Tutorial on the Double Descent Phenomenon in Deep Learning

Marc Lafon* lafon.ma.ml@gmail.com
Alexandre Thomas* hi@alxthm.com

January 28, 2021

Abstract

Combining empirical risk minimization with capacity control is a classical strategy in machine learning when trying to control the generalization gap and avoid overfitting, as the model class capacity gets larger. Yet, in modern deep learning practice, very large over-parameterized models (e.g. neural networks) are optimized to fit perfectly the training data and still obtain great generalization performance. Past the *interpolation point*, increasing model complexity seems to actually lower the test error.

In this tutorial, we explain the concept of *double descent* introduced by [4], and its mechanisms. Section 1 sets the classical statistical learning framework and introduces the double descent phenomenon. By looking at a number of examples, section 2 introduces *inductive biases* that appear to have a key role in double descent by selecting, among the multiple interpolating solutions, a smooth empirical risk minimizer. Finally, section 3 explores the double descent with two linear models, and gives other points of view from recent related works.

Contents

*Equal contribution, work done during the DAC master at Sorbonne Université, Paris, France, under the supervision of Prof. Gérard Biau.

1 Generalization error : classical view and modern practice

1.1 Definitions and results from statistical learning

In statistical learning theory, the supervised learning problem consists of finding a good predictor $h_n : \mathbb{R}^d \to \{0, 1\}$, based on some training data D_n. The data is typically assumed to come from a certain distribution, i.e. $D_n = \{(X_1, Y_1), \dots, (X_n, Y_n)\}$ is a collection of n i.i.d. copies of the random variables (X, Y), taking values in $\mathbb{R}^d \times \{0, 1\}$ and following a data distribution $P(X, Y)$. We also restrict ourselves to a given class of predictors by choosing $h_n \in \mathcal{H}$.

Definition 1 (True risk)**.** *With $\ell(h(X), Y) = \mathbb{1}_{(h(X) \neq Y)}$ the 0-1 loss, the* true risk *(or* true error*) of a predictor $h : \mathbb{R}^d \to \{0, 1\}$ is defined as*

$$L(h) = \mathbb{E}[\ell(h(X), Y)] = \mathbb{P}(h(X) \neq Y)$$

The true risk is also called the expected risk *or the* generalization error*.*

Remark 1. *We choose in this section a classification setting, but a regression setting could be adopted as well, for instance with Y and h_n taking values in $\mathbb{R}$ (which we will sometimes do in the subsequent sections). In this case, the 0-1 loss is replaced by other loss functions, such as the squared error loss $\ell(\hat{y}, y) = (\hat{y} - y)^2$.*

In practice, the true distribution of (X, Y) is unknown, so we have to resort to a proxy measure based on the available data.

Definition 2 (Empirical risk)**.** *The* empirical risk *of a predictor $h : \mathbb{R}^d \to \mathbb{R}$ on a training set D_n is defined as*

$$L_n(h) = \frac{1}{n} \sum_{i=1}^{n} \ell(h(X_i), Y_i)$$

Definition 3 (Bayes risk)**.** *A predictor $h^* : \mathbb{R}^d \to \{0, 1\}$ minimizing the true risk, i.e. verifying*

$$L(h^*) = \inf_{h : \mathbb{R}^d \to \{0,1\}} L(h)$$

is called a Bayes estimator*. Its risk $L^* = L(h^*)$ is called the* Bayes risk

Using D_n, our objective is to find a predictor h_n as close as possible to h^*.

Definition 4 (Consistency)**.** *A predictor h_n is* consistent *if*

$$\mathbb{E}L(h_n) \underset{n \to \infty}{\to} L^*$$

The *empirical risk minimization (ERM)* approach [25] consists in choosing a predictor that minimizes the empirical risk on D_n : $h_n^* \in \operatorname{argmin}_{h \in \mathcal{H}} L_n(h)$. This is something that can be done or approximated in practice, thanks to a wide range of algorithms and optimization procedures, but it is also necessary to ensure that our predictor h_n^* performs well in general

and not only on training data. Depending on the chosen class of predictors $\mathcal{H}$, statistical learning theory can give us guarantees or insights to make sure h_n^* generalizes well to unseen data.

