Investigating the Double Descent Phenomenon in Machine Learning Models

Pierre Clayton
Data Science, Statistics & Learning
ENSAE Paris, Institut Polytechnique
Palaiseau, France
pierre.clayton@ensae.fr

Vincent Gimenes
Data Science, Statistics & Learning
ENSAE Paris, Institut Polytechnique
Palaiseau, France
vincent.gimenes@ensae.fr

Anna Mosaki
Finance, Risk & Data
ENSAE Paris, Institut Polytechnique
Palaiseau, France
anna.mosaki@ensae.fr

Abstract

Combining empirical risk minimization with capacity control has traditionally guided practitioners to avoid overfitting in machine learning. Yet modern deep learning relies on large overparameterized models that often perfectly fit the training data while still demonstrating strong generalization performance. This paradox has motivated research into the *double descent* phenomenon, wherein test error exhibits a second decrease once models enter an overparameterized regime. In this work, we examine double descent across three representative models: linear regression, Random Fourier Features (RFF), and classical neural networks. We provide theoretical insights, reproduce key experiments, and analyze the effect of implicit and explicit inductive biases. Our results highlight how overparameterized models can achieve surprisingly low test error, challenging the classical bias–variance trade-off.

1 Introduction

Modern machine learning often pushes well beyond the classical paradigm that prescribes balancing bias and variance to avoid overfitting. While the traditional U-shaped risk curve indicates that very large models should overfit and thus perform poorly on unseen data, practitioners observe the opposite in many deep learning scenarios: extremely large models that exactly fit the training set yet still achieve excellent generalization performance [1].

The *double descent* phenomenon has been proposed to reconcile this discrepancy by extending the classical risk curve. Once a model's capacity is large enough to interpolate the training data, test error peaks (often dramatically) but can then descend again as the capacity continues to increase [2]. This project investigates double descent for three paradigmatic settings:

- **Linear regression** in the under- and over-determined regimes.
- **Random Fourier Features (RFF)** as a kernel approximation method.
- **Classical neural networks** ranging from simple to overparameterized topologies.

2 Background and Related Work

2.1 Classical Bias–Variance Trade-Off

In the traditional view, model complexity is limited to ensure a balance between bias and variance. A model that is *too simple* underfits, showing high bias, whereas a model that is *too complex* overfits, showing high variance [3]. This trade-off typically suggests an optimal capacity that yields the lowest test error. However, this classical analysis does not account for interpolation regimes in which the model perfectly fits the training data.

2.2 Overparameterization and the Modern Practice - the Double Descent phenomenon

Recent large-scale deep learning approaches often deploy models with far more parameters than training examples, yet these models yield outstanding results on test data [1]. Empirical evidence shows that, once models pass the interpolation threshold (where training error becomes essentially zero), their test error may sharply increase and then decrease again when capacity is further increased [2]. Double descent posits that the test error curve is not strictly U-shaped in many modern settings. Instead, after the initial descent and subsequent ascent near the critical interpolation threshold, the test error can *descend once more* as the model becomes highly overparameterized. This holds for numerous families of models (linear, kernel-based, and neural networks).

2.3 Review of Relevant Literature

Foundational works on classical learning theory, e.g. [3], provide the bias–variance framework. More recent contributions focus on overparameterization [1], the role of implicit regularization in gradient-based optimization [2], and iterative methods in separating the under- and over-determined regimes [5].

3 Theoretical Foundations

This section presents the core theoretical elements underlying double descent for three model classes: linear regression, Random Fourier Features, and classical neural networks. We emphasize how certain inductive biases or training procedures can favor simpler *interpolating* solutions.

3.1 Linear Regression with Gaussian Noise: The p-Submatrix Perspective

We now turn to a refined analysis of linear regression in an overparameterized setting where we select a *random subset* of features. This viewpoint illustrates how double descent arises precisely when the number of active features (or parameters) p nears the sample size n. We present a key theorem (with a sketch of the proof) that pinpoints where the test risk can blow up and how it descends again for even larger p.

