Regularization-wise double descent: Why it occurs and how to eliminate it

Fatih Furkan Yilmaz[*] and Reinhard Heckel[*,†]

[*]Dept. of Electrical and Computer Engineering, Rice University
[†]Dept. of Electrical and Computer Engineering, Technical University of Munich

Abstract

The risk of overparameterized models, in particular deep neural networks, is often double-descent shaped as a function of the model size. Recently, it was shown that the risk as a function of the early-stopping time can also be double-descent shaped, and this behavior can be explained as a super-position of bias-variance tradeoffs. In this paper, we show that the risk of explicit L2-regularized models can exhibit double descent behavior as a function of the regularization strength, both in theory and practice. We find that for linear regression, a double descent shaped risk is caused by a superposition of bias-variance tradeoffs corresponding to different parts of the model and can be mitigated by scaling the regularization strength of each part appropriately. Motivated by this result, we study a two-layer neural network and show that double descent can be eliminated by adjusting the regularization strengths for the first and second layer. Lastly, we study a 5-layer CNN and ResNet-18 trained on CIFAR-10 with label noise, and CIFAR-100 without label noise, and demonstrate that all exhibit double descent behavior as a function of the regularization strength.

1 Introduction

The bias-variance tradeoff has long been a useful principle for selecting and tuning machine learning models. This principle suggests to choose a model sufficiently large to have low bias, but not too large to have small variance. In practice, however, machine learning models seemingly operate beyond this tradeoff. Deep neural networks operate in the overparameterized regime where the model is capable of expressing any given signal, even random noise [Zha+17], but still generalize well. Increasing the model size beyond the interpolation point often decreases the test error beyond the classical U-shaped curve, hence forming a double descent shaped risk curve [Opp95; Bel+19].

Machine learning algorithms are often regularized during training to improve performance, and similar to model size, the amount of regularization can control a bias-variance tradeoff. Indeed, recently, double descent behavior was reported as a function of training epochs and weight decay [Nak+20]. Understanding such double descent behavior is important because it can be critical for good performance, especially for learning from noisy labels [Arp+17; YH20].

The perhaps most popular regularization technique is to add an explicit ℓ_2-norm penalty to the training loss (i.e., a term $\lambda\|\boldsymbol{\theta}\|_2^2$), or training with weight decay, in deep learning semantics. Double descent as a function of the regularization parameter λ has been reported for a ResNet-18 network trained on CIFAR-10 with label noise [Nak+20, Figure 22], but a theoretical understanding and a more extensive empirical study covering a variety of models is still lacking.

In this paper, we therefore study the risk of ℓ_2-regularized models as a function of the regularization strength λ, both in theory and practice. Our contributions are as follows:

- Our empirical results show that various neural networks regularized with an ℓ_2-penalty $\lambda\|\boldsymbol{\theta}\|_2^2$ can exhibit double descent shaped risk curve as a function of $1/\lambda$. That is, the risk or test error

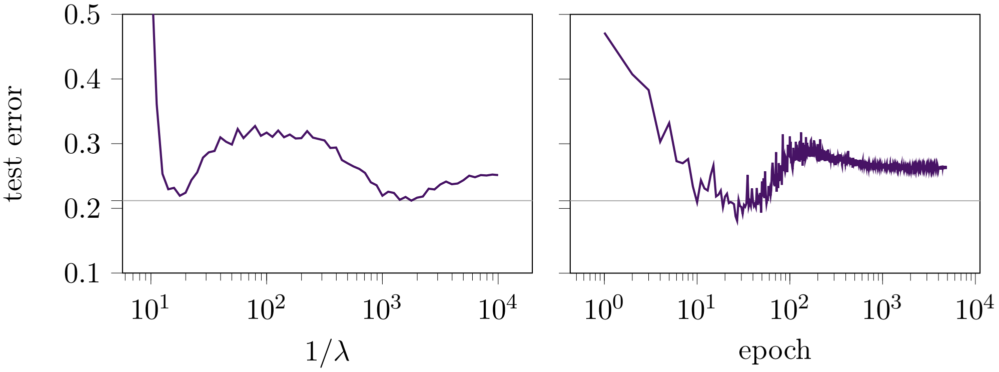


Figure 1: Test performance of a 5-layer convolution network when trained on the CIFAR-10 dataset with 20% label noise. **Left:** Performance as a function of the inverse regularization parameter $1/\lambda$ if the network is trained with ℓ_2-regularization until convergence. **Right:** Performance as a function of the training epochs when the network is trained without ℓ_2 regularization. **Both:** In both cases, the network exhibits double descent behavior as a function of both the regularization by λ and regularization by early stopping the training.

first decreases, then increases, and then decreases again as a function of $1/\lambda$ (see Figure 1). This frequently occurs when training on noisy data, but can also occur when training standard models (a CNN) on a standard noise-less dataset (CIFAR-100).

- Next, we consider a linear ridge regression model and theoretically characterize the risk as a function λ. We show that when the features have different scales, similarly to early-stopped least squares studied in [HY21], the risk of the ridge regression solution as a function of $1/\lambda$ is a superposition of bias-variance tradeoffs, which yields a double descent behavior.
- Finally, we consider a non-linear two-layer neural network and provide numerical examples where double descent occurs as a function of $1/\lambda$ and, motivated by our theory in the linear case, eliminate the double descent by utilizing differently scaled λ values for the two layers. Eliminating double descent is interesting as it typically improves the performance of the best model.

While conceptually our results for explicit ℓ_2-regularization parallel those for early stopping developed in our earlier paper [HY21], early-stopping and ℓ_2-regularization often behave quite differently: Figure 1 shows the test error of a 5-layer CNN as a function of $1/\lambda$ when trained on the noisy CIFAR-10 dataset, and contrasts this to the test error as a function of the training epochs (with no ℓ_2-regularization). Note that the test error as a function of (inverse) regularization strength exhibits a double descent behavior and regularization with early stopping exhibits a double descent behavior (as shown before by [Nak+20]), but the effect of ℓ_2-regularization and early stopping is not the same as the ℓ_2-regularization allows attaining the same best-case performance in two distinct regimes, whereas the early stopped risk does not.

2 Related works

Double descent as a function of the model size has been theoretically established for linear regression [Has+19; BHX20; Mit19] and for random feature regression [MM19; D'A+20]. Double descent has also been studied as a function of training time [HY21; Zha+21] and sample complexity [Nak19]. Nakkiran et al. [Nak+20] have provided several empirical examples of epoch-wise, sample-wise, and regularization-wise double descent for deep networks. Beyond double descent, multiple decent has also been shown and characterized in the paper [LRZ20; HY21].

A recent line of theoretical model-wise double descent works studied the behavior of the risk, specifically by decomposing the risk into bias and variance terms [Jac+20; Yan+20; D'A+20; LR20; LRZ20]. Several works have further decomposed bias-variance terms with respect to the different sources of randomness in training, such as the optimization process or data distribution [Nea+19; AP20b]. Our model also relies on the interaction between the data and the model parameters to study double descent.

For epoch-wise double descent, Heckel and Yilmaz [HY21] characterized the risk as a function of the training time as a superposition of multiple bias-variance tradeoffs, which yields double descent for misaligned features. For a setup with misaligned features, we show an analogous result where we decompose the risk as a function of $1/\lambda$ as a superposition of bias-variance tradeoffs.

Generalization and training dynamics of deep networks with ℓ_2 regularization in the form of *weight decay* has been a topic of interest, particularly regarding finding optimal setups, such as finding the optimal weight matrix based on the data prior for weighted regularization [WX20]. Nakkiran et al. [Nak+21] have shown that optimal ℓ_2 regularization can mitigate model-wise and sample-wise double descent, analytically for linear regression and empirically for CNNs.

Many works used neural-tangent-kernels (NTKs) [JGH18], to study the double descent behavior, as a function of the network width [AP20a] and training epochs [HY21], as well as to understand the dynamics of ℓ_2-regularized neural network training [Wei+19; LG20]. Lewkowycz and Gur-Ari [LG20] demonstrated that the NTK deviates significantly from initialization after a time that is inversely proportional to the regularization strength.

3 Ridge regression risk as a function of the regularization parameter

We start with studying the risk of the ridge regression estimator with regularization parameter λ, for fitting a linear model to data generated by a Gaussian linear model. We show that the risk as a function of $1/\lambda$ is a superposition of U-shaped bias-variance tradeoffs. If the features of the Gaussian linear model have different scales, those bias-variance tradeoff curves can add up to a double (or multiple) descent shaped risk curve.

3.1 Data model and risk

We consider the same linear regression setup as Heckel and Yilmaz [HY21]. Consider a regression problem, and suppose data is generated from a Gaussian linear model as $y = \langle \mathbf{x}, \boldsymbol{\theta}^* \rangle + z$, where $\mathbf{x} \in \mathbb{R}^d$ is a zero-mean Gaussian feature vector with diagonal co-variance matrix $\boldsymbol{\Sigma} = \mathrm{diag}(\sigma_1^2, \ldots, \sigma_d^2)$, and z is independent, zero-mean Gaussian noise with variance σ^2. We are given a training set $\mathcal{D} = \{(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_n, y_n)\}$ consisting of n data points drawn iid from this Gaussian linear model.