1.2 Classical view

The gap between any predictor $h_n \in \mathcal{H}$ and h^* can be decomposed as follows.

$$L(h_n) - L^* = \underbrace{L(h_n) - \inf_{h \in \mathcal{H}} L(h)}_{\text{estimation error}} + \underbrace{\inf_{h \in \mathcal{H}} L(h) - L^*}_{\text{approximation error}}$$

Remark 2. *In addition to the approximation error (approximating reality with a model) and estimation error (learning a model with finite data) which fits in the statistical learning framework and are the focus of this tutorial, there is actually another source of error, the* optimization error*. This is the gap between the risk of the predictor returned by the optimization procedure and an empirical risk minimizer h_n^*.*

Proposition 5. *For any empirical risk minimizer $h_n^* \in \text{argmin}_{h \in \mathcal{H}} L_n(h)$, the estimation error verifies*

$$L(h_n^*) - \inf_{h \in \mathcal{H}} L(h) \leq 2 \sup_{h \in \mathcal{H}} |L_n(h) - L(h)|$$

Proof. We have

$$L(h_n^*) - \inf_{h \in \mathcal{H}} L(h) \leq |L(h_n^*) - L_n(h_n^*)| + |L_n(h_n^*) - \inf_{h \in \mathcal{H}} L(h)|$$

With

$$|L(h_n^*) - L_n(h_n^*)| \leq \sup_{h \in \mathcal{H}} |L_n(h) - L(h)|$$

since $h_n^* \in \mathcal{H}$, and :

$$|L_n(h_n^*) - \inf_{h \in \mathcal{H}} L(h)| = |\inf_{h \in \mathcal{H}} L_n(h) - \inf_{h \in \mathcal{H}} L(h)| \leq \sup_{h \in \mathcal{H}} |L_n(h) - L(h)|$$

after separating the cases where $|\inf_{h \in \mathcal{H}} L_n(h) - \inf_{h \in \mathcal{H}} L(h)| > 0$ and $|\inf_{h \in \mathcal{H}} L_n(h) - \inf_{h \in \mathcal{H}} L(h)| < 0$. □

The classical machine learning strategy is to find the right $\mathcal{H}$ to keep both the approximation error and the estimation error low.

1. When $\mathcal{H}$ is too small, no predictor $h \in \mathcal{H}$ is able to model the complexity of the data and to approach the Bayes risk. This is called *underfitting*.
2. When $\mathcal{H}$ is too large, the bound from proposition 5 (maximal generalization gap over $\mathcal{H}$) will increase, and the chosen empirical risk minimizer h_n^* may generalize poorly despite having a low training error. This is called *overfitting*.

Remark 3. *Similarly, the expected error can also be decomposed into a bias term due to model mis-specification and a variance term due to random noise being modeled by*

h_n^*. *This is the* bias-variance *trade-off, and is also highly dependent on the capacity of* $\mathcal{H}$*, the chosen class of predictors.*

Exercise 1 (Bias-Variance decomposition)**.** *Assume that* $Y = h(X) + \epsilon$*, with* $\mathbb{E}[\epsilon] = 0, Var(\epsilon) = \sigma^2$*. Show that, for any* $x \in \mathbb{R}^d$*, the expected error of a predictor* h_n *obtained with the random dataset* D_n *is :*

$$\mathbb{E}[(Y - h_n(X))^2 | X = x] = (h(x) - \mathbb{E}h_n(x))^2 + \mathbb{E}[(\mathbb{E}h_n(x) - h_n(x))^2] + \sigma^2$$

In order to ensure a consistent estimator h_n, we can control $\mathcal{H}$ explicitly e.g. by choosing the number of features used in a linear classifier, or the number of layers of a neural network.

Theorem 6 (Vapnik-Chervonenkis inequality)**.** *For any data distribution* $P(X, Y)$*, by using* $V_{\mathcal{H}}$ *the VC-dimension of the class* $\mathcal{H}$ *as a measure of the class complexity, one has*

$$\mathbb{E} \sup_{h \in \mathcal{H}} |L_n(h) - L(h)| \leq 4\sqrt{\frac{V_{\mathcal{H}} \log(n+1)}{n}}$$

A complete introduction to Vapnik-Chervonenkis theory is outside the scope of this tutorial, but $V_{\mathcal{H}}$ can be defined as the cardinality of the largest set of points that can be shattered, i.e. there is at least one $h \in \mathcal{H}$ that can assign all possible labels to the set. Combining this result with proposition 5 gives a useful bound on the generalization error for a number of model classes. For instance, if $\mathcal{H}$ is a class of linear classifiers using d features (potentially non-linear transformations of input x), then we have : $V_{\mathcal{H}} \leq d + 1$.

