Double Descent in Linear Regression

Nina Plotko

May 5 2022

1 Introduction

This literature review seeks to understand how the double descent curve can be seen when training a linear regression. The papers I will reviewing are those of Nakkiran (I) and Hastie, et al. (II). I will use the information in these papers to draw conclusions about the nature of the double descent phenomenon and bias-variance tradeoff. I will also replicate an experiment by Nakkiran and conduct my own experiment.

1.1 Outline

In sections 2-4, I will setup some preliminary concepts and terminology to demonstrate my understanding of the phenomenon. In section 5, I will describe, in some detail, how the experiments in the papers were setup, and how I will use this knowledge to setup my own experiment. In this section, I will also provide the results of my experiments. In section 6, I will summarize key findings and takeaways as it pertains to double descent in linear regression.

1.2 Scope of Review

In the papers I will review, specifically in (II), the author describes experiments in the misspecified linear regression case. My goal is to first understand double descent in the simplest case to then perform an experiment. I will focus on case of isotropic features, as this geometry between the weight vector and the features' covariance matrix leads to the simplest calculations. I will, then, make adjustments to this simplest case to see how the descent curve changes. I will also look at ridge regularization as an examination of how explicit regularization affects interpolation.

2 What is Double Descent?

First observed in deep neural networks, double descent is the surprising phenomenon in which training a complex model to interpolation (zero training error) results in the test error curve—plotted over model complexity—exhibiting a second descent. In the classic statistics regime, training to interpolation results in overfitting. Thus, the model cannot generalize well and will perform worse on test data points. This follows the classic assumption that complex models are worse. As it turns out, the U-shaped curve of the bias-variance tradeoff only occurs in underparameterized models. The highest values of the first ascent in the U-shaped curve is called the *interpolation threshold*, where the number of parameters equals the number of training data points. This is the peak of test error. What is so interesting about double descent is that the overparameterized models generalize surprisingly well. Past the interpolation threshold, there exists a large number of models that can interpolate the data, so the goal in training is picking the best model. Further, it's been found that the global minimum of test error can be found in the overparameterized regime, where the test error curve monotonically decreases as the number of parameters increases.

Because the double descent phenomemon is so novel, researchers in this field have done many experiments to visualize the double descent curve under a number of settings. These adjustments include, but are not limited to, adding new training samples, adjusting the noise, and applying different types of regularization.

3 Connection to Linear Regression

In a well-conditioned linear regression model, the least-squares objective has a unique minimizer given by:

$$\hat{\beta} = (X^T X)^{-1} X y$$

When the number of parameters exceeds the number of training data points ($d > n$), the least-squares objective does not have a unique minimizer. This is because we cannot take the inverse of $X^T X$. The main solution to this problem is running gradient descent on the objective (corresponds to min-$\ell 2$ norm):

$$\min_{\hat{\beta}} ||X\hat{\beta} - y||_2^2$$

The solution found by gradient descent is $\hat{\beta} = X^\dagger y$, where $X^\dagger$ denotes the pseudo-inverse of the matrix X. This takes two forms depending on the ratio of parameters to number of training examples. Since X is full rank in our problem setup (detailed in section 5), this solution has many possible values that interpolate the training data. This corresponds to:

$$\hat{\beta} = X^\dagger y = \begin{cases} \underset{\beta : X\beta = y}{\operatorname{argmin}} ||\beta||_2^2 & \text{when } n \leq d \\ \underset{\beta}{\operatorname{argmin}} ||X\beta - y||_2^2 & \text{when } n > d \end{cases}$$

4 Preliminary Concepts to Understand

I will now go through a number of concepts that determine the calculations I will describe in section 5. These concepts have been found to have the largest effect on the interpolation threshold and the double descent curve, so these are the qualities to be adjusted in the experiments that follow.

Signal to Noise Ratio

Signal to noise ratio (SNR) is simply the ratio of the model's signal to the noise's signal. Hastie, et al.(II) describes SNR as:

$$SNR = \frac{||\beta||_2^2}{\sigma^2}$$

In the following experiments, I hold the $\ell 2$ norm of the weights equal to 1, so I vary the SNR values by varying the standard deviation of the noise. Higher values of SNR point to lower irreducible errors.