Consider a linear estimator parameterized by a vector $\hat{\boldsymbol{\theta}} \in \mathbb{R}^d$ which predicts the label associated with a feature vector $\mathbf{x}$ as $\hat{y} = \left\langle \mathbf{x}, \hat{\boldsymbol{\theta}} \right\rangle$. The (mean-squared) risk of this estimator is

$$R(\hat{\boldsymbol{\theta}}) = \mathbb{E}\left[\left(y - \left\langle \mathbf{x}, \hat{\boldsymbol{\theta}} \right\rangle\right)^2\right],$$

where the expectation is over an example $(\mathbf{x}, y)$ drawn independently (of the training set) from the underlying linear model. The risk of the estimator can be written as a function of the variances of the features and of the coefficients of the underlying true linear model, $\boldsymbol{\theta}^* = [\theta_1^*, \ldots, \theta_d^*]$, as

$$R(\hat{\boldsymbol{\theta}}) = \sigma^2 + \sum_{i=1}^{d} \sigma_i^2 (\theta_i^* - \hat{\theta}_i)^2. \tag{1}$$

3.2 Risk of the ridge regression estimator

Consider the ridge regression estimator defined as

$$\hat{\boldsymbol{\theta}}_\lambda = \arg\min_{\boldsymbol{\theta}} \frac{1}{2} \sum_{i=1}^{n} (\langle \mathbf{x}_i, \boldsymbol{\theta} \rangle - y_i)^2 + \frac{\lambda}{2} \|\boldsymbol{\theta}\|_2^2.$$

We show that in the underparameterized regime, where $d \ll n$, the risk of the ridge regression estimate, $R(\hat{\boldsymbol{\theta}}_\lambda)$, is very well approximated by

$$\bar{R}(\tilde{\boldsymbol{\theta}}_\lambda) = \sigma^2 + \sum_{i=1}^{d} \underbrace{\sigma_i^2 (\theta_i^*)^2 \left(\frac{\lambda}{\sigma_i^2 + \lambda}\right)^2 + \frac{\sigma^2}{n} \sigma_i^2 \left(\frac{\sigma_i}{\sigma_i^2 + \lambda}\right)^2}_{V_i(\lambda)}, \tag{2}$$

as formalized by the theorem below. We focus on the underparameterized regime because only in that regime a linear estimator can have small risk for data generated from a linear model (with non-vanishing features). We consider the overparameterized regime in a more general setting empirically in the next section.

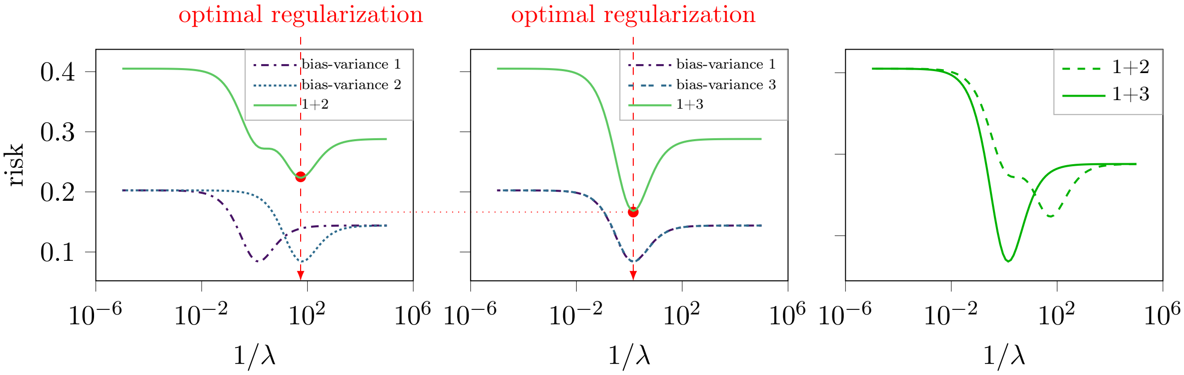
Theorem 1. *With probability at least $1 - 2d^{-5} - 2de^{-n/8} - e^{-d} - 2e^{-32}$ over the random training set generated by a linear Gaussian model with parameters $\boldsymbol{\theta}^*$ and $\boldsymbol{\Sigma}$, the difference of the ℓ_2-regularized least squares risk and the risk expression in (2) is at most*

$$\begin{aligned} \left| R(\hat{\boldsymbol{\theta}}_\lambda) - \bar{R}(\tilde{\boldsymbol{\theta}}_\lambda) \right| \leq c \Bigg[\frac{\max_i \sigma_i^8}{\min_i (\sigma_i^2 + \lambda)^4} \frac{d}{n} & \cdot \\ \left(\left(\frac{\min_i \sigma_i^2 + \lambda}{\max_i \sigma_i^2} + 1 \right) \|\boldsymbol{\Sigma}\boldsymbol{\theta}^*\|_2 + \frac{d \log d}{\sqrt{n}} \sigma \right)^2 + \frac{\sqrt{d}}{n} \sigma^2 \Bigg]. \end{aligned} \tag{3}$$

Here, c is a numerical constant.

Theorem 1 establishes that the risk $R(\hat{\boldsymbol{\theta}}_\lambda)$ is well approximated by the expression $\bar{R}(\tilde{\boldsymbol{\theta}}_\lambda)$, provided the model is sufficiently underparameterized (i.e., d/n is small).

As a consequence, the risk of the ridge regression solution, as a function of $1/\lambda$, is a superposition of U-shaped bias variance tradeoffs. This yields double descent whenever the features of the



a) constant λ b) elimination with diff. λ c) before & after elimination

Figure 2: Ridge regression risk for a two-feature Gaussian linear model as a function of the inverse regularization strength parameter λ. **a:** Two U-shaped bias-variance tradeoffs $V_i(\lambda)$ for the parameters $\theta_1^* = 1.5, \sigma_1 = 1$ (bias-variance 1) and $\theta_2^* = 10, \sigma_2 = 0.15$ (bias-variance 2), along with their sum (1+2) which determines the risk. **b:** Same plot, but this time the bias-variance tradeoff $V_2(\lambda)$ is shifted to the left by increasing the inverse regularization strength $1/\lambda_2$ according to Proposition 1 (yielding bias-variance tradeoff 3), so that its minimum overlaps with that of bias-variance tradeoff 1. This eliminates double descent and gives better performance. **c:** The resulting risk curves before and after elimination, demonstrating that the minimum of the risk after double descent elimination is smaller than before elimination.

underlying data have different scales. This follows from noting that the term $\sigma_i^2(\theta_i^*)^2\left(\frac{\lambda}{\sigma_i^2+\lambda}\right)^2$ in the RHS of (2) increases in λ, whereas the other term $\frac{\sigma^2}{n}\sigma_i^2\left(\frac{\sigma_i}{\sigma_i^2+\lambda}\right)^2$ decreases in λ. See Figure 2(a) as an example.

3.3 Eliminating double descent with scaled regularization

We next show that double descent can be eliminated by utilizing differently scaled λ for different parts (parameters) of the model. For this, we consider a generalized ridge regression problem where we allow different regularization strength to be used for each parameter (sometimes called Tikhonov regularization). Specifically, we let

$$\hat{\boldsymbol{\theta}}_{\boldsymbol{\Lambda}} = \arg\min_{\boldsymbol{\theta}} \frac{1}{2}\sum_{i=1}^{n}(\langle \mathbf{x}_i, \boldsymbol{\theta}\rangle - y_i)^2 + \|\boldsymbol{\Lambda}\boldsymbol{\theta}\|_2^2, \tag{4}$$

where $\boldsymbol{\Lambda}$ is a $\mathbb{R}^{d\times d}$ diagonal matrix containing regularization parameters $\sqrt{\lambda_i}$ along its diagonal.

Proposition 1. *For the generalized ridge regression problem described above, the minimum of the risk expression* $\min_{\lambda_1,\ldots,\lambda_d} \bar{R}(\tilde{\theta}_{\boldsymbol{\Lambda}})$ *is achieved by choosing the regularization strengths associated with different features as* $\lambda_i = \frac{\sigma^2}{n}(\theta_i^*)^{-2}$.

In Figure 2b, we show that double descent can be eliminated, and that this improves the optimal risk, by utilizing the regularization parameters in Proposition 1. Note that double descent is eliminated by picking the optimal regularization strength associated with feature j as

$\bar{\lambda}_{opt} = \lambda_j = \frac{\sigma^2}{n}(\theta_j^*)^{-2}$ and scaling the regularization strengths of the rest of the features proportionally with $(\theta_j^*/\theta_i^*)^2$ to align the minima of the U-shaped bias-variance tradeoff curve $V_i(\lambda_i)$ with the minima of the bias-variance tradeoff curve of the j^{th} feature $V(\bar{\lambda}_{opt})$.

Note that the optimal regularization strength does not depend on the variances of the features, i.e., at the optimal regularization point, the effect of the feature variances on the bias and variance components of the risk is equal in magnitude, i.e., the tradeoff does not depend on the feature variances other than a constant scaling factor for the both bias and variance terms (see SM C.6).

3.4 Relation to early stopping

As already illustrated in Figure 1, in general, ℓ_2 regularization and early stopping have a different effect. However, they *can* have a similar [AKT19], and even equivalent effect in very particular setups. For example, for the linear model studied so far, Tikhonov regularization and early stopping has the same effect if we adjust the regularization strength parameters associated with individual parameters.

Consider the Tikhonov estimator defined in (4). Also consider the estimator which applies t steps of gradient descent to the non-regularized loss $\sum_{i=1}^n (\langle \mathbf{x}_i, \boldsymbol{\theta} \rangle - y_i)^2$, and suppose that each parameter θ_i is updated with an associated stepsize of η_i. This estimator, denoted by $\hat{\boldsymbol{\theta}}^t$ corresponds to early-stopping least-squares. This estimator was studied by Heckel and Yilmaz [HY21] and shown to have risk

$$R(\hat{\boldsymbol{\theta}}^t) \approx \sigma^2 + \sum_{i=1}^d \underbrace{\sigma_i^2 (\theta_i^*)^2 (1 - \eta_i \sigma_i^2)^{2t} + \frac{\sigma^2}{n}(1 - (1 - \eta_i \sigma_i^2)^t)^2}_{U_i(t)} . \tag{5}$$

As formalized by the following proposition, if λ_i are chosen based on the feature variance σ_i and the corresponding stepsize η_i, then the risk expressions for the corresponding Tikhonov estimator and early-stopped least squares are equivalent:

Proposition 2. *Let $\boldsymbol{\Lambda} = diag(\sqrt{\lambda_1}, \ldots, \sqrt{\lambda_d})$ with $\lambda_i = \frac{\sigma_i^2}{1-(1-\eta_i\sigma_i^2)^t} - \sigma_i^2$. Then the risk of Tikhonov regularized least-squares is equal to the risk of early-stopping the gradient descent iterations applied to the non-regularized loss at time t as given in equation* (5).

The above proposition characterizes the requirement such that the bias variance tradeoff curves induced by regularized-least squares are equivalent when using ℓ_2 regularization or regularization by early stopping. However, note that this requires the regularization parameters λ_i to be dependent on the feature variances σ_i^2. In general, where the regularization parameters and stepsizes are the same for each parameter, the risk corresponding to regularization by early stopping and ℓ_2 regularization is different.