Other measures of the richness of the model class $\mathcal{H}$ also exist, such as the *Rademacher complexity*, and can be useful in situations where $V_{\mathcal{H}} = +\infty$, or in regression settings.

1.3 Modern practice

Following results from section 1.1, a widely adopted view is that, after a certain threshold, "larger models are worse" as they will overfit and generalize poorly. Yet, in modern machine learning practice, very large models with enough parameters to reach almost zero training error are frequently used. Such models are able to fit almost perfectly (i.e. *interpolate*) the training data and still generalize well, actually performing better than smaller models (e.g. to classify 1.2M examples, AlexNet had 60M parameters and VGG-16 and VGG-19 both exceeded 100M parameters [8]). Understanding generalization of overparameterized models in modern deep learning is an active field of research, and we focus on the *double descent* phenomenon, first demonstrated by [3] and illustrated in Figure 1.

For simpler class of models, classical statistical learning guarantee that the test risk decreases when the class of models gets more complex, until a point where the bounds do not control the risk anymore. However it seems that, beyond a certain threshold, increasing the capacity of the models actually decreases the test risk again. This is the "modern" interpolating regime, with overparameterized models. As this phenomenon depends not only on the class of predictors $\mathcal{H}$, but also on the training algorithm and regularization techniques, we define a *training procedure* $\mathcal{T}$ to be any procedure that takes as input a dataset D_n and outputs a classifier h_n, i.e. $h_n = \mathcal{T}(D_n) \in \mathcal{H}$. We can now make an informal hypothesis, after defining the notion of *effective model complexity* (from [16]).

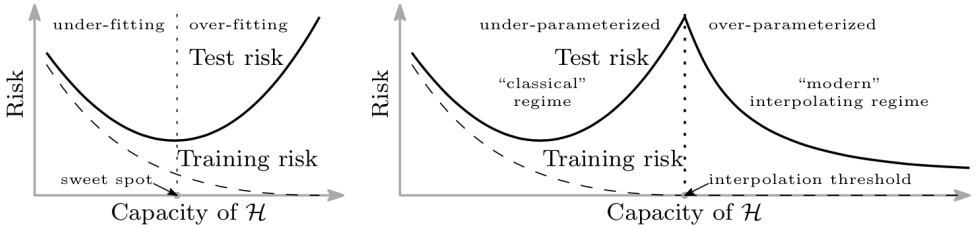


Figure 1: The *classical risk curve* arising from the bias-variance trade-off and the *double descent risk curve* with the observed modern interpolation regime. Taken from [3]

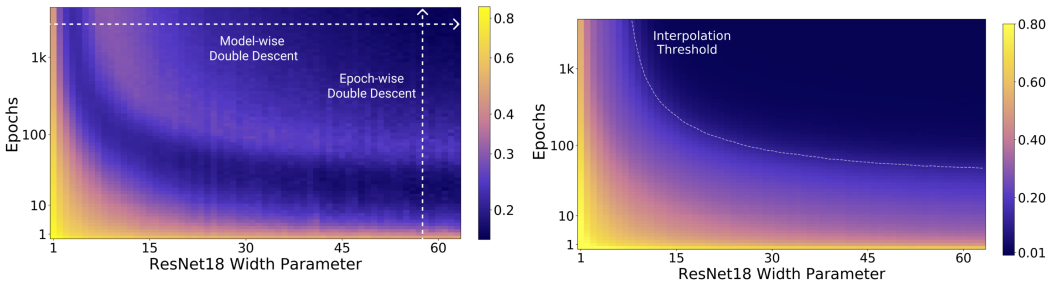
Definition 7 (Effective Model Complexity)**.** *The* Effective Model Complexity (EMC) *of a training procedure $\mathcal{T}$, w.r.t. distribution $P(X, Y)$, is the maximum number of samples n on which $\mathcal{T}$ achieves on average ≈ 0 training error. That is, for $\epsilon > 0$:*

$$EMC_{P,\epsilon}(\mathcal{T}) = \max\{n \in \mathbb{N} | \mathbb{E}L(h_n) \leq \epsilon\}$$