Setup. Consider $X \in \mathbb{R}^{n\times d}$ formed by n i.i.d. samples $X_i \in \mathbb{R}^d$ with $X_i \sim \mathcal{N}(0, I)$, and let $y \in \mathbb{R}^n$ be defined via

$$y_i \;=\; X_i^\top w \;+\; \sigma\,\varepsilon_i, \quad \varepsilon_i \sim \mathcal{N}(0,1),$$

where $w \in \mathbb{R}^d$ is the true parameter and $\sigma > 0$ is a noise factor. Thus in matrix form we have $y = X\,w + \sigma\,\varepsilon$.

Random Sub-matrix of Size p. To investigate how the generalization error varies with the number of active features, we select a random subset $\mathcal{T} \subset \{1, \ldots, d\}$ of cardinality p. Let $X_{\sim p}$ be the sub-matrix of X containing only the columns indexed by $\mathcal{T}$. Likewise, write $w_{\sim p}$ for the corresponding p-dimensional sub-vector of w. The remaining $q = d - p$ coordinates define $X_{\sim q}$ and $w_{\sim q}$, but these are *discarded* for the model.

We then solve the least squares problem restricted to the sub-matrix $X_{\sim p}$:

$$\hat{w}_{\sim p} \;=\; X_{\sim p}^{+}\, y, \qquad \hat{w}_{\sim q} \;=\; 0,$$

where $X_{\sim p}^{+}$ is the Moore-Penrose pseudo-inverse of $X_{\sim p}$. We embed $(\hat{w}_{\sim p}, \hat{w}_{\sim q})$ back into a full d-dimensional parameter (assigning zeros to the q unused coordinates). Denote this full vector by $\hat{w}$. The theorem below characterizes the expected test risk of the predictor $u \mapsto u^{\top}\hat{w}$ as p varies.

Theorem 1 (Double Descent Risk for Random p-Submatrix) ***From [6]*** *Let $X_i \sim \mathcal{N}(0, I_d)$ and $y_i = X_i^{\top} w + \sigma\,\varepsilon_i$ with $\varepsilon_i \sim \mathcal{N}(0,1)$ i.i.d. for $i = 1, \ldots, n$. Suppose X has rank n. Fix (p, q) such that $p + q = d$, and let $X_{\sim p} \in \mathbb{R}^{n \times p}$ be the randomly selected p-column sub-matrix of X. Define*

$$\hat{w}_{\sim p} \;=\; X_{\sim p}^{+}\, y, \quad \hat{w}_{\sim q} = 0, \quad \hat{w} = (\hat{w}_{\sim p},\, \hat{w}_{\sim q}).$$

Then the expected risk, $\mathbb{E}\big[(y - x^{\top}\hat{w})^2\big]$, has the following form:

$$\mathbb{E}\big[(y - x^{\top}\hat{w})^2\big] \;=\; \begin{cases} \Big(\|w_{\sim q}\|^2 + \sigma^2\Big)\Big(1 + \frac{p}{n-p-1}\Big), & \text{if } p \le n-2, \\ +\infty, & \text{if } n-1 \le p \le n+1, \\ \|w_{\sim p}\|^2\Big(1 - \frac{n}{p}\Big) \;+\; \Big(\|w_{\sim q}\|^2 + \sigma^2\Big)\Big(1 + \frac{n}{p-n-1}\Big), & \text{if } p \ge n+2. \end{cases}$$

Interpretation.

- **When** $p < n$: The risk evolves smoothly, reminiscent of a classical bias–variance trade-off. Increasing p shrinks the bias but might moderately raise variance.
- **When** $p \approx n$: The risk may blow up to $+\infty$ in the limit, illustrating that near the *interpolation threshold*, small singular values of $X_{\sim p}$ make the solution norm large and drastically worsen generalization.
- **When** $p > n$: The model enters an *overparameterized regime*, with enough degrees of freedom to interpolate more "smoothly." The risk *decreases* again, demonstrating the second descent in the double descent curve.