Overparameterized Ratio

The overparameterized ratio γ is the number of parameters over the number of training data points (d/n). This is the factor that is being constantly adjusted in all the following experiments. In the first experiment (5.1), we vary the sample size. In the other experiments, we vary the number of parameters.

Isotropic Features

Hastie, et al.(II) examine three cases to understand overparameterized models: isotropic features, anisotropic features (specifically latent space features), and a nonlinear model. For the coming experiments, I will focus on isotropic features. The authors examine these three cases because the geometry between the covariance matrix and the weight vector affects the generalization error curve.

Isotropic features are those in which the covariance matrix Σ is equal to the identity matrix I_d. This is the simplest case where the risk is only dependent on β through $||\beta||_2^2$ (II).

5 Experiments

In these experiments, I will choose a setting to vary and either replicate the experiment discussed in the paper or perform my own experiment.

5.1 Varying the Number of Samples for a Fixed Number of Parameters

Nakkiran (I) proposed the following setup:

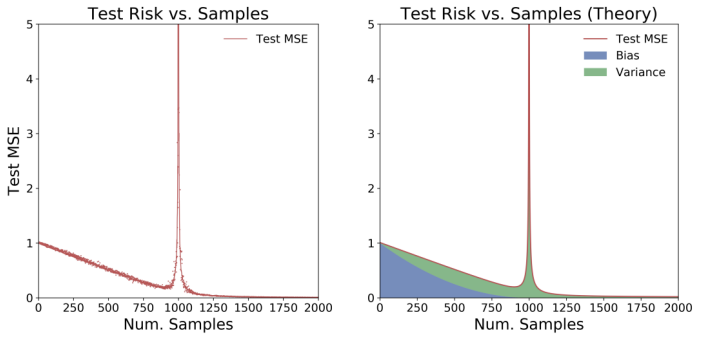
We will start with a fixed number of parameters (d = 1000) and vary the size of training data matrix:

$$\text{Covariates: } X \sim N(0, I_d)$$
$$\text{Response: } y = \langle x, \beta \rangle + N(0, \sigma^2)$$

We will choose an arbitrary β such that $||\beta||_2 \leq 1$. In my experiment replication, I generated a (size d) vector of standard uniform distribution and divided it by its norm to ensure $||\beta||_2 = 1$. I generated a vector of various values of n. For each value of n, I ran 50 simulations where I generated X, y as described above and ran linear regression in two ways:

1. If $n < d$, I computed $\hat{\beta} = X^{\dagger} y$.
2. If $n > d$, I computed closed form least-squares solution $\hat{\beta} = (X^T X)^{-1} X^T y$

Results



(a) Test MSE for $d = 1000, \sigma = 0.1$. (b) Test MSE in theory for $d = 1000, \sigma = 0.1$

Figure 1: Preetum Nakkiran's Results

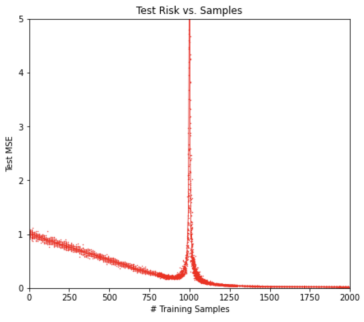


Figure 2: Replicated Results

I was able to replicate Nakkiran's results almost perfectly. Next, I varied some settings to see how the curve would change. I adjusted the sample size to 500 samples and ran the same experiments. I mainly wanted to see how the noise affected the shape of the curve and how the different dimension affected the curve. I tested the following settings:

1. SNR = 100 with $\sigma = 0.1$ (Original Setting)
2. SNR = 1 with $\sigma = 1$

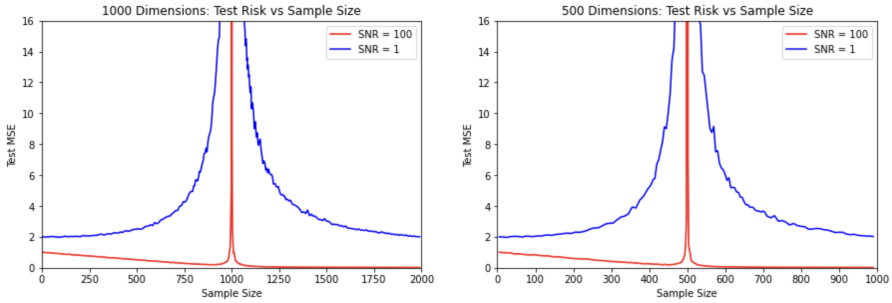


Figure 3: Overall Results

The left image of Figure 3 depicts the experiment under 1000 training samples, and the right image of Figure 3 depicts the experiment under 500 samples. Interestingly enough, the number of dimensions did not affect the curve very much. The difference between the red curves for the two fixed dimension sizes are very similar. This makes sense, as double descent is normally a function of γ, and I ran the test for the same values of $\gamma \in \{0, 2\}$. The different setting of the SNR formed a much wider curve. We can also see that the test risk on either side of the interpolation threshold looks very similar. As previously mentioned, we can see the irreducible error is much smaller for high SNR values (as in the original settings of the experiment). The code for this experiment is under "Experiment1_VaryDim.ipynb" in the github.

5.2 Varying the Number of Parameters for a Fixed Number of Samples

Hastie, et al. (II) proposed that the double descent curve looks different depending on the setting of SNR. In the above experiment, we set the standard deviation for noise to be $\sigma = 0.1$. Further, the SNR in this setting was 100 (from the fact that $||\beta||_2 = 1$). The findings in the paper suggest two interesting facts about the double descent curve:

1. For well-specified linear regression models, the global minimum for risk occurs in the underparameterized regime. If SNR ≤ 1, we can observe a monotonic decrease of risk in the overparameterized regime. If SNR > 1, as in the above experiment, the risk has a local minimum in the overparameterized regime.

2. For misspecified linear regression models, "when SNR > 1, the risk can attain its *global* minimum in the overparameterized regime." (II)

My goal in the following experiments is to recreate these findings with my own experiments. I will use a similar setup approach to the first experiment, using a Gaussian data matrix.

Well-Specified Model

There are two factors I will vary for the well-specified model to replicate the findings of paper (II). First, I will vary the number of parameters. Next, I will vary the SNR by changing σ, so I test values of SNR $\in \{100, 4, 1\}$. For these experiments, I test values of $n \in \{200, 500, 1000\}$.

Results

The results for this experiment are shown in Figure 4. The leftmost image shows the test risk for 200 samples, the middle image shows the results for 500 samples, and the rightmost image shows the results for 1000 samples. As in the first experiment, changing the number of samples for the experiment does not have much effect on the curve, as the curve is a function of the overparameterized ratio, γ. We can observe a local minimum in the overparameterized regime for $SNR = 100$ and begins to increase for higher dimensions. It is unclear for $SNR = 4$ whether this is a local minimum, but the test risk does not seem to change past a ratio of $\gamma = 1.5$. The code for this experiment is under "Experiment2_VarySamples.ipynb" in the github.

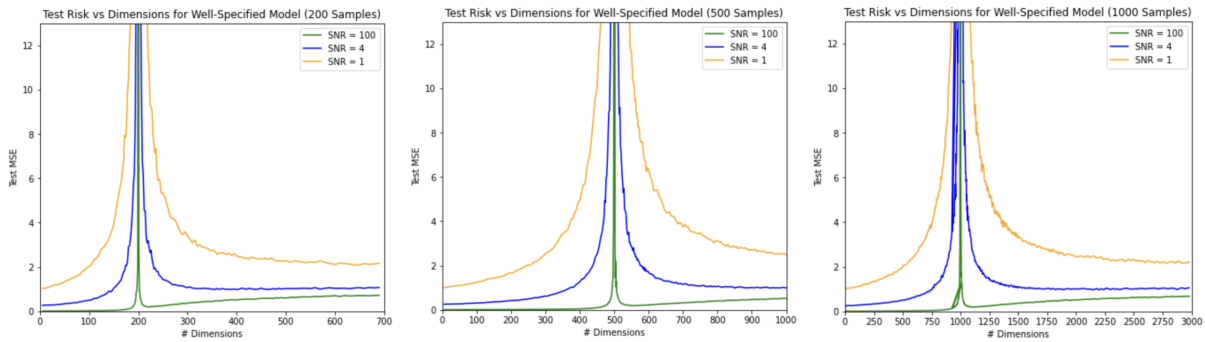


Figure 4: Well Specified Models

Misspecified Model

In the paper (II), the authors specify the data model as such:

$$\begin{aligned} ((x_i, w_i), \epsilon_i) &\sim P_{x,w} \times P_\epsilon, \qquad i = 1, ..., n \\ y_i &= x_i^T \beta + w_i^T \theta + \epsilon_i, \qquad i = 1, ..., n \end{aligned}$$

Then, they collect the features in a matrix $[XW] \in \mathbb{R}^{n \times (d+q)}$, but they fit $\hat{\beta}$ using X, y, not W. I will perform simpler versions:

Training on a dataset missing parameters I add 25 percent to the number of parameters and train with the original number of parameters. The model looks like:

$$\begin{aligned} q &= \text{int}(d * 1.25) \\ X &\sim N(0, I_q) \\ y_i &= x_i^T \beta + \epsilon_i, \qquad i = 1, ..., n \end{aligned}$$

When training, I only give the matrix $X' \in \mathbb{R}^{n \times d}$

Training on a dataset missing parameters and adding irrelevant parameters Using the same model as above, I now add another variable $W \in \mathbb{R}^{n \times p}$, where $p = 75$. I then train on the matrix $[X'W] \in \mathbb{R}^{n \times (d+p)}$

Training on anisotropic parameters For this portion of the experiment, I created a covariance matrix for each dimension tested. This covariance matrix is random, symmetric, and positive-definite. I, then, tested the model with missing parameters and irrelevant parameters.

Results

Figure 5 and Figure 6 show the misspecified models as a function of dimension and gamma. The results from missing parameters. The results from missing parameters and adding irrelevant samples looked very similar. The only difference being the shift in the interpolation threshold (caused by the missing parameters).

Figure 7 shows the results from the anisotropic misspecified models. In this case, adding irrelevant samples did affect the shape of the curve. Instead of decreasing monotonically, we can observe a local minimum in the overparameterized regime. It's also interesting to note the shift in the interpolation threshold. The number of dimensions trained with is actually the x-value of the graph plus 75 parameters. While I was not able to obtain a global minimum as in (II), it is interesting to see the effect of the extra samples. With isotropic features, adding irrelevant features did not make a difference in the curve, unlike the anisotropic features.

The code for these experiments is under "Experiment3_VaryDim_Misspecified.ipynb" in the github.