Hypothesis 8 (Generalized Double Descent hypothesis, informal)**.** *For any data distribution $P(X, Y)$, neural-network-based training procedure $\mathcal{T}$, and small $\epsilon > 0$, if we consider the task of predicting labels based on n samples from P then, as illustrated on figure 1:*

- Under-parameterized regime. *If $EMC_{P,\epsilon}(\mathcal{T})$ is sufficiently smaller than n, any perturbation of $\mathcal{T}$ that increases its effective complexity will decrease the test error.*
- Critically parameterized regime. *If $EMC_{P,\epsilon}(\mathcal{T}) \approx n$, then a perturbation of $\mathcal{T}$ that increases its effective complexity might decrease or increase the test error.*
- Over-parameterized regime. *If $EMC_{P,\epsilon}(\mathcal{T})$ is sufficiently larger than n, any perturbation of $\mathcal{T}$ that increases its effective complexity will decrease the test error.*

Empirically, this definition of effective model capacity translates into multiple axis along which the double descent can be observed : *epoch-wise*, *model-wise* (e.g. increasing the width of convolutional layers or the embedding dimension of transformers) and even with regularization, by decreasing weight decay. Figure 2 illustrates this.



(a) Test error as a function of model size and train epochs

(b) Train error of the corresponding models

Figure 2: All models are Resnet18s trained on CIFAR-10 with 15% label noise (training labels artificially made incorrect), data-augmentation, and Adam for up to 4K epochs. Taken from [16]

2 Inductive biases

In the supervised learning problem, the model needs to generalize patterns observed in the training data to unseen situations. In that sense, the learning procedure has to use mechanisms similar to inductive reasoning. As there are generally many possible generalizable solutions, [15] advocated the need for inductive biases in learning generalization. Inductive biases are assumptions made in order to prioritized one solution over another both exhibiting the same performance on the training data. For example, a common inductive bias is the Occam's razor principle stating that in case of equally good solutions the "simplest" one should be preferred. Another form of inductive bias is to incorporate some form of prior knowledge about the structure of the data, its generation process or to constrain the model to respect specific properties.

In the under-parameterized regime, regularization can be used for capacity control and is a form of inductive bias. One common choice is to search for small norm solutions, e.g. adding a penalty term, the L_2 norm of the weights vector. This is known as Tikhonov regularization in the linear regression setting (also known as Ridge regression in this case).

In the over-parameterized regime, as the complexity of $\mathcal{H}$ and the EMC increases, the number of interpolating solutions (i.e. achieving almost zero training error) increases and the question of the selection of a particular element in $\operatorname{argmin}_{h \in \mathcal{H}} L_n(h)$ is crucial. Inductive biases, explicit or implicit, are a way to find predictors that generalize well.

2.1 Explicit inductive biases

Several common inductive biases can be used to observe a model-wise double descent [4] (e.g. as the number of parameters N increases).

Least Norm For the model class of Random Fourier Features (defined in section 3.2), by choosing explicitly the minimum norm linear regression in the feature space. This bias towards the choice of parameters of minimum norm is common to a lot of machine learning model. For example, the ridge regression induce a constraint on the L_2 norm of the solution and the lasso regression on the L_1 norm. We can also see the support vector machine (SVM) as a way of inducing a least norm bias because maximizing the margin is equivalent to minimizing the norm of the parameter under the constraint that all points are well classified.

Model architecture Another way of inducing a bias is by choosing a particular class of functions that we think is well suited for our problem. The authors in [2] discuss different type of inductive bias considered by different type of neural network architectures. Working with images it is better to use a convolutional neural network (CNN) as it can induce translational equivariance, whereas the recurrent neural network (RNN) is better suited to capture long-term dependencies in a sequence data. Using a naive Bayes classifier is of great utility if we know that the features are independent, etc.

Ensembling Random forest models use yet another type of inductive bias. By averaging potentially non-smooth interpolating trees, the interpolating solution has a higher degree of smoothness and generalizes better than any individual interpolating tree.

2.2 Implicit Bias of gradient descent

Gradient descent is a widely used optimization procedure in machine learning, and has been observed to converge on solutions that generalize surprisingly well, thanks to an implicit inductive bias.