These three regimes align with the double descent phenomenon: a "U-shape" for small p, a sharp peak near $p = n$, and then a further drop once p surpasses n by a comfortable margin.

Sketch of Proof. (For a complete argument, see [6])

1. **Rewrite the expected risk.** Let $x \sim \mathcal{N}(0, I_d)$ be an independent test point and note
$$\mathbb{E}[(y - x^{\top}\hat{w})^2] \;=\; \mathbb{E}\big[(x^{\top}(w - \hat{w}) + \sigma\,\varepsilon)^2\big] \;=\; \sigma^2 \;+\; \mathbb{E}\big[\|\,w - \hat{w}\|^2\big],$$
because x and ε are independent zero-mean Gaussians. Furthermore, $\hat{w}_{\sim q} = 0$ implies $\|w - \hat{w}\|^2 = \|w_{\sim p} - \hat{w}_{\sim p}\|^2 + \|w_{\sim q}\|^2$.
2. **Analyze** $\|w_{\sim p} - \hat{w}_{\sim p}\|^2$**.** Decompose:
$$\hat{w}_{\sim p} \;-\; w_{\sim p} \;=\; w_{\sim p} \;-\; X_{\sim p}^{+}\, y \;=\; \big(I - X_{\sim p}^{+} X_{\sim p}\big)\, w_{\sim p} \;-\; X_{\sim p}^{+}\,\eta,$$
where $\eta = y - X_{\sim p}\, w_{\sim p}$. One observes:
   - $I - X_{\sim p}^{+} X_{\sim p}$ is the orthogonal projection onto the null space $\mathrm{Ker}(X_{\sim p})$.
   - $X_{\sim p}^{+}$ itself is a linear operator from $\mathbb{R}^n$ to $\mathbb{R}^p$, mapping vectors into $\mathrm{Im}(X_{\sim p}^{+})$.

   Consequently, the two terms are orthogonal in $\mathbb{R}^p$, allowing a Pythagorean theorem argument to split their squared norms. The calculations at this step use the "trace trick" and the notion of distribution of inverse-Wishart for pseudo-inverse matrices but is beyond our understanding, unfortunately.
3. **Interpretation of the infinite risk case.** The "$+\infty$" arises as p nears n because singular values of $X_{\sim p}$ can collapse, forcing large coefficients in $\hat{w}_{\sim p}$. As a result, the variance term skyrockets, pushing the test risk arbitrarily high in the limit. Once $p > n$, these collapsed singular modes vanish more gently due to additional degrees of freedom.

Overall, this analysis highlights that a random selection of p features exhibits a double descent risk profile. The threshold $p \approx n$ is particularly precarious, showing how interpolation may degrade generalization, only for the risk to decline again when p grows far beyond n. Such behavior is consistent with broader double descent observations in neural networks and other overparameterized models.

3.2 Random Fourier Features: A Bridge between Linear Models and Gaussian Kernels

Random Fourier Features (RFF) provide a particularly illustrative framework for understanding the double descent phenomenon. They connect ideas from *linear regression* (since they are linear in their learnable parameters) and *kernel methods* (since they approximate a non-linear kernel). In this section, we describe how varying the dimension N of the random feature map can transition from an under-parameterized to an over-parameterized regime, reproducing the characteristic double descent curve. We then present a key theorem, originally from [2], that formalizes how certain *minimum-norm* interpolating solutions can still generalize well even when interpolating perfectly.