4 Double descent in ℓ_2-regularized two-layer neural networks

In this section, we study the risk of a two layer network with weight decay (i.e., ℓ_2-penalty), on data drawn from a Gaussian linear model with a diagonal covariance matrix. We first show empirically that the risk as a function of the regularization parameter has a double descent curve if the variances of the Gaussian model's features decay at a geometric rate, and that the double descent can be eliminated by penalizing the weights in the first and second layers differently.

While it would be nice to explain this theoretically, this is not possible with current linearization techniques: we show that regularization-wise double descent in neural networks occurs outside of the regime where the network dynamics can be characterized by an associated linear model (often called the neural-tangent-kernel (NTK) regime.

4.1 Risk of an overparameterized two-layer network exhibits double descent

We consider a two-layer neural network with relu-nonlinearities, $f(\mathbf{x}) = \text{relu}(\mathbf{W}_1\mathbf{x})\mathbf{w}_2$, where $\mathbf{W}_1 \in \mathbb{R}^{k\times d}$ and $\mathbf{w}_2 \in \mathbb{R}^k$ are the weights in the first and second layer. The network is trained with gradient descent on the mean-squared error loss with ℓ_2-penalty on data drawn from the linear model introduced in Section 3.1 with a diagonal covariance matrix with geometrically decaying covariances and Gaussian zero-mean additive noise. For each value of the regularization parameter λ, we initialize the network with standard Kaiming initialization and train until convergence with stepsize $\eta = 5e - 3$.

Figure 3 shows that the resulting risk follows a double descent curve as a function of $1/\lambda$. Figure 3 also shows that the risk of early-stopped gradient descent, while operating in the same range of values, does not exhibit double descent. This again illustrates that ℓ_2 regularization and regularization by early stopping in general result in different risk curves, as formalized in the previous section for linear models.

Recall that double descent for linear models occurs because different features are scaled differently, and can be mitigated by scaling λ_i appropriately, as formalized in Proposition 1 and demonstrated in Figure 2. Motivated by this result, we hypothesize that the first and second layers of the two-layer neural network overfit the noise at different scales. Thus, utilizing properly scaled λ_1 and λ_2 for the parameters in the first and second layers should mitigate double descent and potentially improve performance.

In Figure 3, we show that double descent is indeed eliminated by using a larger λ for the second layer and that the best performance (i.e., the performance achieved at the optimal regularization point) is improved relative to the best performance for the risk curve where double descent is not eliminated. Note that in the linear case studied in Section 3.3, when the parameters of the underlying data model are known or can be estimated, the optimal λ_i, i.e. per feature regularization strength, can be found analytically. In contrast, for neural networks, this requires treating per-layer regularization strengths as hyperparameters and tuning them accordingly.

4.2 Double descent occurs outside the linear regime in neural networks

Given our theoretical results for the linear model, and the similar empirical behavior of linear models and neural networks, it is tempting to think that the behavior of the two-layer network from the previous section (and potentially deeper networks) can be described theoretically by linearizing the network around the initialization, and studying the linearized model as a proxy for the actual non-linear network. This regime is known as the NTK regime [JGH18] because the model behaves like a kernel method with a kernel associated with the neural network called neural tangent kernel.

Unfortunately, double descent as a function of λ occurs outside of the regime where a linear approximation is accurate, as we discuss here.

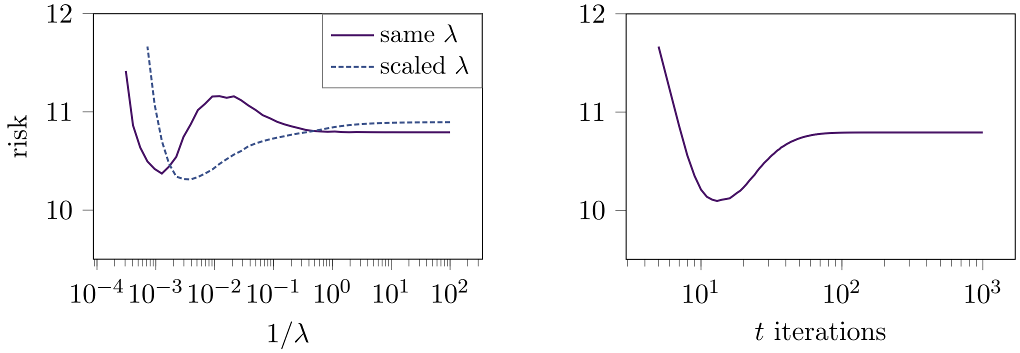


Figure 3: **Left:** Risk of the two-layer neural network trained on the linear data with a diagonal covariance matrix with geometrically decaying singular values and added noise with ℓ_2 regularization as a function of the inverse regularization parameter $1/\lambda$. The risk exhibits the double descent behavior. **Right:** Same risk as a function of the training iterations t for $\lambda = 0$. The early-stopped risk does not yield a double descent behavior. **Both:** Training dynamics of regularization by early stopping cannot be approximated by the solutions of the corresponding ℓ_2 regularization problem.

Consider a neural network with parameter vector $\boldsymbol{\theta}$ and input $\mathbf{x}$, denoted by $f_{\boldsymbol{\theta}}(\mathbf{x})$. Suppose we train the network on a dataset $\{(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_n, y_n)\}$ by applying gradient descent to the ℓ_2-regularized least-squares loss

$$\mathcal{L}_\lambda(\boldsymbol{\theta}) = \frac{1}{2}\sum_{i=1}^{n}(f_{\boldsymbol{\theta}}(\mathbf{x}_i) - y_i)^2 + \frac{\lambda}{2}\|\boldsymbol{\theta}\|_2^2$$

until convergence. The predictions of the network in a small radius around the initialization $\boldsymbol{\theta}_0$ are well described by the linear approximation $\mathbf{f}_{\boldsymbol{\theta}} \approx \mathbf{J}(\boldsymbol{\theta} - \boldsymbol{\theta}_0) + \mathbf{f}_{\boldsymbol{\theta}_0}$, where

$$\mathbf{f}_{\boldsymbol{\theta}} = \begin{bmatrix} f_{\boldsymbol{\theta}}(\mathbf{x}_1) \\ \ldots \\ f_{\boldsymbol{\theta}}(\mathbf{x}_n) \end{bmatrix} \text{ and } \mathbf{J} = \begin{bmatrix} \nabla_{\boldsymbol{\theta}} f_{\boldsymbol{\theta}}(\mathbf{x}_1) \\ \ldots \\ \nabla_{\boldsymbol{\theta}} f_{\boldsymbol{\theta}}(\mathbf{x}_n) \end{bmatrix} \tag{6}$$

are the prediction of the network and the Jacobian of the network at initialization, respectively. The linear approximation is only accurate in a radius around the initialization, in which each individual parameter changes very little. However, as we argue in more detail in the supplement, the individual parameters change too much for this approximation to be accurate (see Figure 7, left), unless the singular values of the Jacobian are large relative to λ. However, we note that the individual parameters change too much for this approximation to be accurate, unless the singular values of the Jacobian are large relative to λ. If the singular values are sufficiently large for the NTK approximation to be accurate, however, the regularization has a vanishing effect, and in the regime where the regularization has a vanishing effect, no double descent occurs. We refer to SM D for a more detailed analytical discussion.

5 Double descent in deep networks

We next study a 5-layer CNN and ResNet-18 to demonstrate that regularization-wise double descent occurs in standard deep learning settings. We first look at the test error of a 5-layer CNN trained on

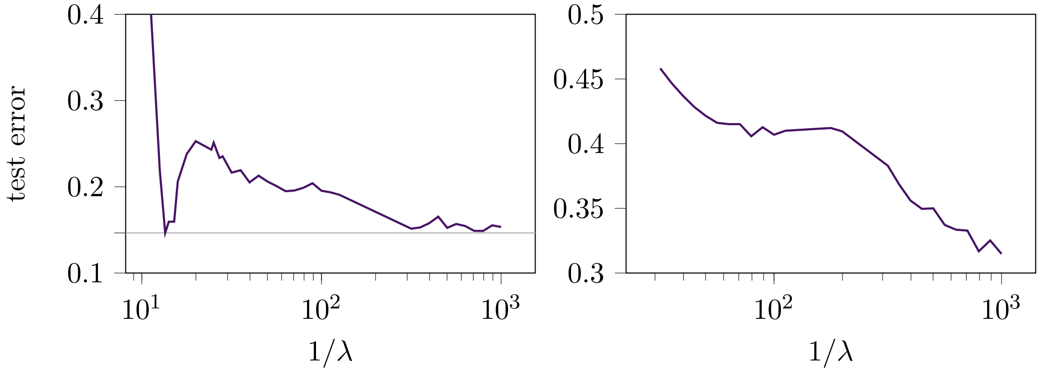


Figure 4: Regularization-wise double descent for models and datasets of more practical interest: **(Left)** Test performance of ResNet-18 as a function of the inverse regularization parameter ($1/\lambda$) when trained on the CIFAR-10 dataset with 20% label noise exhibits double descent; **(Right)** Test performance of the 5-layer CNN as a function of the inverse regularization parameter ($1/\lambda$) when trained on the CIFAR-100 dataset with *no label noise* also exhibits (subtler) double descent

the CIFAR-10 dataset with 20% label noise as a function of the regularization strength, or weight decay.We also compare this curve to the unregularized training curve as a function of training epochs, which also exhibits double descent, to demonstrate that the two regularizations function distinctively differently.

Our results in Figure 1 show that the test error as a function of regularization strength follows a double descent curve.Moreover, while there is a clear optimal λ value where the minimum test error is achieved in the small λ regime, which coincides with the typical values of weight decay used in practice, a similar performance can be achieved in the much larger λ regime.

Note that double descent can be potentially eliminated with *more* regularization. Nakkiran et al. [Nak+21] showed that sample-wise double descent can be eliminated by employing optimal ℓ_2-regularization. We report that regularization-wise double descent can also be eliminated by employing early-stopping in conjunction with weight decay and epoch-wise double descent by employing optimally-tuned weight decay (see SM A, Figure 5).

Moreover, in both cases, eliminating the double descent improves the performance compared to the case where ℓ_2 regularization or early stopping is individually applied.