We recall that the gradient descent update rule for parameter w using a loss function $\mathcal{L}$ is the following (where $\eta > 0$ is the step size):

$$w_{k+1} = w_k - \eta \nabla \mathcal{L}(w)$$

2.2.1 Gradient descent in under-determined least squares problem

Consider a non-random dataset $\{(x_i, y_i)\}_{i=1}^n$, with $(x_i, y_i) \in \mathbb{R}^d \times \mathbb{R}$, for $i \in \{1, \dots, n\}$ and let $\boldsymbol{X} \in \mathbb{R}^{n \times d}$ be the matrix which rows are the x_i^T and $y \in \mathbb{R}^n$ the column vector which elements are the y_i. We consider the linear least squares:

$$\min_{w \in \mathbb{R}^d} \mathcal{L}(w) = \min_{w \in \mathbb{R}^d} \frac{1}{2} \|\boldsymbol{X}w - y\|^2 \tag{1}$$

We will study the property of the solution found using gradient descent.

Definition 9 (*Moore-Penrose pseudo-inverse*)**.** *Let $\boldsymbol{A} \in \mathbb{R}^{n \times d}$ be a matrix, the Moore-Penrose pseudo-inverse is the only matrix $\boldsymbol{A}^+$ satisfying the following properties:*

(i)	$\boldsymbol{A}\boldsymbol{A}^+\boldsymbol{A} = \boldsymbol{A}$	*(iii)*	$(\boldsymbol{A}^+\boldsymbol{A})^T = \boldsymbol{A}^+\boldsymbol{A}$
(ii)	$\boldsymbol{A}^+\boldsymbol{A}\boldsymbol{A}^+ = \boldsymbol{A}^+$	*(iv)*	$(\boldsymbol{A}\boldsymbol{A}^+)^T = \boldsymbol{A}\boldsymbol{A}^+$

Furthermore, if $\text{rank}(\boldsymbol{A}) = min(n, d)$ *then $\boldsymbol{A}^+$ has a simple algebraic expression:*

- *If $n < d$, then* $\text{rank}(\boldsymbol{A}) = n$ *and* $\boldsymbol{A}^+ = \boldsymbol{A}^T(\boldsymbol{A}\boldsymbol{A}^T)^{-1}$
- *If $d < n$, then* $\text{rank}(\boldsymbol{A}) = d$ *and* $\boldsymbol{A}^+ = (\boldsymbol{A}^T\boldsymbol{A})^{-1}\boldsymbol{A}^T$
- *If $d = n$, then $\boldsymbol{A}$ is invertible and* $\boldsymbol{A}^+ = \boldsymbol{A}^{-1}$

Lemma 10. *For a matrix $\boldsymbol{A} \in \mathbb{R}^{n \times d}$, $Im(I\text{-}\boldsymbol{A}^+\boldsymbol{A}) = Ker(\boldsymbol{A})$, $Ker(\boldsymbol{A}^+) = Ker(\boldsymbol{A}^T)$ and $Im(\boldsymbol{A}^+) = Im(\boldsymbol{A})$.*

Proof. Left as an exercise. □

Theorem 11. *The set of solutions $\mathcal{S}_{LS}$ of the least square problem (i.e. minimizing (1)) is exactly:*

$$\mathcal{S}_{LS} = \{\boldsymbol{X}^+y + (\boldsymbol{I}\text{-}\boldsymbol{X}^+\boldsymbol{X})u, u \in \mathbb{R}^d\}$$

Proof sketch. Writing $\boldsymbol{X}w - y = \boldsymbol{X}w - \boldsymbol{X}\boldsymbol{X}^+y - (\boldsymbol{I} - \boldsymbol{X}\boldsymbol{X}^+)y$ prove using pseudo-inverse properties that $\boldsymbol{X}w - \boldsymbol{X}\boldsymbol{X}^+y$ and $(\boldsymbol{I} - \boldsymbol{X}\boldsymbol{X}^+)y$ are orthogonal. Then using the Pythagorean theorem show that $\|\boldsymbol{X}w - y\|^2 \geq \|(\boldsymbol{I} - \boldsymbol{X}\boldsymbol{X}^+)y\|^2$, this inequality being an equality if and only if $\boldsymbol{X}w = \boldsymbol{X}\boldsymbol{X}^+y$. Then $\boldsymbol{X}^+y$ is one solution of (1) and by Lemma 10 we can conclude that $\{\boldsymbol{X}^+y + (\boldsymbol{I} - \boldsymbol{X}^+\boldsymbol{X})u, u \in \mathbb{R}^d\}$, is the set of solutions. ∘