Definition and Motivation. An RFF model introduces a map $z : \mathbb{R}^d \to \mathbb{R}^N$ such that each input $x \in \mathbb{R}^d$ is transformed via randomized cosine features. In the case of approximating a Gaussian kernel, one typically sets:

$$z(x) \;=\; \sqrt{\tfrac{2}{N}} \begin{bmatrix} \cos(\omega_1^\top x + b_1) \\ \vdots \\ \cos(\omega_N^\top x + b_N) \end{bmatrix},$$

where $\omega_i \sim \mathcal{N}(0, \sigma^2 I_d)$ and b_i is sampled uniformly in $[0, 2\pi]$. A model in the RFF family is then

$$h_\beta(x) \;=\; \beta^\top z(x),$$

where $\beta \in \mathbb{R}^N$ contains the **learnable** parameters. Observe that $h_\beta(x)$ is **linear in** β yet **non-linear in** x, which is crucial: it links linear methods and kernel methods.

From Linear Regression to Kernel Methods. The random map $z(x)$ is designed to approximate the Gaussian kernel $k(x, y) = \exp\big(-\|x - y\|^2/(2\sigma^2)\big)$. Indeed, from harmonic analysis, one knows $k(x, y) = \mathbb{E}_\omega[\exp(i\,\omega^\top(x - y))]$. Restricting ω to $\mathcal{N}(0, \sigma^2 I_d)$ and taking the real part leads to a random projection via cosines. By choosing N large, one obtains an increasingly accurate approximation of k. Training a model $\beta^\top z(x)$ is then essentially a *linear* problem in β that approximates a Gaussian kernel machine at a fraction of the usual cost, especially when $N \ll n$.

Dimension N as Model Capacity. In an RFF model, the number of features N naturally controls the expressiveness:

$$\mathcal{H}_N \;\subset\; \mathcal{H}_{N+1}, \quad \text{for every } N \geq 0.$$

When $N \ll n$, the model can be under-parameterized (not enough features to capture all training variations), leading to a classical bias–variance trade-off. As N grows toward n, the model reaches an *interpolation threshold*, beyond which the training set can be perfectly fitted. Around this threshold, test error can spike (a hallmark of double descent). Finally, once N becomes significantly larger than n, the model enters an over-parameterized regime, allowing for many interpolating solutions. A bias toward *small norm* solutions (e.g., via gradient descent initialization or explicit constraints) can lead to surprisingly good generalization, driving the test error back down.

Analogy to Linear Regression and Double Descent. This picture parallels the analysis in Section 3.1. Near $N \approx n$, the RFF feature matrix has singular directions prone to "blow-up," creating a high-variance fit. But for $N \gg n$, a minimum-norm (or similar) solution β^* can restrict coefficient magnitudes and yield good test performance. This produces a double descent curve:

- **Under-parameterized** ($N < n$): Good old bias–variance trade-off.
- **Critical threshold** ($N \approx n$): Potentially large spike in test error.
- **Over-parameterized** ($N > n$): Multiple perfect fits exist, but an *implicit regularization* can favor low-norm solutions with strong generalization.

A Theorem on Interpolating Solutions. In a noiseless setting, or when σ is small, one can show that such *minimum norm* solutions in a large feature space often maintain good generalization. A rigorous example is the following theorem adapted from [2] (Theorem 22), which shows that any *interpolating* solution in a Reproducing Kernel Hilbert Space (RKHS) for the Gaussian kernel can remain close to the true function h^* over the entire input domain. Although the result is stated for $\mathcal{H}_\infty$ (the infinite-dimensional kernel space), finite but large N approximations via RFF inherit similar behavior.