While CNNs are commonly used for vision applications, standard architectures feature more complex mechanisms, such as residual links, and hence the training dynamics of such models can significantly vary from that of the simple 5-layer CNN. We therefore also study the test error of the ResNet-18 model trained on the CIFAR-10 dataset with 20% label noise. We show that, in Figure 4 (left), the test error for ResNet-18 also exhibits double descent even though the achieved performance across all λ values is better for ResNet-18 than the 5-layer CNN as can be expected. Moreover, similarly to the case of the 5-layer CNN, a similar test error can be achieved at both small and large λ regimes.

For deep learning models trained on image classification datasets, the double descent phenomenon is primarily observed when the model is trained on noisy data. For example, epoch-wise double descent [Nak+20; HY21] has only been observed in practical setups when training on noisy data (i.e., data with label noise).

We next show that regularization-wise double descent can also occur in more practical settings, i.e. when there is no label noise, which is the most common situation in practice. Our results in Figure 4 (right) show that the test error of the 5-layer CNN trained on the CIFAR-100 dataset with no label noise also exhibits double descent, albeit in a less pronounced manner. This is expected, since higher levels of noise in general lead to a more pronounced double descent curve.

6 Conclusion

In this work, we studied regularization-wise double descent in an effort to bring its understanding to the same level as the previously well-studied model-wise, epoch-wise and sample-wise double descents. We demonstrated that the test error of standard deep networks trained on standard image classification datasets can follow a double descent curve as a function of ℓ_2 regularization strength (weight decay) both when there is label noise (CIFAR-10) and without any label noise (CIFAR-100).

We show that regularization-wise double descent can be explained as a superposition of bias-variance tradeoffs pertaining to different features of the data (for a linear model) or parts of the neural network, and that double descent can be eliminated by scaling the regularization strengths accordingly.

Code

Code to reproduce the experiments is available at https://github.com/MLI-lab/regularization-wise_double_d

Acknowledgements

F. F. Yilmaz and R. Heckel are (partially) supported by NSF under award IIS-1816986. R. Heckel is also supported by the Institute of Advanced Studies at the Technical University of Munich, and the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) - 456465471, 464123524.

References

[AP20a] B. Adlam and J. Pennington. “The neural tangent kernel in high dimensions: Triple descent and a multi-scale theory of generalization”. In: *International Conference on Machine Learning (ICML)*. 2020.

[AP20b] B. Adlam and J. Pennington. “Understanding double descent requires a fine-grained bias-variance decomposition”. In: *Advances in Neural Information Processing Systems (NeurIPS)*. 2020.

[AKT19] A. Ali, J. Z. Kolter, and R. J. Tibshirani. “A continuous-time view of early stopping for least squares regression”. In: *International Conference on Artificial Intelligence and Statistics (AISTATS)*. 2019.

[Arp+17] D. Arpit, S. Jastrzębski, N. Ballas, D. Krueger, E. Bengio, M. S. Kanwal, T. Maharaj, A. Fischer, A. Courville, Y. Bengio, et al. “A closer look at memorization in deep networks”. In: *International Conference on Machine Learning (ICML)*. 2017.

[Bel+19] M. Belkin, D. Hsu, S. Ma, and S. Mandal. “Reconciling modern machine-learning practice and the classical bias–variance trade-off”. In: *Proceedings of the National Academy of Sciences* (2019).

[BHX20] M. Belkin, D. Hsu, and J. Xu. “Two models of double descent for weak features”. In: *SIAM Journal on Mathematics of Data Science* (2020).

[D’A+20] S. D’Ascoli, M. Refinetti, G. Biroli, and F. Krzakala. “Double trouble in double descent: bias and variance(s) in the lazy regime”. In: *International Conference on Machine Learning (ICML)*. 2020.

[FR13] S. Foucart and Rauhut, Holger. *A mathematical introduction to compressive sensing.* Springer Berlin Heidelberg, 2013.

[Has+19] T. Hastie, A. Montanari, S. Rosset, and R. J. Tibshirani. “Surprises in high-dimensional ridgeless least squares interpolation”. In: *arXiv:1903.08560 [cs, math, stat]* (2019).

[HY21] R. Heckel and F. F. Yilmaz. “Early stopping in deep networks: Double descent and how to eliminate it”. In: *International Conference on Learning Representations (ICLR)*. 2021.

[HJ12] R. A. Horn and C. R. Johnson. *Matrix analysis.* 2nd. Cambridge University Press, 2012.

[JGH18] A. Jacot, F. Gabriel, and C. Hongler. “Neural tangent kernel: Convergence and generalization in neural networks”. In: *Advances in Neural Information Processing Systems (NeurIPS)*. 2018.

[Jac+20] A. Jacot, B. Simsek, F. Spadaro, C. Hongler, and F. Gabriel. “Implicit regularization of random feature models”. In: *International Conference on Machine Learning (ICML)*. 2020.

[LG20] A. Lewkowycz and G. Gur-Ari. “On the training dynamics of deep networks with L_2 regularization”. In: *Advances in Neural Information Processing Systems (NeurIPS)*. 2020.

[LR20] T. Liang and A. Rakhlin. “Just interpolate: kernel ridgeless regression can generalize”. In: *The Annals of Statistics* 3 (2020).

[LRZ20] T. Liang, A. Rakhlin, and X. Zhai. "On the multiple descent of minimum-norm interpolants and restricted lower isometry of kernels". In: *Proceedings of Thirty Third Conference on Learning Theory.* 2020.

[MM19] S. Mei and A. Montanari. "The generalization error of random features regression: precise asymptotics and double descent curve". In: *Communications on Pure and Applied Mathematics* (2019).

[Mit19] P. P. Mitra. "Understanding overfitting peaks in generalization error: analytical risk curves for l_2 and l_1 penalized interpolation". In: *arXiv:1906.03667 [physics, stat]* (2019).

[Nak19] P. Nakkiran. "More data can hurt for linear regression: Sample-wise double descent". In: *arXiv:1912.07242 [cs, math, stat]* (2019).

[Nak+20] P. Nakkiran, G. Kaplun, Y. Bansal, T. Yang, B. Barak, and I. Sutskever. "Deep double descent: Where bigger models and more data hurt". In: *International Conference on Learning Representations (ICLR).* 2020.

[Nak+21] P. Nakkiran, P. Venkat, S. M. Kakade, and T. Ma. "Optimal regularization can mitigate double descent". In: *International Conference on Learning Representations (ICLR).* 2021.

[Nea+19] B. Neal, S. Mittal, A. Baratin, V. Tantia, M. Scicluna, S. Lacoste-Julien, and I. Mitliagkas. "A modern take on the bias-variance tradeoff in neural networks". In: (2019).

[Opp95] M. Opper. "Statistical mechanics of learning : Generalization". In: *The Handbook of Brain Theory and Neural Networks.* 1995.

[Wai19] M. Wainwright. *High dimensional statistics: A non-asymptotic viewpoint.* Cambridge University Press, 2019.

[Wei+19] C. Wei, J. D. Lee, Q. Liu, and T. Ma. "Regularization matters: Generalization and optimization of neural nets vs their induced kernel". In: *Advances in Neural Information Processing Systems (NeurIPS)* (2019).

[WX20] D. Wu and J. Xu. "On the optimal weighted ℓ_2 regularization in overparameterized linear regression". In: *Advances in Neural Information Processing Systems (NeurIPS)* (2020).

[Yan+20] Z. Yang, Y. Yu, C. You, J. Steinhardt, and Y. Ma. "Rethinking bias-variance trade-off for generalization of neural networks". In: *International Conference on Machine Learning (ICML).* 2020.

[YH20] F. F. Yilmaz and R. Heckel. "Image recognition from raw labels collected without annotators". In: *arXiv:1910.09055 [cs, stat]* (2020).

[Zha+17] C. Zhang, S. Bengio, M. Hardt, B. Recht, and O. Vinyals. "Understanding deep learning requires rethinking generalization". In: *International Conference on Learning Representations (ICLR).* 2017.

[Zha+21] X. Zhang, D. Wu, H. Xiong, and B. Dai. "Optimization variance: Exploring generalization properties of DNNs". In: *arXiv:2106.01714 [cs]* (2021).

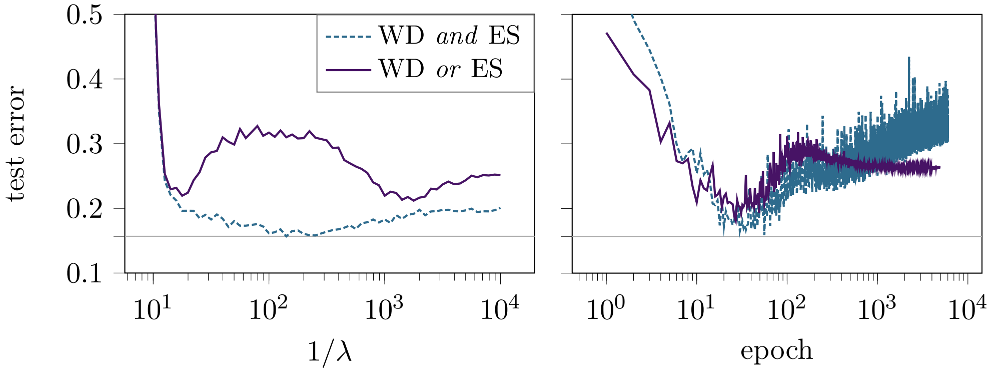


Figure 5: Comparison of individually or jointly applied regularization by early stopping and weight decay: Test performance of the 5-layer convolution network when trained on the CIFAR-10 dataset with 20% label noise. **Left:** Performance as a function of the regularization strength for training with (*solid*) weight decay only—WD—and (*dashed*) weight decay together with early stopping—WD and ES. **Left:** Performance as a function of the training epochs for (*solid*) standard training and (*dashed*) training with weight decay. **Both:** Better performance is achieved by jointly utilizing weight decay and early stopping—WD and ES.

A Double descent behavior of deep networks in the presence of both weight decay and early stopping

Here, we expand on the results provided in Figure 1 and show that both regularization-wise and epoch-wise double descent can be eliminated by employing additional forms of regularization. Specifically, in Figure 5, our results show that utilizing early stopping eliminates regularization-wise double descent, whereas utilizing (tuned) weight decay eliminates the corresponding epoch-wise double descent. Note that performance achieved in the case where early stopping and weight decay are used together is much better than that obtained by using either weight decay or early stopping alone.