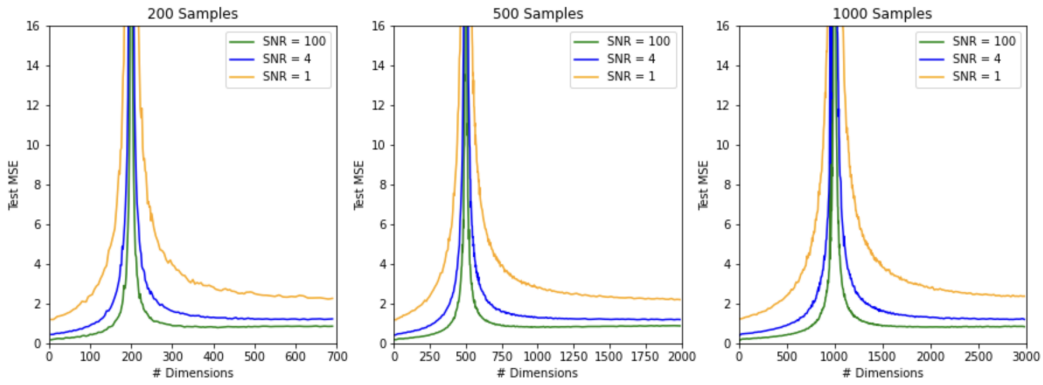


Figure 5: Misspecified Models

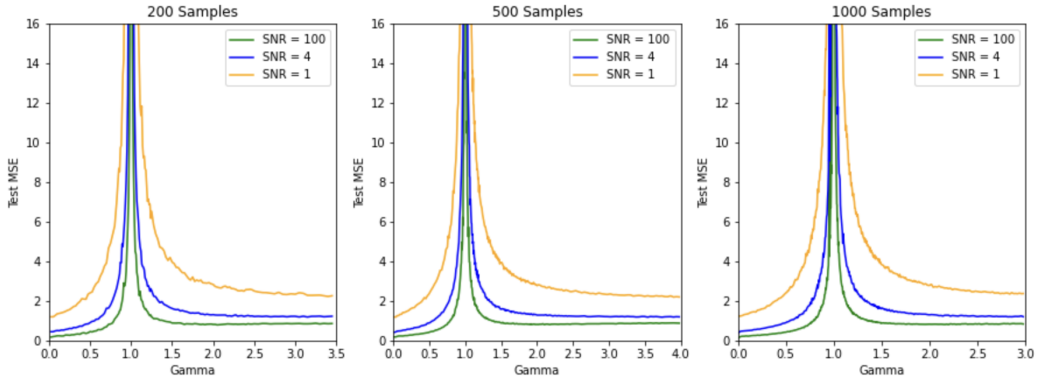
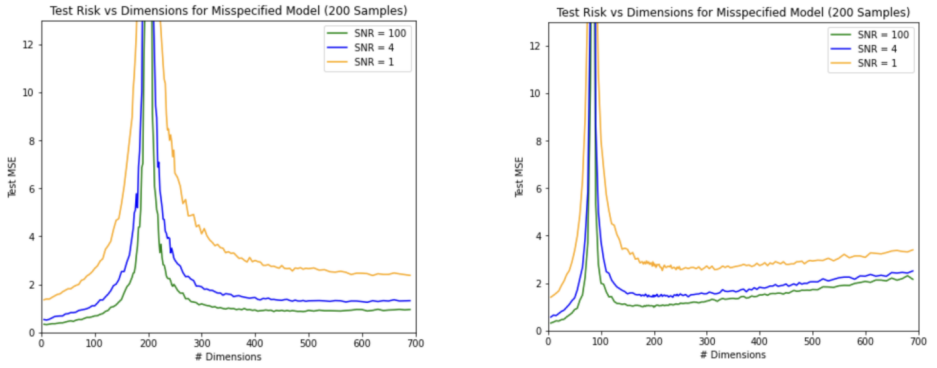


Figure 6: Misspecified Models (as a function of γ)



(a) Missing Parameters

(b) Missing Parameters and Added Irrelevant Parameters

Figure 7: Anisotropic Misspecified Models

5.2.1 Ridge Regularization

To visualize the effect of explicit regularization, I experimented with optimally tuned ridge regularization. Hastie, et al. describe regularization as hiding the double descent curve, so I try to experiment by: (1) Using cross-validation to pick the hyperparameter $\lambda \in \{1e-6, 1e-4, 1e-2, 1e-1, 1, 10\}$ for each dimension size, and (2) Varying the number of dimensions over a fixed sample size. I ran this experiment for the three values

of SNR I've tested in previous experiments. At this point, I noticed that applying the same experiment to a different number of samples does not give any different results. The curve is a function of γ, so I will manipulate gamma for only one value of n, 200 samples, as I did for the anisotropic model.

5.2.2 Results

Figure 8 shows the test risk curve when applying an optimally-tuned regularization term. For high SNR, the double descent curve is not observed. We see close to zero test error until the number of dimensions approaches 200 (or $\gamma \to 1$). When the noise signal is equal to the model's signal (SNR = 1), we do observe a descent, although its shape is much different. The code for this experiment is under "Experiment4_VarySamples_Ridge.ipynb" in the github.