Theorem 2 (From [2]) *Fix any target function $h^* \in \mathcal{H}_\infty$. Let $\{(X_i, Y_i)\}_{i=1}^n$ be i.i.d. random variables, where X_i is drawn uniformly at random from a compact cube $\Omega \subset \mathbb{R}^d$, and $Y_i = h^*(X_i)$ for all i. Then there exist positive constants $A, B > 0$ such that, for any interpolating $h \in \mathcal{H}_\infty$ (i.e. $h(X_i) = Y_i$ for all i), with high probability (over the sampling of X_i):*

$$\sup_{x \in \Omega} \big| h(x) \; - \; h^*(x) \big| \; < \; A \, \exp\Big[- B \, \big(\tfrac{n}{\log n} \big)^{\frac{1}{d}} \Big] \, \Big(\, \|h^*\|_{\mathcal{H}_\infty} + \|h\|_{\mathcal{H}_\infty} \Big).$$

Interpretation. Even though h *interpolates* the training data (thus fitting perfectly), this result guarantees that if $\|h\|_{\mathcal{H}_\infty}$ is not too large (i.e., h is a relatively small-norm function in the Gaussian RKHS), then h remains close to h^* uniformly on Ω, with high probability. This corroborates the idea that *small norm* or *smooth* interpolants, which can arise by implicit or explicit bias, do not necessarily overfit.

Proof of Theorem 2 (Key steps). We outline the main steps to clarify why any interpolating h in $\mathcal{H}_\infty$ must be a good approximation of h^*, provided h has limited norm and n is large.

1. **Setup and Notation.** Since X_i are drawn uniformly from the compact domain Ω, define $h^* \in \mathcal{H}_\infty$ as given. An interpolating function $h \in \mathcal{H}_\infty$ satisfies $h(X_i) = Y_i = h^*(X_i)$ for all i.
2. **Decomposition of the Error $\sup_{x\in\Omega} |h(x) - h^*(x)|$.** We want to control $\|h - h^*\|_\infty$ over Ω. In the Gaussian RKHS $\mathcal{H}_\infty$, one typically uses reproducing-kernel properties (i.e., $|f(x)| \le \|f\|_{\mathcal{H}_\infty} \, \|\kappa_x\|_{\mathcal{H}_\infty}$) and suitable covering-number bounds for Ω.
3. **Covering Argument on Ω.** One constructs an ϵ-net on Ω whose size grows like $(\frac{1}{\epsilon})^d$ (up to constants). Using standard arguments in kernel methods, with high probability over sampling $\{X_i\}_{i=1}^n$, the function h that interpolates X_i can be shown to remain close to h^* at all points in Ω, especially if $\|h - h^*\|_{\mathcal{H}_\infty}$ is controlled.
4. **Bounding the Norm $\|h\|_{\mathcal{H}_\infty}$.** Because h and h^* coincide on the n data points X_i, a concentration argument implies that if $\|h\|_{\mathcal{H}_\infty}$ were huge, we would see large fluctuations that contradict interpolation or small variance claims. More precisely, an argument involving the kernel matrix $[k(X_i, X_j)]_{i,j}$ ensures that $\|h\|_{\mathcal{H}_\infty}$ must remain of the same order as $\|h^*\|_{\mathcal{H}_\infty}$ plus lower-order terms in n.
5. **Exponential Decay with Respect to $\big(n/\log n\big)^{1/d}$.** Kernel approximation theorems show that with n samples, the maximum deviation $\sup_{x\in\Omega} |h(x) - h^*(x)|$ can be forced down by a factor roughly $\exp\big[-B\,(n/\log n)^{1/d}\big]$, due to dimension-$d$ geometric factors and kernel concentration. The exact exponent $1/d$ comes from the volume/covering dimension, while the $\log n$ factor arises in bounding certain tail probabilities or condition-number constraints in the kernel matrix.
6. **Combine All Bounds to Conclude.** Putting these steps together yields the precise inequality from Theorem 2: with high probability, any h interpolating the n samples (and not exploding in norm) remains ϵ-close (in sup norm) to h^*. The constant A encapsulates geometric and kernel-dependent factors, and B emerges from concentration rates on Ω. The factor $\|h\|_{\mathcal{H}_\infty} + \|h^*\|_{\mathcal{H}_\infty}$ reflects how the norms of the interpolating function and the target function control the uniform bound.