B Double descent as a function of dropout regularization

Our results showcasing the double descent behavior as a function of the ℓ_2 regularization strength motivates the investigation of other types of regularization and whether double descent also occurs for other explicit regularization methods. In Figure 6, we show the test error of the 5-layer CNN with dropout added after the activations of each layer trained on the noisy CIFAR-10. The test error exhibits a U-shaped curve as a function of the dropout probability with optimal dropout probability $p_{dropout} = 0.4$.

C Discussion and proof statements for linear ridge regression

In this section, we provide detailed analysis and proofs for the theoretical statements on the linear ridge regression risk studied in Section 3.

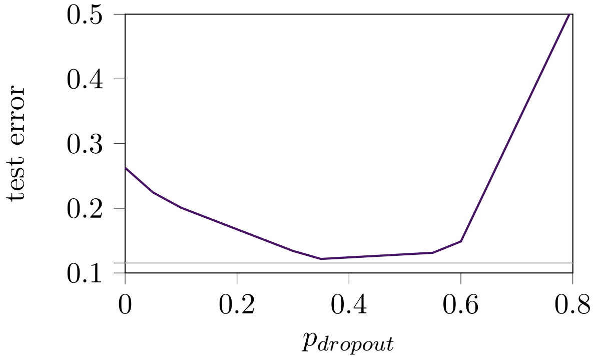


Figure 6: Test performance of the 5-layer convolution network as a function of the dropout probability when trained on the CIFAR-10 dataset with 20% label noise.

C.1 Intuition for the risk expression (2)

We first provide intuition on why the risk is governed by the risk expression given in (2).

First, note that the risk of the resulting estimator can be written as a function of the variances of the features, σ_i^2, and of the coefficients of the underlying true linear model, $\boldsymbol{\theta}^* = [\theta_1^*, \ldots, \theta_d^*]$, as

$$R(\hat{\boldsymbol{\theta}}_\lambda) = \sigma^2 + \sum_{i=1}^{d} \sigma_i^2 (\theta_i^* - \hat{\theta}_{\lambda,i})^2. \tag{7}$$

which follows from noting that z and $\mathbf{x}$ are independently drawn.

Next, note that we aim to find the estimator which minimizes the ell_2-regularized MSE loss

$$\mathcal{L}_\lambda(\boldsymbol{\theta}) = \frac{1}{2}\|\mathbf{X}\boldsymbol{\theta} - \mathbf{y}\|_2^2 + \frac{\lambda}{2}\|\boldsymbol{\theta}\|_2^2.$$

Recall that, as introduced in Section 3.1 , the matrix $\mathbf{X} \in \mathbb{R}^{n \times d}$ contains the scaled training feature vectors $\frac{1}{\sqrt{n}}\mathbf{x}_1, \ldots, \frac{1}{\sqrt{n}}\mathbf{x}_n$ as rows, and $\mathbf{y} = \frac{1}{\sqrt{n}}[y_1, \ldots, y_n]$ are the corresponding scaled responses. Then, the solution of the ℓ_2 regularized problem can be found by simply setting the gradient of the loss function to zero and solving for $\boldsymbol{\theta}$, which yields

$$\boldsymbol{\theta}_\lambda - \boldsymbol{\theta}^* = ((\mathbf{X}^T\mathbf{X} + \lambda\mathbf{I})^{-1}\mathbf{X}^T\mathbf{X} - \mathbf{I})\boldsymbol{\theta}^* + (\mathbf{X}^T\mathbf{X} + \lambda\mathbf{I})^{-1}\mathbf{X}^T\mathbf{z},$$

where $\mathbf{z} = [z_1, \ldots, z_n]$ is the noise. As we formalize below, in the under-parameterized regime where $n \gg d$, we have that $\mathbf{X}^T\mathbf{X} \approx \boldsymbol{\Sigma}^2$. Therefore the original solution is close to the proximal solution $\tilde{\boldsymbol{\theta}}_\lambda$ defined by

$$\tilde{\boldsymbol{\theta}}_\lambda - \boldsymbol{\theta}^* = ((\boldsymbol{\Sigma}^T\boldsymbol{\Sigma} + \lambda\mathbf{I})^{-1}\boldsymbol{\Sigma}^T\boldsymbol{\Sigma} - \mathbf{I})\boldsymbol{\theta}^* + (\boldsymbol{\Sigma}^T\boldsymbol{\Sigma} + \lambda\mathbf{I})^{-1}\mathbf{X}^T\mathbf{z}, \tag{8}$$

The proximal solution is close to the original solution obtained by solving for the minimizer of the ℓ_2-regularized loss function. Note that, from (8), we get, for the i-th entry of $\tilde{\boldsymbol{\theta}}_\lambda$

$$\tilde{\boldsymbol{\theta}}_{\lambda,i} - \boldsymbol{\theta}_i^* = \tilde{\mathbf{x}}_i^T\mathbf{z}\frac{1}{\sigma_i^2 + \lambda} - \frac{\lambda}{\sigma_i^2 + \lambda}\boldsymbol{\theta}_i^*,$$

where $\tilde{\mathbf{x}}_i$ is the i-th *column* of $\mathbf{X}$ (not the i-th example/feature vector!). Next note that, $\mathbb{E}\left[(\tilde{\mathbf{x}}_i^T \mathbf{z})^2\right] \approx \sigma^2\sigma_i^2$ because the entries of $\mathbf{z}$ are $\mathcal{N}(0, \sigma^2)$ distributed, and the entries of $\tilde{\mathbf{x}}_i$ are $1/\sqrt{n}\mathcal{N}(0, \sigma_i^2)$ distributed. Using this expectation in the solution $\tilde{\boldsymbol{\theta}}_\lambda$, and evaluating the resulting risk of those iterates via the formula for the risk given by (7) yields the risk expression (2). The proof of Theorem 1 in this appendix makes this intuition precise by formally bounding the difference of the proximal solution $\tilde{\boldsymbol{\theta}}_\lambda$ to the original solution $\boldsymbol{\theta}_\lambda$.

C.2 Proof of Theorem 1

In this section, we provide the formal proof for Theorem 1.

The difference between the two risk terms can be further dissected into two separate terms:

$$\left|R(\boldsymbol{\theta}_\lambda) - \bar{R}(\tilde{\boldsymbol{\theta}}_\lambda)\right| \leq \left|R(\boldsymbol{\theta}_\lambda) - R(\tilde{\boldsymbol{\theta}}_\lambda)\right| + \left|R(\tilde{\boldsymbol{\theta}}_\lambda) - \bar{R}(\tilde{\boldsymbol{\theta}}_\lambda)\right|. \tag{9}$$

We bound the two terms on the RHS of (9) separately. We first provide a bound for the first term with the lemma below.

Lemma 1. *Define $\tilde{\mathbf{X}}$ so that $\mathbf{X} = \tilde{\mathbf{X}}\boldsymbol{\Sigma}$. Suppose that $\left\|\mathbf{I} - \tilde{\mathbf{X}}^T\tilde{\mathbf{X}}\right\| \leq \epsilon$, with $\epsilon \leq (\min_i \sigma_i^2 + \lambda)/2$ Then*

$$\left|R(\boldsymbol{\theta}_\lambda) - R(\tilde{\boldsymbol{\theta}}_\lambda)\right| \leq 4\epsilon^2 \left(\frac{\max_i \sigma_i^4}{\min_i(\sigma_i^2+\lambda)^2}\right)^2 \left(\left(\frac{\min_i \sigma_i^2 + \lambda}{\max_i \sigma_i^2} + 1\right) \|\boldsymbol{\Sigma}\boldsymbol{\theta}^*\|_2 + \left\|\tilde{\mathbf{X}}^T\mathbf{z}\right\|_2\right)^2 \tag{10}$$

We apply the lemma by first verifying its condition by referring to the derivations in [HY21, Lemma 1]. Note that the entries of the matrix $\tilde{\mathbf{X}}$ are iid Gaussians drawn from $\mathcal{N}(0, 1/n)$, and the same concentration inequality from [FR13, Chapter 9] results in, for any $\beta \in (0, 1)$,

$$\mathrm{P}\left[\left\|\mathbf{I} - \tilde{\mathbf{X}}^T\tilde{\mathbf{X}}\right\| \geq \beta\right] \leq e^{-\frac{n\beta^2}{15}+4d}.$$

With $\beta = \sqrt{\frac{75d}{n}}$ we obtain that, with probability at least $1 - e^{-d}$,

$$\left\|\mathbf{I} - \tilde{\mathbf{X}}^T\tilde{\mathbf{X}}\right\| \leq \sqrt{75\frac{d}{n}}.$$

We next bound $\left\|\tilde{\mathbf{X}}^T\mathbf{z}\right\|_2$ with high probability:

Lemma 2. *With $\tilde{\mathbf{X}}$ previously defined such that $\mathbf{X} = \tilde{\mathbf{X}}\boldsymbol{\Sigma}$, with probability at least $1 - 2d(e^{-\beta^2/2} + e^{-n/8})$,*

$$\left\|\tilde{\mathbf{X}}^T\mathbf{z}\right\|_2 \leq 2\frac{d}{\sqrt{n}}\sigma\beta$$

Applying the lemma with $\beta^2 = 10\log(d)$, we obtain that with probability at least $1 - 2d^{-5} - 2de^{-n/8} - e^{-d}$ we have

$$\left|R(\boldsymbol{\theta}_\lambda) - R(\tilde{\boldsymbol{\theta}}_\lambda)\right| \leq 4\frac{75d}{n}\left(\frac{\max_i \sigma_i^4}{\min_i(\sigma_i^2+\lambda)^2}\right)^2\left(\left(\frac{\min_i \sigma_i^2 + \lambda}{\max_i \sigma_i^2} + 1\right)\|\boldsymbol{\Sigma}\boldsymbol{\theta}^*\|_2 + 2\frac{d}{\sqrt{n}}\sigma 10\log d\right)^2$$

We finally bound the second term in (9):

Lemma 3. *Provided that $d/n \leq \max_i((\sigma_i + \lambda)/\sigma_i^2)^4$, with probability at least $1 - 4e^{-\frac{\beta^2}{8}}$, we have that*

$$\left|R(\tilde{\boldsymbol{\theta}}_\lambda) - \bar{R}(\tilde{\boldsymbol{\theta}}_\lambda)\right| \leq \frac{\sigma^2}{n}\beta 3\sqrt{d}, \tag{11}$$

with $\bar{R}(\tilde{\boldsymbol{\theta}}_\lambda)$ as defined in (2).