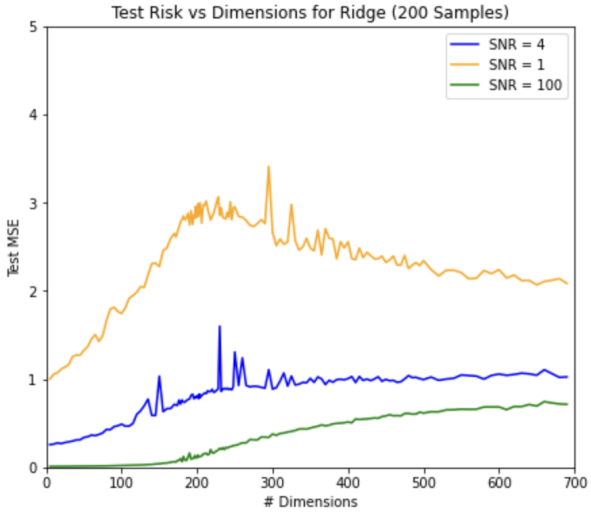


Figure 8: Ridge Regularization

6 Takeaways

In this section, I describe the main takeaways from articles I've read and the takeaways from the experiments I conducted.

6.1 Model Complexity

The difference of experiment 5.1 and experiment 5.2 is one that highlights the ambiguity of the term, *model complexity*. The first experiment flipped the problem on the side and viewed model complexity as a function of the training data size. We were still able to see the double descent curve (although in this case, the overparameterized regime occurred on the left side of the graph). It is interesting to explore the model complexity in terms of parameters, but we can also explore this term with the same number of given parameters while varying the degree of polynomial. Wilber and Werness (III) observed the effect of varying the polynomial degree. Here, the authors worked with a dataset with 36 parameters and varied the degree polynomial past 500 degrees. Even in this setting, the interpolation threshold occurred at the 36th degree. As the number of nonlinear features increased dramatically ($\gamma \to 15$), the interpolations got smoother. The only thing that changed with the addition of a nonlinear feature was how the model connected the points in the training set. In terms of deep learning algorithms, the double descent curve can also be seen for epochs, called *epoch-wise double descent*. In this case, model complexity is the training time of each epoch. I believe the double descent phenomenon is so interesting because scientists have not agreed on one specific measure for model complexity. Rather, it is a term to describe the richness of the model space.

6.2 More Data Can Hurt

In the underparameterized regime, more training samples results in higher test set accuracy. In the overparameterized regime, models actually perform better with a smaller amount of training samples. Imagine we trained two training sets with a fixed dimension size; one with the full amount of training examples, the other with three-quarters of the examples. The model trained with less samples would achieve a lower generalization error. This is a big problem within the machine learning community, as it goes against the statistical foundation that more training data attains better results.

6.3 Applications of Misspecified Models

Examining the misspecified case was a necessity to this paper because of its overarching applications. Model misspecification can come from feature correlation, irrelevant features, missing features, and the selection of an incorrect model class. In most cases of data science, we are not given the "full picture" to predict with. Cases in which we assume we are missing key information would be an opportunity for data scientist to explore overparameterized models.

6.4 The Effect of Regularization

The last experiment demonstrates the role of regularization in regression, and that is why the results obtained are expected. Regularization is a practice to prevent overfitting, or interpolating the data, so a model can generalize better. Ridge regularization is a form of explicit regularization. The dampened double descent curve from explicit regularization has been found by a number of researchers within the field. Including the experiment in the paper was to present a case in which double descent does not occur. Beyond explicit regularization, like ridge or lasso, implicit regularization also prevents overfitting. Practices like early stopping and stochastic gradient descent may also mask the double descent curve. This is a topic that many researchers are looking at now.

7 Appendix

I Nakkiran, Preetum. "More Data Can Hurt for Linear Regression: Sample-wise Double Descent." v1. 2019. https://doi.org/10.48550/arXiv.1912.07242

II Hastie, Trevor, et al. "Suprises in High-Dimensional Ridgeless Least Squares Interpolation." v5. 2020. https://doi.org/10.48550/arXiv.1903.08560

III Wilber, J.; Werness, B. "Double Descent." *MLU*, December, 2019. https://mlu-explain.github.io/double-descent/