This theorem demonstrates a key theoretical principle: even though interpolating solutions may seem to "overfit," in high-capacity spaces with *smoothness* or *small norm* biases, they can stay close to the true target function on the entire domain.

Concluding Remarks on RFF Double Descent. By connecting this result to a finite-dimensional RFF approximation ($\mathcal{H}_N$), we see that as $N \to \infty$, the class of RFF models approaches $\mathcal{H}_\infty$. Hence, for large N, the same principle holds: among the many interpolating solutions, those of small norm can still generalize well. Conversely, near $N \approx n$, ill-conditioning can produce a spike in test error, echoing the exact pattern we observe in double descent.

Hence, RFFs show a direct path from under-parameterized regime (small N) to over-parameterized regime (large N), with a peak at the interpolation threshold. Thanks to implicit or explicit norm-based

bias, the model "resumes" good generalization once N is large enough, mirroring the phenomenon analyzed in linear regression and other over-parameterized settings.

4 Methodology and Implementation

We designed experiments to test the presence of double descent in many settings: linear regression, RFF models, classical fully connected neural networks and tree based models. Our code is available in a public repository[1].

4.1 Dataset Description and Preprocessing

MNIST. We primarily focus on the MNIST dataset (handwritten digits), using a reduced training set size to emphasize under- vs. over-parameterization. Pixel intensities are normalized to $[0, 1]$ and flattened.

Cifar-10. CIFAR-10 is a dataset of 60,000 32x32 color images across 10 classes. We use a reduced training set of 1000 samples to highlight under- and over-parameterization. Pixel intensities are normalized to [0, 1], and labels are one-hot encoded for classification.

Taxi Trip Pricing dataset : We focus on the Taxi Trip Pricing dataset, a structured dataset with features such as trip duration, distance, and pickup/dropoff locations. A reduced training set size emphasizes under- vs. over-parameterization. The categorical variables are encoded as a hot number to facilitate model training.

Additional Data. In more extensive tests, we incorporate synthetic datasets to illustrate the phenomenon across varying complexity.

4.2 Model Architectures

Linear Regression We implemented classical linear regression with an increasing of the complexity by the increasing of the number of parameters. We thus added polynomial features.

Random Fourier Features We generate RFF expansions of dimension N, then apply a linear model on top:

$$\hat{y} = \beta^T z(x).$$

As N sweeps from small to very large, we observe how test error evolves.

Tree based models We tried the main tree based models such as XGboost, a random forest, Adaboost, and simple decision trees.

Convolutionnal Neural Network (CNN) We implemented a shallow CNN (2 convulotion layer, 1 dense layer) with a growing complexity (we increased the number of nodes and the number of filters).

Fully Connected (Dense) Neural Network We experiment with a single hidden layer up to multiple layers, varying the number of hidden units from below n to well above n.

4.3 Implementation Details

Code Infrastructure. We use *TensorFlow* and *Keras* for neural networks, while RFF expansions are handled via custom Python scripts. Classical ML models were used through the scikit learn API.

[1] `https://github.com/Pierre-Clayton/AdvancedML-Project`

5 Empirical Analysis and Results

5.1 Experiences at small scale

One of us focused on illustrating the double descent phenomenon at small scale, with classical supervised ML algorithms. We evaluated the performance of the test with *mean squared error* for classical linear regressors or linear regressors based on RFF and tree based models. Plots typically show train/test curves against parameter count or feature dimension.