For the proof of Lemma 3 we refer the reader to the proof of [HY21, Lemma 2] and note that (3) can be obtained by following the same steps with the additional assumption regarding the underparameterization as stated in Lemma 3.

We note that the assumption of the lemma is generally satisfied as we operate in the underparameterized regime and poses no strict restriction on the setup. Applying the two bounds (10) and (11) to the RHS of the bound (9) concludes the proof. The remainder of the proof is devoted to proving Lemma 1.

C.3 Proof of Lemma 1

Recall that the solutions of the original and closely related problem are given by

$$\begin{aligned}
\boldsymbol{\theta}_\lambda - \boldsymbol{\theta}^* &= ((\mathbf{X}^T\mathbf{X} + \lambda\mathbf{I})^{-1}\mathbf{X}^T\mathbf{X} - \mathbf{I})\boldsymbol{\theta}^* + (\mathbf{X}^T\mathbf{X} + \lambda\mathbf{I})^{-1}\mathbf{X}^T\mathbf{z},\\
\tilde{\boldsymbol{\theta}}_\lambda - \boldsymbol{\theta}^* &= ((\boldsymbol{\Sigma}^2 + \lambda\mathbf{I})^{-1}\boldsymbol{\Sigma}^2 - \mathbf{I})\boldsymbol{\theta}^* + (\boldsymbol{\Sigma}^2 + \lambda\mathbf{I})^{-1}\mathbf{X}^T\mathbf{z}.
\end{aligned}$$

Note that $\mathbf{X} = \tilde{\mathbf{X}}\boldsymbol{\Sigma}$, where we defined $\tilde{\mathbf{X}}$ which has iid Gaussian entries $\mathcal{N}(0, 1/n)$. With this notation, and using that $\boldsymbol{\Sigma}$ is diagonal and therefore commutes with symmetric matrices, we obtain the following expressions for the residuals of the two solutions:

$$\begin{aligned}
\boldsymbol{\Sigma}\boldsymbol{\theta}_\lambda - \boldsymbol{\Sigma}\boldsymbol{\theta}^* &= \boldsymbol{\Sigma}((\boldsymbol{\Sigma}^2\tilde{\mathbf{X}}^T\tilde{\mathbf{X}} + \lambda\mathbf{I})^{-1}\boldsymbol{\Sigma}^2\tilde{\mathbf{X}}^T\tilde{\mathbf{X}} - \mathbf{I})\boldsymbol{\theta}^* + (\boldsymbol{\Sigma}^2\tilde{\mathbf{X}}^T\tilde{\mathbf{X}} + \lambda\mathbf{I})^{-1}\boldsymbol{\Sigma}^2\tilde{\mathbf{X}}^T\mathbf{z},\\
\boldsymbol{\Sigma}\tilde{\boldsymbol{\theta}}_\lambda - \boldsymbol{\Sigma}\boldsymbol{\theta}^* &= \boldsymbol{\Sigma}((\boldsymbol{\Sigma}^2 + \lambda\mathbf{I})^{-1}\boldsymbol{\Sigma}^2 - \mathbf{I})\boldsymbol{\theta}^* + (\boldsymbol{\Sigma}^2 + \lambda\mathbf{I})^{-1}\boldsymbol{\Sigma}^2\tilde{\mathbf{X}}^T\mathbf{z}.
\end{aligned}$$

The difference between the residuals is

$$\begin{aligned}
\boldsymbol{\Sigma}\boldsymbol{\theta}_\lambda - \boldsymbol{\Sigma}\tilde{\boldsymbol{\theta}}_\lambda = \boldsymbol{\Sigma}^2((\boldsymbol{\Sigma}^2\tilde{\mathbf{X}}^T\tilde{\mathbf{X}} + \lambda\mathbf{I})^{-1}\tilde{\mathbf{X}}^T\tilde{\mathbf{X}} - (\boldsymbol{\Sigma}^2 + \lambda\mathbf{I})^{-1})\boldsymbol{\Sigma}\boldsymbol{\theta}^*&\\
+ \boldsymbol{\Sigma}^2((\boldsymbol{\Sigma}^2\tilde{\mathbf{X}}^T\tilde{\mathbf{X}} + \lambda\mathbf{I})^{-1} - (\boldsymbol{\Sigma}^2 + \lambda\mathbf{I})^{-1})\tilde{\mathbf{X}}^T\mathbf{z}.&
\end{aligned}$$

$$\begin{aligned}
= \boldsymbol{\Sigma}^2(\boldsymbol{\Sigma}^2\tilde{\mathbf{X}}^T\tilde{\mathbf{X}} + \lambda\mathbf{I})^{-1}(I - \tilde{\mathbf{X}}^T\tilde{\mathbf{X}})\boldsymbol{\Sigma}\boldsymbol{\theta}^*&\\
+ \boldsymbol{\Sigma}^2((\boldsymbol{\Sigma}^2\tilde{\mathbf{X}}^T\tilde{\mathbf{X}} + \lambda\mathbf{I})^{-1} - (\boldsymbol{\Sigma}^2 + \lambda\mathbf{I})^{-1})(\boldsymbol{\Sigma}\boldsymbol{\theta}^* - \tilde{\mathbf{X}}^T\mathbf{z}).&
\end{aligned}$$

Where, we added and subtracted $\boldsymbol{\Sigma}^2(\boldsymbol{\Sigma}^2\tilde{\mathbf{X}}^T\tilde{\mathbf{X}}+\lambda\mathbf{I})^{-1}\boldsymbol{\Sigma}\boldsymbol{\theta}^*$ and re-arranged the terms. We bound the norm of the difference between the residuals $\left\|\boldsymbol{\Sigma}\boldsymbol{\theta}_\lambda-\boldsymbol{\Sigma}\tilde{\boldsymbol{\theta}}_\lambda\right\|_2$ by applying Cauchy-Schwarz inequality to the corresponding terms of the RHS of the equation above. We have, for the first term,

$$\begin{aligned}\left\|\boldsymbol{\Sigma}^2(\boldsymbol{\Sigma}^2\tilde{\mathbf{X}}^T\tilde{\mathbf{X}}+\lambda\mathbf{I})^{-1}(I-\tilde{\mathbf{X}}^T\tilde{\mathbf{X}})\boldsymbol{\Sigma}\boldsymbol{\theta}^*\right\| &\le \left\|\boldsymbol{\Sigma}^2\right\|\left\|(I-\tilde{\mathbf{X}}^T\tilde{\mathbf{X}})\right\|\left\|(\boldsymbol{\Sigma}^2\tilde{\mathbf{X}}^T\tilde{\mathbf{X}}+\lambda\mathbf{I})^{-1}\right\|\|\boldsymbol{\Sigma}\boldsymbol{\theta}^*\|_2\\
&\le \max_i \sigma_i^2\epsilon\frac{1}{\min_i\sigma_i^2(1-\epsilon)+\lambda}\|\boldsymbol{\Sigma}\boldsymbol{\theta}^*\|_2\\
&\overset{\text{(i)}}{\le} 2\epsilon\frac{\max_i\sigma_i^2}{\min_i\sigma_i^2+\lambda}\|\boldsymbol{\Sigma}\boldsymbol{\theta}^*\|_2\end{aligned}$$

where we used $1-\epsilon\le\|\tilde{\mathbf{X}}^T\tilde{\mathbf{X}}\|\le 1+\epsilon$ and (i) follows by the assumption $\epsilon\le min_i(\sigma_i^2+\lambda)/2$ both of which follow from the conditions of the lemma.

We next bound the norm of the second term in the difference between the residuals. We have,

$$\begin{aligned}&\left\|\boldsymbol{\Sigma}^2((\boldsymbol{\Sigma}^2\tilde{\mathbf{X}}^T\tilde{\mathbf{X}}+\lambda\mathbf{I})^{-1}-(\boldsymbol{\Sigma}^2+\lambda\mathbf{I})^{-1})(\boldsymbol{\Sigma}\boldsymbol{\theta}^*-\tilde{\mathbf{X}}^T\mathbf{z})\right\|\\
&\qquad\le\left\|\boldsymbol{\Sigma}^2\right\|\left\|(\boldsymbol{\Sigma}^2\tilde{\mathbf{X}}^T\tilde{\mathbf{X}}+\lambda\mathbf{I})^{-1}-(\boldsymbol{\Sigma}^2+\lambda I)^{-1}\right\|\left\|\boldsymbol{\Sigma}\boldsymbol{\theta}^*-\tilde{\mathbf{X}}^T\mathbf{z}\right\|_2\\
&\qquad\overset{\text{(i)}}{\le}\max_i\sigma_i^2\left\|(\boldsymbol{\Sigma}^2\tilde{\mathbf{X}}^T\tilde{\mathbf{X}}+\lambda\mathbf{I})^{-1}\right\|\left\|\boldsymbol{\Sigma}^2(\mathbf{I}-\tilde{\mathbf{X}}^T\tilde{\mathbf{X}})\right\|\left\|(\boldsymbol{\Sigma}^2+\lambda\mathbf{I})^{-1}\right\|\left\|\boldsymbol{\Sigma}\boldsymbol{\theta}^*-\tilde{\mathbf{X}}^T\mathbf{z}\right\|_2\\
&\qquad\le\max_i\sigma_i^2\frac{1}{\min_i(\sigma_i^2(1-\epsilon)+\lambda)}\frac{1}{\min_i(\sigma_i^2+\lambda)}\left\|\boldsymbol{\Sigma}^2\right\|\left\|\mathbf{I}-\tilde{\mathbf{X}}^T\tilde{\mathbf{X}}\right\|\left\|\boldsymbol{\Sigma}\boldsymbol{\theta}^*-\tilde{\mathbf{X}}^T\mathbf{z}\right\|_2\\
&\qquad\le 2\epsilon\frac{\max_i\sigma_i^4}{\min_i(\sigma_i^2+\lambda)^2}\left(\|\boldsymbol{\Sigma}\boldsymbol{\theta}^*\|_2+\|\tilde{\mathbf{X}}^T\mathbf{z}\|_2\right)\end{aligned}$$

where the last inequality follows by the assumption $\epsilon\le min_i(\sigma_i^2+\lambda)/2$, and (i) follows by noting that the matrix $\boldsymbol{\Sigma}^2\tilde{\mathbf{X}}^T\tilde{\mathbf{X}}+\lambda\mathbf{I}$ can be viewed as a perturbation of the non-singular matrix $\boldsymbol{\Sigma}^2+\lambda\mathbf{I}$ such that $\boldsymbol{\Sigma}^2\tilde{\mathbf{X}}^T\tilde{\mathbf{X}}+\lambda\mathbf{I}=(\boldsymbol{\Sigma}^2+\lambda\mathbf{I})-\boldsymbol{\Sigma}^2(\mathbf{I}-\tilde{\mathbf{X}}^T\tilde{\mathbf{X}})$, and applying a standard bound from the literature (see [HJ12, Chapter 5, Equation 5.8.1]) on the difference of the inverse of the two matrices. Combining the two bounds yields (10), which concludes the proof.