Remark 4. *Depending on the* rank *of* $\boldsymbol{X}$*, the set of solutions* $\mathcal{S}_{LS}$ *will differ depending on the expression of* $\boldsymbol{X}^+$*:*

- *If* $n < d$ *and* $\text{rank}(\boldsymbol{X}) = n$*, then* $\boldsymbol{X}^+ = \boldsymbol{X}^T(\boldsymbol{X}\boldsymbol{X}^T)^{-1}$*:* $\mathcal{S}_{LS} = \{\boldsymbol{X}^T(\boldsymbol{X}\boldsymbol{X}^T)^{-1}y + (\boldsymbol{I} - \boldsymbol{X}^T(\boldsymbol{X}\boldsymbol{X}^T)^{-1}\boldsymbol{X})u, u \in \mathbb{R}^d\}$
- *If* $d < n$ *and* $\text{rank}(\boldsymbol{X}) = d$*, then* $\boldsymbol{X}^+ = (\boldsymbol{X}^T\boldsymbol{X})^{-1}\boldsymbol{X}^T$*:* $\mathcal{S}_{LS} = \{\boldsymbol{X}^T(\boldsymbol{X}\boldsymbol{X}^T)^{-1}y\}$
- *If* $d = n$ *and* $\boldsymbol{X}$ *is invertible, then* $\boldsymbol{X}^+ = \boldsymbol{X}^{-1}$*:* $\mathcal{S}_{LS} = \{\boldsymbol{X}^{-1}y\}$

Proposition 12. *Assuming that* $\boldsymbol{X}$ *has* rank n *and* $n < d$*, the least square problem (1) has infinitely many solutions and* $\boldsymbol{X}^+y = \boldsymbol{X}^T(\boldsymbol{X}\boldsymbol{X}^T)^{-1}y$ *is the minimum euclidean norm solution.*

Proof. From the previous remark, we know that $\mathcal{S}_{LS} = \{\boldsymbol{X}^T(\boldsymbol{X}\boldsymbol{X}^T)^{-1}y + (\boldsymbol{I} - \boldsymbol{X}^T(\boldsymbol{X}\boldsymbol{X}^T)^{-1}\boldsymbol{X})u, u \in \mathbb{R}^d\}$

For arbitrary $u \in \mathbb{R}^d$,

$$\begin{aligned}(\boldsymbol{X}^+y)^T(\boldsymbol{I} - \boldsymbol{X}^+\boldsymbol{X})u \overset{\text{(ii)}}{=} (\boldsymbol{X}^+\boldsymbol{X}\boldsymbol{X}^+y)^T(\boldsymbol{I} - \boldsymbol{X}^+\boldsymbol{X})u &= (\boldsymbol{X}^+y)^T(\boldsymbol{X}^+\boldsymbol{X})^T(\boldsymbol{I} - \boldsymbol{X}^+\boldsymbol{X})u \\ &\overset{\text{(iii)}}{=} (\boldsymbol{X}^+y)^T\boldsymbol{X}^+\boldsymbol{X}(\boldsymbol{I} - \boldsymbol{X}^+\boldsymbol{X})u \\ &= (\boldsymbol{X}^+y)^T\boldsymbol{X}^+(\boldsymbol{X} - \boldsymbol{X}\boldsymbol{X}^+\boldsymbol{X})u \overset{\text{(i)}}{=} 0\end{aligned}$$

using (i), (ii) and (iii) from Definition 9. Thus, $(\boldsymbol{X}^+y)$ and $(\boldsymbol{I} - \boldsymbol{X}^+\boldsymbol{X})u$ are orthogonal $\forall u \in \mathbb{R}^d$, and applying the Pythagorean theorem gives:

$$\left\|(\boldsymbol{X}^+y) + (\boldsymbol{I} - \boldsymbol{X}^+\boldsymbol{X})u\right\|^2 = \left\|(\boldsymbol{X}^+y)\right\|^2 + \left\|(\boldsymbol{I} - \boldsymbol{X}^+\boldsymbol{X})u\right\|^2 \geq \left\|(\boldsymbol{X}^+y)\right\|^2$$