5.1.1 Results for Linear Regression Model

We decided to apply this model on synthetic data to visualize well and to be sure to have clear results. We generate 100 data points x from a uniform law on $[-2\pi, 2\pi]$ and then the target y was $y = 2cos(2x) + 3exp(\frac{x}{3}) + \epsilon$ where $\epsilon \sim N(0, 0.3)$. We increased complexity thanks to Polynomial Features from scikit learn which creates polynomial features up to a degree d, with interactions. Here are the results:

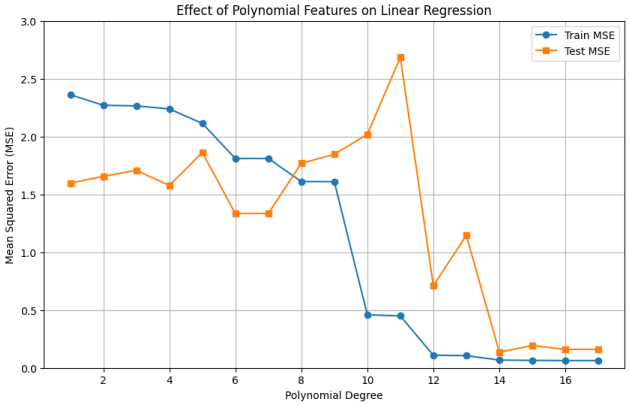

Figure 1: RMSE vs degree of polynomial features for the Linear Regression Model on synthetic data.

We see a bit the double descent phenomenom.

5.1.2 Results for the RFF Model

Now, with RFF expansions, once N approximates or surpasses n, the training error drops to near zero. We observe a characteristic peak in test error near this interpolation point, followed by a *second descent* once $N > n$. This phenomenon is clearly illustrated by our experiment on Taxi trip Pricing dataset:

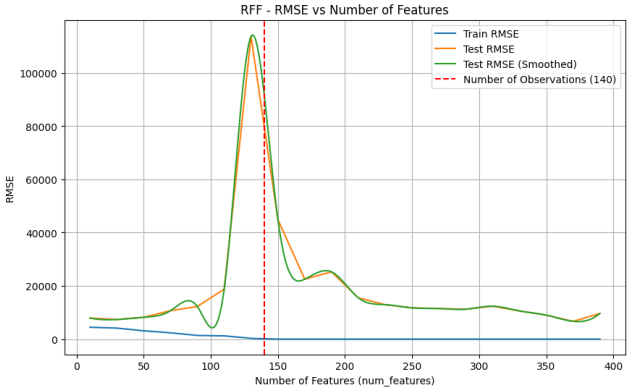

Figure 2: RMSE vs Number of tree for the RFF Model on the Taxi trip pricing dataset. The training error drops near zero, and the test error shows a peak near the interpolation point (red dotted line).

5.1.3 Results for tree based models

For this kind of models, it was harder to show the double descent curve due to the robustness of those models, especially for the classical Random forest models, or XGBoost models. Nonetheless, we succeed to show a little double descent with the Adaboost model on the Taxi trip Dataset: (Figure 3)

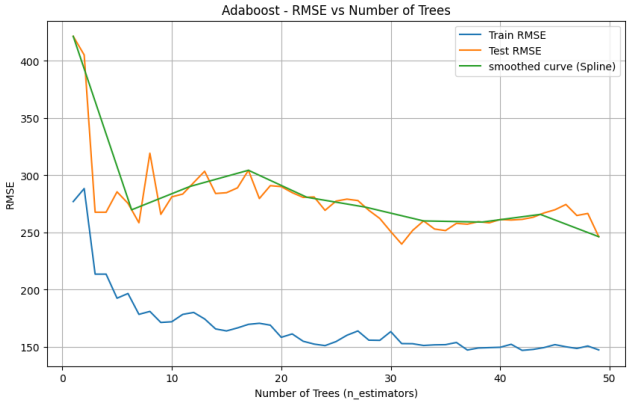

Figure 3: RMSE vs Number of Features for the Adaboost Model on the Taxi trip pricing dataset.

5.2 Experiences at (relatively) large scale

The phenomenom was hard to illustrate at small scale, with classical ML algorithms. One of us focused to illustrating the double descent at large scale since according to the litterature and based on what we witness with the essors of LLM, the double descent seems to be a phenomenom which occurs mainly at large scale, for very large neural networks. We tried nonetheless to show this phenomenom with our small ressources.