C.4 Proof of Lemma 2

We have

$$\left\|\tilde{\mathbf{X}}^T\mathbf{z}\right\|_2=\left|\sum_{l=1}^d(\tilde{\mathbf{x}}_l^T\mathbf{z})^2\right|^{1/2}\le\sum_{l=1}^d\|\tilde{\mathbf{x}}_l^T\mathbf{z}\|_2$$

Conditioned on $\mathbf{z}$, the random variable $\tilde{\mathbf{x}}_i^T\mathbf{z}$ is zero-mean Gaussian with variance $\|\mathbf{z}\|_2/n$. Thus, $\mathrm{P}\left[|\tilde{\mathbf{x}}_i^T\mathbf{z}|\ge\frac{\|\mathbf{z}\|_2}{\sqrt{n}}\beta\right]\le 2e^{-\beta^2/2}$. Moreover, as provided in (13), with probability at least $1-2e^{-n/8}$,

$\|\mathbf{z}\|_2^2 \leq 2\sigma^2$. Combining the two with the union bound, we obtain

$$\mathrm{P}\left[|\tilde{\mathbf{x}}_i^T \mathbf{z}|^2 \geq \frac{2\sigma^2}{n}\beta^2\right] \leq 2e^{-\beta^2/2} + 2e^{-n/8}.$$

Utilizing the union bound again, we obtain

$$\left|\tilde{\mathbf{x}}_l^T \mathbf{z}\right| \leq 2\frac{d}{\sqrt{n}}\sigma\beta$$

which holds with probability at least $1 - 2d(e^{-\beta^2/2} + e^{-n/8})$.

C.5 Proof of Lemma 3

For proving Lemma 3, we follow a similar argument to [HY21, Lemma 3]. We have

$$R(\tilde{\boldsymbol{\theta}}_\lambda) = \sigma^2 + \sum_{i=1}^{d} \sigma_i^2 \underbrace{\left(\sigma_i \theta_i^* \frac{\lambda}{\sigma_i^2 + \lambda} + \frac{\sigma_i}{\sigma_i^2 + \lambda}\tilde{\mathbf{x}}_i^T \mathbf{z}\right)^2}_{Z_i}.$$

Where, $\sum_{i=1}^d Z_i$ corresponds to an off-centered chi-squared distribution with the Z_i. The random variable Z_i, conditioned on $\mathbf{z}$, is a squared Gaussian with variance upper bounded by $\frac{\|\mathbf{z}\|_2}{\sqrt{n}}$ and has expectation

$$\mathbb{E}\left[Z_i\right] = \sigma_i^2(\theta_i^*)^2\left(\frac{\lambda}{\sigma_i^2 + \lambda}\right)^2 + \frac{\|z\|_2^2}{n}\left(\frac{\sigma_i}{\sigma_i^2 + \lambda}\right)^2$$

By a standard concentration inequality of sub-exponential random variables (see e.g. [Wai19, Chapter 2, Equation 2.21]), we get, for $\beta \in (0, \sqrt{d})$ and conditioned on $\mathbf{z}$, that the event

$$\mathcal{E}_1 = \left\{\left|\sum_{i=1}^{d}(Z_i - \mathbb{E}\left[Z_i\right])\right| \leq \frac{\|\mathbf{z}\|_2^2}{n}\sqrt{d}\beta\right\} \tag{12}$$

occurs with probability at least $1 - 2e^{-\frac{\beta^2}{8}}$. With the same standard concentration inequality for sub-exponential random variables, we have that the event

$$\mathcal{E}_2 = \left\{\left|\|\mathbf{z}\|_2^2 - \sigma^2\right| \leq \frac{\sigma^2\beta}{\sqrt{n}}\right\} \tag{13}$$

also occurs with probability at least $1-2e^{-\frac{\beta^2}{8}}$. By the union bound, both events hold simultaneously with probability at least $1 - 4e^{-\frac{\beta^2}{8}}$. On both events, we have that

$$\begin{aligned}
\left|R(\tilde{\boldsymbol{\theta}}^t) - \bar{R}(\tilde{\boldsymbol{\theta}}^t)\right| &= \left|\sum_{i=1}^{d}(Z_i - \mathbb{E}\left[Z_i\right]) + \frac{1}{n}\left(\|\mathbf{z}\|_2^2 - \sigma^2\sigma_i^2\right)\left(\frac{\sigma_i}{\sigma_i + \lambda}\right)^2\right| \\
&\leq \left|\sum_{i=1}^{d}(Z_i - \mathbb{E}\left[Z_i\right])\right| + \frac{d}{n}\max_i\left[\left(\frac{\sigma_i}{\sigma_i + \lambda}\right)^2 |\|\mathbf{z}\|_2^2 - \sigma^2\sigma_i^2|\right] \\
&\leq \frac{\|\mathbf{z}\|_2^2}{n}\sqrt{d}\beta + \frac{d}{n}\frac{1}{\sqrt{n}}\sigma^2\beta\max_i\left[\left(\frac{\sigma_i}{\sigma_i + \lambda}\right)^2 \sigma_i^2\right] \\
&\leq \frac{2\sigma^2}{n}\sqrt{d}\beta + \frac{d}{n}\frac{1}{\sqrt{n}}\sigma^2\beta\max_i\left[\left(\frac{\sigma_i}{\sigma_i + \lambda}\right)^2 \sigma_i^2\right] \\
&\leq \frac{2\sigma^2}{n}\sqrt{d}\beta + \frac{d}{n}\frac{1}{\sqrt{n}}\sigma^2\beta\max_i\left(\frac{\sigma_i^2}{\sigma_i + \lambda}\right)^2 \\
&\overset{\text{(i)}}{\leq} \frac{\sigma^2}{n}\beta 3\sqrt{d}.
\end{aligned}$$

where (i) follows from the assumption $d/n \leq \max_i((\sigma_i + \lambda)/\sigma_i^2)^4$, which concludes the proof of our lemma.

C.6 Proof of Proposition 1

Here, we provide the formal proof for Proposition 1.

Note that we consider the generalized ridge regression problem, but with a diagonal regularization matrix $\boldsymbol{\Lambda}$ (i.e. Tikhonov regularization). Specifically, $\boldsymbol{\Lambda}$ is the $\mathbb{R}^{d\times d}$ diagonal matrix containing regularization parameters $\sqrt{\lambda_i}$ pertaining to each different features along its diagonal.

It then directly follows from the proof of Theorem 1 in Section C.2, by simply replacing $\lambda\mathbf{I}$ with $\boldsymbol{\Lambda}^{1/2}$, that the risk for the above generalized ridge regression problem is well estimated by the following expression:

$$\bar{R}(\tilde{\boldsymbol{\theta}}_{\boldsymbol{\Lambda}}) = \sigma^2 + \sum_{i=1}^{d}\underbrace{\sigma_i^2\theta_{i,*}^2\left(\frac{\lambda_i}{\sigma_i^2 + \lambda_i}\right)^2 + \frac{\sigma^2}{n}\sigma_i^2\left(\frac{\sigma_i}{\sigma_i^2 + \lambda_i}\right)^2}_{V_i(\boldsymbol{\Lambda})}, \tag{14}$$

We consider the set of values $\{\lambda_1, \ldots, \lambda_d\}$ that minimizes the risk expression in (14). Since $\bar{R}(\tilde{\boldsymbol{\theta}}_{\boldsymbol{\Lambda}})$ contains a summation of terms pertaining to each feature, we take the derivative of $\bar{R}(\tilde{\boldsymbol{\theta}}_{\boldsymbol{\Lambda}})$ with respect to λ_i:

$$\begin{aligned}\frac{\partial}{\partial\lambda_i}\bar{R}(\tilde{\boldsymbol{\theta}}_{\boldsymbol{\Lambda}}) &= \frac{\partial}{\partial\lambda_i}\left(\sigma^2 + \sum_{j=1}^{d} V_j(\boldsymbol{\Lambda})\right)\\ &= \frac{\partial V_i(\boldsymbol{\Lambda})}{\partial\lambda_i}\\ &= 2\sigma_i^2\theta_{i,*}^2\left(\frac{\lambda_i}{\sigma_i^2+\lambda_i}\right)\frac{(\sigma_i^2+\lambda_i)-\lambda_i}{(\sigma_i^2+\lambda_i)^2} - 2\frac{\sigma^2}{n}\sigma_i^2\left(\frac{\sigma_i}{\sigma_i^2+\lambda_i}\right)\frac{\sigma_i}{(\sigma_i^2+\lambda_i)^2}\\ &= \frac{2\sigma_i^4\theta_{i,*}^2\lambda_i - 2\sigma^2\sigma_i^4/n}{(\sigma_i^2+\lambda_i)^3}.\end{aligned}$$

Setting it above to 0, we get

$$\lambda_i = \frac{\sigma^2}{n}\theta_{i,*}^{-2}. \tag{15}$$

Plugging this back into the expression at (14), we get the risk at the optimal scaling as

$$\begin{aligned}\bar{R}(\tilde{\boldsymbol{\theta}}_{\boldsymbol{\Lambda}_{opt}}) &= \sigma^2 + \sum_{i=1}^{d}\sigma_i^2\theta_{i,*}^2\frac{\sigma^4}{n^2}\theta_{i,*}^{-4}\left(\frac{1}{\sigma_i^2+\frac{\sigma^2}{n}\theta_{i,*}^{-2}}\right)^2 + \frac{\sigma^2}{n}\sigma_i^2\left(\frac{\sigma_i}{\sigma_i^2+\frac{\sigma^2}{n}\theta_{i,*}^{-2}}\right)^2\\ &= \sigma^2 + \sum_{i=1}^{d}\frac{\sigma^2}{n}\sigma_i^2\left(\frac{\sigma_i}{\sigma_i^2+\frac{\sigma^2}{n}\theta_{i,*}^{-2}}\right)^2(\frac{\sigma^2}{n}\theta_{i,*}^{-2}+\sigma_i^2)\\ &= \sigma^2 + \frac{\sigma^2}{n}\sum_{i=1}^{d}\frac{\sigma_i^2}{\sigma_i^2+\frac{\sigma^2}{n}\theta_{i,*}^{-2}}.\end{aligned}$$

C.7 Proof of Proposition 2

Proof of Proposition 2 follows directly by equating the terms in the summation of the risk expression given in (8) for the generalized ridge regression problem and the risk expression of the early-stopped least squares given in (5), as studied in Heckel and Yilmaz [HY21].