□

Theorem 13. *If the linear least square problem (1) is under-determined, i.e.* $(n < d)$ *and* $\text{rank}(\boldsymbol{X}) = n$*, using gradient descent with a fixed learning rate* $0 < \eta < \frac{1}{\sigma_{max}(\boldsymbol{X})}$*, where* $\sigma_{max}(\boldsymbol{X})$ *is the largest eigenvalue of* $\boldsymbol{X}$*, from an initial point* $w_0 \in Im(\boldsymbol{X}^T)$ *will converge to the minimum norm solution of (1).*

Proof. As $\boldsymbol{X}$ is assumed to be of row rank n, we can write its singular value decomposition as :

$$\boldsymbol{X} = \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^T = \boldsymbol{U}\begin{bmatrix}\boldsymbol{\Sigma}_1 & 0\end{bmatrix}\begin{bmatrix}\boldsymbol{V}_1^T \\ \boldsymbol{V}_2^T\end{bmatrix}$$

where $\boldsymbol{U} \in \mathbb{R}^{n\times n}$ and $\boldsymbol{V} \in \mathbb{R}^{d\times d}$ are orthogonal matrices, $\boldsymbol{\Sigma} \in \mathbb{R}^{n\times d}$ is a rectangular diagonal matrix and $\boldsymbol{\Sigma}_1 \in \mathbb{R}^{n\times n}$ is a diagonal matrix. The minimum norm solution w^* can be rewritten as :

$$w^* = \boldsymbol{X}^T(\boldsymbol{X}\boldsymbol{X}^T)^{-1}y = \boldsymbol{V}_1\boldsymbol{\Sigma}_1^{-1}\boldsymbol{U}^Ty$$

The gradient descent update rule is the following (where $\eta > 0$ is the step size):

$$w_{k+1} = w_k - \eta\nabla\mathcal{L}(w) = w_k - \eta\boldsymbol{X}^T(\boldsymbol{X}w_k - y) = (\boldsymbol{I} - \eta\boldsymbol{X}^T\boldsymbol{X})w_k + \eta\boldsymbol{X}^Ty$$

Then, by induction, we have :

$$w_k = (\boldsymbol{I} - \eta\boldsymbol{X}^T\boldsymbol{X})^k w_0 + \eta\sum_{l=0}^{k-1}(\boldsymbol{I} - \eta\boldsymbol{X}^T\boldsymbol{X})^l\boldsymbol{X}^Ty$$

Using the singular value decomposition of $\boldsymbol{X}$, we can see that $\boldsymbol{X}^T\boldsymbol{X} = \boldsymbol{V}\boldsymbol{\Sigma}^T\boldsymbol{\Sigma}\boldsymbol{V}^T$. Furthermore, as $\boldsymbol{V}$ is orthogonal, $\boldsymbol{V}^T\boldsymbol{V} = \boldsymbol{I}$.
Then, the gradient descent iterate at step k can be written:

$$\begin{aligned} w_k &= \boldsymbol{V}(\boldsymbol{I} - \eta\boldsymbol{\Sigma}^T\boldsymbol{\Sigma})^k\boldsymbol{V}^T w_0 + \eta\boldsymbol{V}\Big(\sum_{l=0}^{k-1}(\boldsymbol{I} - \eta\boldsymbol{\Sigma}^T\boldsymbol{\Sigma})^l\boldsymbol{\Sigma}^T\Big)\boldsymbol{U}^T y \\ &= \boldsymbol{V}\begin{bmatrix}(\boldsymbol{I} - \eta\boldsymbol{\Sigma}_1^2)^k & 0 \\ 0 & \boldsymbol{I}\end{bmatrix}\boldsymbol{V}^T w_0 + \eta\boldsymbol{V}\Big(\sum_{l=0}^{k-1}\begin{bmatrix}(\boldsymbol{I} - \eta\boldsymbol{\Sigma}_1^2)^l\boldsymbol{\Sigma}_1 \\ 0\end{bmatrix}\Big)\boldsymbol{U}^T y \end{aligned}$$

