

Poggio, T. A., Kur, G., and Banburski, A. (2019). Double descent in the condition number. *ArXiv preprint*, abs/1912.06190.

Qu'etu, V. and Tartaglione, E. (2023a). Can we avoid double descent in deep neural networks?

Qu'etu, V. and Tartaglione, E. (2023b). Dodging the sparse double descent. *ArXiv preprint*, abs/2303.01213.

Rahimi, A. and Recht, B. (2007). Random features for large-scale kernel machines. In Platt, J. C., Koller, D., Singer, Y., and Roweis, S. T., editors, *Advances in Neural Information Processing Systems 20, Proceedings of the Twenty-First Annual Conference on Neural Information Processing Systems, Vancouver, British Columbia, Canada, December 3-6, 2007*, pages 1177–1184. Curran Associates, Inc.

Rangamani, A., Rosasco, L., and Poggio, T. A. (2020). For interpolating kernel machines, minimizing the norm of the erm solution maximizes stability. *Analysis and Applications*.

Ribeiro, A. H., Hendriks, J. N., Wills, A. G., and Schön, T. B. (2020). Beyond occam's razor in system identification: Double-descent when modeling dynamics. *ArXiv preprint*, abs/2012.06341.

Ribeiro, A. H. and Schon, T. (2022). Overparameterized linear regression under adversarial attacks. *IEEE Transactions on Signal Processing*, 71:601–614.

Rice, L., Wong, E., and Kolter, J. Z. (2020). Overfitting in adversarially robust deep learning. In *Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 13-18 July 2020, Virtual Event*, volume 119 of *Proceedings of Machine Learning Research*, pages 8093–8104. PMLR.

Sa-Couto, L., Ramos, J. M., Almeida, M., and Wichert, A. M. (2022). Understanding the double descent curve in machine learning. *ArXiv preprint*, abs/2211.10322.

Saglietti, L. and Zdeborov'a, L. (2020). Solvable model for inheriting the regularization through knowledge distillation. *ArXiv preprint*, abs/2012.00194.

Sahraee-Ardakan, M., Mai, T., Rao, A. B., Rossi, R. A., Rangan, S., and Fletcher, A. K. (2021). Asymptotics of ridge regression in convolutional models. In *International Conference on Machine Learning*.

Schaeffer, R., Khona, M., Robertson, Z., Boopathy, A., Pistunova, K., Rocks, J. W., Fiete, I. R., and Koyejo, O. (2023). Double descent demystified: Identifying, interpreting & ablating the sources of a deep learning puzzle. *ArXiv preprint*, abs/2303.14151.

Singh, S. P., Lucchi, A., Hofmann, T., and Scholkopf, B. (2022). Phenomenology of double descent in finite-width neural networks. *ArXiv preprint*, abs/2203.07337.

Singla, V., Singla, S., Jacobs, D., and Feizi, S. (2021). Low curvature activations reduce overfitting in adversarial training. *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 16403–16413.

Soudry, D., Hoffer, E., Nacson, M. S., and Srebro, N. (2018). The implicit bias of gradient descent on separable data. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*. OpenReview.net.

Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., and Wojna, Z. (2016). Rethinking the inception architecture for computer vision. In *2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016*, pages 2818–2826. IEEE Computer Society.

Vallet, F., Cailton, J., and Réfrégier, P. (1989). Linear and nonlinear extension of the pseudo-inverse solution for learning boolean functions. *EPL*, 9:315–320.

Wang, J. and Bento, J. (2022). Optimal activation functions for the random features regression model. *ArXiv preprint*, abs/2206.01332.

Wilson, A. G. and Izmailov, P. (2020). Bayesian deep learning and a probabilistic perspective of generalization. *ArXiv preprint*, abs/2002.08791.

Wu, D. and Xu, J. (2020). On the optimal weighted ℓ_2 regularization in overparameterized linear regression. *ArXiv preprint*, abs/2006.05800.

Xiao, H., Rasul, K., and Vollgraf, R. (2017). Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms. *ArXiv preprint*, abs/1708.07747.

Xing, Y., Song, Q., and Cheng, G. (2019). Benefit of interpolation in nearest neighbor algorithms. *ArXiv preprint*, abs/1909.11720.

Xu, J. and Hsu, D. J. (2019). On the number of variables to use in principal component regression. In Wallach, H. M., Larochelle, H., Beygelzimer, A., d'Alché-Buc, F., Fox, E. B., and Garnett, R., editors, *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, pages 5095–5104.

Xu, T., Zhou, Y., Ji, K., and Liang, Y. (2018). When will gradient methods converge to max-margin classifier under relu models? *Stat*, 10.

Yang, Z., Yu, Y., You, C., Steinhardt, J., and Ma, Y. (2020). Rethinking bias-variance trade-off for generalization of neural networks. *ArXiv preprint*, abs/2002.11328.

Zavatone-Veth, J. A., Tong, W., and Pehlevan, C. (2022). Contrasting random and learned features in deep bayesian linear regression. *Physical review. E*, 105 6-1:064118.