It is straightforward to see that the terms inside the respective summations become equal when λ_i are chosen as $\lambda_i = \frac{\sigma_i^2}{1-(1-\eta_i\sigma_i^2)^t} - \sigma_i^2$.

D Details of how double descent occurs outside the linear regime in neural networks

In this section, we discuss in more detail how the individual parameters of a network with p many parameters trained by applying gradient descent with stepsize η to the ℓ_2-regularized least-squares loss with regularization strength λ change across gradient descent iterations.

Note that for an overparameterized network, the network Jacobian $\mathbf{J} \in \mathbb{R}^{n\times p}$ is a wide matrix that typically has full row rank (albeit the small singular values can be very small). Let $\mathbf{J} = \mathbf{U}\boldsymbol{\Sigma}\mathbf{V}^T$

be the singular value decomposition of the Jacobian, where $\mathbf{V} \in \mathbb{R}^{p \times n}$ are the right-singular vectors. Note that only the directions of the parameter vector $\boldsymbol{\theta} \in \mathbb{R}^p$ that align with the right-singular vectors $\mathbf{V}$ impact the predictions of the linear model of the network, however the parameter vector also changes in the directions of the orthogonal complement of the right singular vectors, denoted by $\mathbf{V}_\perp \in \mathbb{R}^{p \times (p-n)}$, due to the ℓ_2-penalty. Specifically, with $\tilde{\mathbf{V}}^T = [\mathbf{V}^T, \mathbf{V}_\perp^T]$, the parameter update $\boldsymbol{\theta}_t$ at gradient iteration t takes the form

$$\begin{aligned}\boldsymbol{\theta}_t &= \tilde{\mathbf{V}} \left(\mathbf{I} - \eta \begin{bmatrix} \boldsymbol{\Sigma}^2 + \lambda \mathbf{I} & 0 \\ 0 & \lambda \mathbf{I} \end{bmatrix} \right)^t \tilde{\mathbf{V}}^T \boldsymbol{\theta}_0 + \eta \sum_{\tau=0}^{t-1} \tilde{\mathbf{V}} \left(\begin{bmatrix} \boldsymbol{\Sigma}^2 + \lambda \mathbf{I} & 0 \\ 0 & \lambda \mathbf{I} \end{bmatrix} \right)^\tau \tilde{\mathbf{V}}^T \mathbf{J}^T \mathbf{y} \\ &= \tilde{\mathbf{V}} \left(\mathbf{I} - \eta \begin{bmatrix} \boldsymbol{\Sigma}^2 + \lambda \mathbf{I} & 0 \\ 0 & \lambda \mathbf{I} \end{bmatrix} \right)^t \tilde{\mathbf{V}}^T \boldsymbol{\theta}_0 + \mathbf{V} \mathrm{diag}(\ldots, \frac{\sigma_i}{\sigma_i^2 + \lambda}(1 - (1 - \eta(\sigma_i^2 + \lambda))^t), \ldots) \mathbf{U}^T \mathbf{y}\end{aligned}$$

Then, the norm of the change in the parameters that is relevant to fitting the data is

$$\left\| \mathbf{V}^T (\boldsymbol{\theta}_t - \boldsymbol{\theta}_0) \right\|_2^2 = \sum_i^n (1 - (1 - \eta(\sigma_i^2 + \lambda))^t)^2 \left(-\frac{1}{\sigma_i} \langle \mathbf{u}_i, \mathbf{J}\theta_0 \rangle + \frac{\sigma_i}{\sigma_i^2 + \lambda} \langle \mathbf{u}_i, \mathbf{y} \rangle \right)^2 . \qquad (16)$$

Note that the convergence rate for the above depends primarily on the smallest singular value $\sigma_{\min}$. For a sufficiently small stepsize, we have $(1 - \eta(\sigma_i^2 + \lambda))^t \approx \exp(-\eta t(\sigma_i^2 + \lambda))$, which means that this part converges when $\exp(-\eta t(\sigma_{\min}^2 + \lambda))$ gets close to zero. This is the part that is relevant to fitting the data and if initialized appropriately, this change is not more than $O(n)$.

We next consider the change of the coefficient vector that is not relevant to fitting the training data:

$$\begin{aligned}\left\| \mathbf{V}_\perp^T (\boldsymbol{\theta}_t - \boldsymbol{\theta}_0) \right\|_2^2 &= (1 - (1 - \eta\lambda)^t)^2 \left\| \mathbf{V}_\perp^T \boldsymbol{\theta}_0 \right\|_2^2 \\ &\approx (1 - e^{-\eta\lambda t})^2 O(p).\end{aligned} \qquad (17)$$

Therefore, the change in the coefficients for any λ is on the order of p, and hence is not contained within a small radius around the initialization, where the NTK approximation accurately captures the dynamics of the nonlinear network, unless $1 - e^{-\eta\lambda t}$ is very small (see Figure 7 (left) for an illustration).

In order to observe how this translates to the relationship between the smallest singular value of the network Jacobian $\sigma_{\min}$, and λ, consider the following assumption on $1 - e^{-\eta\lambda t}$ being sufficiently small as parameterized by a small number δ, i.e. $1 - e^{-\eta\lambda t} \leq \delta$. We then have $\lambda \leq \frac{-1}{\eta t} \ln(1-\delta) \approx \frac{\delta}{\eta t}$. Note that we are also interested in the training regime until the network is close to convergence. This occurs when $\exp(-\eta t(\sigma_{\min}^2 + \lambda)) \approx 0$ or $\exp(-\eta t(\sigma_{\min}^2 + \lambda)) \leq \epsilon$ for small ϵ. This in turn leads to the condition $\sigma_{\min}^2 \geq \frac{1/\epsilon - \delta}{\eta t}$.

Based on these conditions on the $\sigma_{\min}$ and λ, in order for the change in the parameters to be confined in a small radius around the network initialization, we need $\sigma_{\min}^2 \gg \lambda$. Based on our empirical observations, in the regime where double descent is observed, λ is much greater than $\sigma_{\min}^2$ and the above condition does not hold.

While in this section we study how the parameters of a network change throughout the training for any λ with respect to a fixed kernel, a similar result was shown for how the associated neural tangent kernel changes across gradient flow time t (iterations) with respect to λ (see [LG20, Theorem 1]). Specifically, Lewkowycz and Gur-Ari [LG20] have shown that, when gradient flow is applied

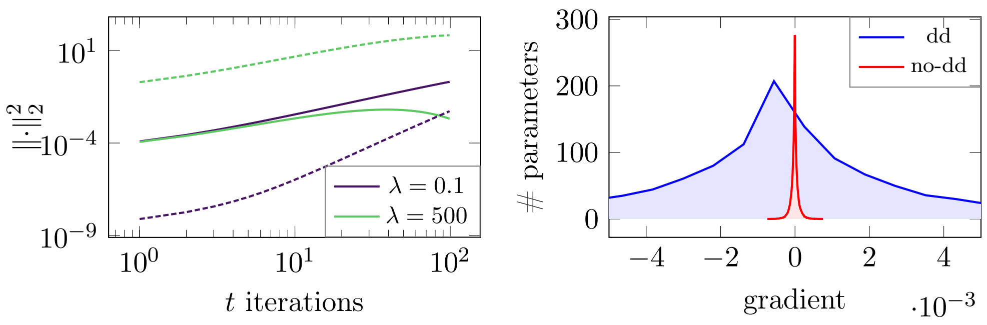


Figure 7: **Left:** The norm of the change in the parameters that is relevant to fitting the data (*solid*) and not relevant to fitting the data (*dashed*) for large and small values of λ. The results show that the parameters primarily change in the directions that are not relevant for fitting the data when λ becomes larger. This moves the neural network outside of the NTK regime (see SM D for details). **Right:** Distribution of the gradients corresponding to the first layer parameters of the network at the first gradient iteration ($t = 1$) for $\lambda = 0.001$. The red curve (scaled back $\sim$3 times for the sake of visualization) corresponds to the data setup where the difference in the scales of the data features is suppressed, hence resulting in no double descent behavior. The blue curve corresponds to the setting where the features are scaled as discussed before with double descent present as a function of the regularization strength. The results indicate that the dynamics of the network is different from the very beginning for the two regimes even for small λ.

to the ℓ_2-regularized MSE loss, the singular values of the kernel decay exponentially from the initialization with respect to λt, whereas the singular vectors remain static. This is in agreement with our discussion that $\sigma_{\min}^2 \gg \lambda$ is needed for a fixed kernel at initialization to accurately capture the training dynamics of the non-linear network throughout the course of the gradient descent.

Lastly, we show that even for small λ, the linearization (or NTK approximation) is not a good approximation for the network in a setup where regularization-wise double descent occurs. Specifically, when the disparity between the variances across the features of the data is sufficiently large to yield double descent, the change in the parameters of the network is large even for small λ. This can be seen in Figure 7 (right) for a two layer neural network. As indicated by the blue curve here, in the setting where the underlying data structure has differently scaled features and double descent is observed, the parameters change significantly from the initialization early on during the training even at smaller regularization strength. Note that, based on the decay of the kernel, this is not projected to occur until $t \sim 10^3$ for $\lambda = 0.001$ given in this example.