5.2.1 Results for the Dense Model

Here is the results for Dense models trained on the famous MNIST dataset:

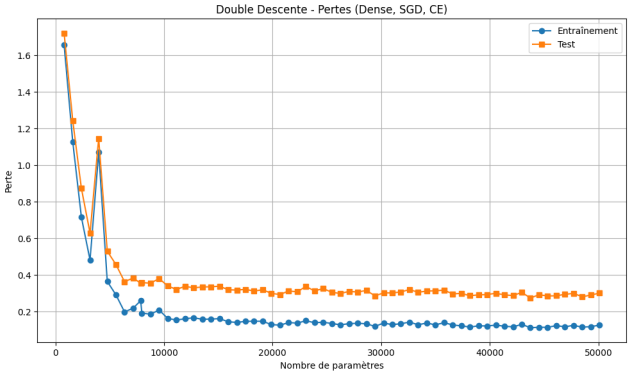

Figure 4: MSE vs Number of parameters for dense neural network on the MNIST dataset.

We can clearly observe the double descent around 4000, which is exactly the number of data points provided to the model during the training sessions.

5.2.2 Results for other models

We have tried many different models (RFF + dense, CNN), but the results were globally disapointing. You will see some of our graphs in the notebook directly.

5.2.3 Comparative Discussion of Optimization and Loss Functions

The phenomenon persists whether we use cross-entropy or mean squared error, and with optimizers like SGD or Adam. Subtle differences appear in how pronounced the peak is, suggesting that optimization dynamics can modulate overfitting severity near the threshold.

6 Discussion

6.1 Interpretation of Theoretical and Empirical Findings

Our findings confirm that double descent is a universal phenomenon, observed across linear regression, RFF expansions, and dense neural networks. This aligns with prior studies [2, 5], emphasizing that in overparameterized settings, standard gradient descent selects solutions with *minimum norm* or *smoother* characteristics, which inherently generalize better.

6.2 Role of Inductive Biases

Gradient-based optimization exhibits an implicit preference for small-norm solutions, even in the absence of explicit regularization [2]. In highly overparameterized regimes, these implicit biases allow the models to generalize effectively, explaining the surprising second descent in test error.

6.3 Limitations and Future Work

Our work is constrained to relatively simple architectures and controlled datasets. Several directions for future exploration include:

- Extending experiments to Transformer models,
- Evaluating double descent on larger and more diverse datasets,
- Investigating the role of implicit biases in complex, non-convex optimization landscapes.

7 Conclusion

This work demonstrates that the double descent phenomenon transcends traditional model types, including linear, kernel-based (RFF), and neural networks. By exploring both theoretical insights and practical experiments across datasets such as MNIST, we show that increasing model capacity beyond the interpolation threshold can enhance test performance. These findings challenge our conventional belief as students in a statistic school that overfitting necessarily harms generalization. This highlight the critical role of implicit and explicit inductive biases in shaping learning outcomes.

References

[1] C. Zhang, S. Bengio, Y. Dai, M. Hardt, B. Recht, and O. Vinyals. Understanding deep learning requires rethinking generalization. In *International Conference on Learning Representations*, 2017.

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

[3] T. Hastie, R. Tibshirani, and J. Friedman. *The Elements of Statistical Learning*. Springer, 2009.

[4] S. Geman, E. Bienenstock, and R. Doursat. Neural networks and the bias/variance dilemma. *Neural Computation*, 4(1):1–58, 1992.

[5] P. Nakkiran, G. Kaplun, Y. Bansal, T. Yang, B. Barak, and I. Sutskever. Deep double descent: Where bigger models and more data hurt. *arXiv preprint arXiv:1912.02292*, 2019.

[6] M. Lafon, A. Thomas. Deep double descent: A tutorial. Available at: `https://marclafon.github.io/assets/pdf/tutorial_deep_double_descent.pdf`, 2020.