By choosing $0 < \eta < 1/\sigma_{max}(\boldsymbol{\Sigma}_1)$ with $\sigma_{max}(\boldsymbol{\Sigma}_1)$ the largest eigenvalue of $\boldsymbol{\Sigma}_1$, we guarantee that the eigenvalues of $\boldsymbol{I} - \eta\boldsymbol{\Sigma}^T\boldsymbol{\Sigma}$ are all strictly less than 1. Then :

$$\boldsymbol{V}\begin{bmatrix}(\boldsymbol{I} - \eta\boldsymbol{\Sigma}_1^2)^k & 0 \\ 0 & \boldsymbol{I}\end{bmatrix}\boldsymbol{V}^T w_0 \xrightarrow[k\to\infty]{} \boldsymbol{V}\begin{bmatrix}0 & 0 \\ 0 & \boldsymbol{I}\end{bmatrix}\boldsymbol{V}^T w_0 = \boldsymbol{V}_2\boldsymbol{V}_2^T w_0$$

and

$$\eta\sum_{l=0}^{k-1}\begin{bmatrix}(\boldsymbol{I} - \eta\boldsymbol{\Sigma}_1^2)^l\boldsymbol{\Sigma}_1 \\ 0\end{bmatrix} \xrightarrow[k\to\infty]{} \eta\begin{bmatrix}\sum_{l=0}^{\infty}(\boldsymbol{I} - \eta\boldsymbol{\Sigma}_1^2)^l\boldsymbol{\Sigma}_1 \\ 0\end{bmatrix} = \begin{bmatrix}\eta(\boldsymbol{I} - \boldsymbol{I} + \eta\boldsymbol{\Sigma}_1^2)^{-1}\boldsymbol{\Sigma}_1 \\ 0\end{bmatrix} = \begin{bmatrix}\boldsymbol{\Sigma}_1^{-1} \\ 0\end{bmatrix}$$

Finally, noting w_∞ the limit of gradient descent iterates we have in the limit :

$$w_\infty = \boldsymbol{V}_2\boldsymbol{V}_2^T w_0 + \boldsymbol{V}_1\boldsymbol{\Sigma}_1^{-1}\boldsymbol{U}^T y \quad = \boldsymbol{V}_2\boldsymbol{V}_2^T w_0 + \boldsymbol{X}^T(\boldsymbol{X}\boldsymbol{X}^T)^{-1}y = \boldsymbol{V}_2\boldsymbol{V}_2^T w_0 + w^*$$

Because w_0 is in the range of $\boldsymbol{X}^T$, we can write $w_0 = \boldsymbol{X}^T z$ for some $z \in \mathbb{R}^n$.

$$\boldsymbol{V}_2\boldsymbol{V}_2^T w_0 = \boldsymbol{V}\begin{bmatrix}0 & 0 \\ 0 & \boldsymbol{I}\end{bmatrix}\boldsymbol{V}^T\boldsymbol{X}^T z = \boldsymbol{V}\begin{bmatrix}0 & 0 \\ 0 & \boldsymbol{I}\end{bmatrix}\boldsymbol{V}^T\boldsymbol{V}\boldsymbol{\Sigma}^T\boldsymbol{U}^T z \quad = \boldsymbol{V}\begin{bmatrix}0 & 0 \\ 0 & \boldsymbol{I}\end{bmatrix}\begin{bmatrix}\boldsymbol{\Sigma}_1 \\ 0\end{bmatrix}\boldsymbol{U}^T = 0$$

Therefore gradient descent will converge to the minimum norm solution. □

2.2.2 Gradient descent on separable data

In this section we are concerned with the effect of using gradient descent on a classification problem on a linearly separable dataset and using a smooth (we will explain in what sens), strictly decreasing and non-negative surrogate loss function. For the sake of clarity, we will prove the results using the exponential loss function $\ell : x \mapsto e^{-x}$ but the results will be expressed for the more general case.

Definition 14 (*Linearly separable dataset*). *A dataset $\mathcal{D}_n = \{(x_i, y_i)\}_{i=1}^n$ where $\forall i \in [\![1, n]\!], (x_i, y_i) \in \mathbb{R}^d \times \{-1, 1\}$ is linearly separable if $\exists\ w_*$ such that $\forall i : y_i w_*^T x_i > 0$.*

The results of this section hold assuming the considered loss functions respect the following properties :

Assumption 1. *The loss function ℓ is positive, differentiable, monotonically decreasing to zero, (i.e. $\ell(u) > 0$, $\ell'(u) < 0$, $\lim_{u\to\infty}\ell(u) = \lim_{u\to\infty}\ell'(u) = 0$) and $\lim_{u\to-\infty}\ell'(u) \neq 0$.*

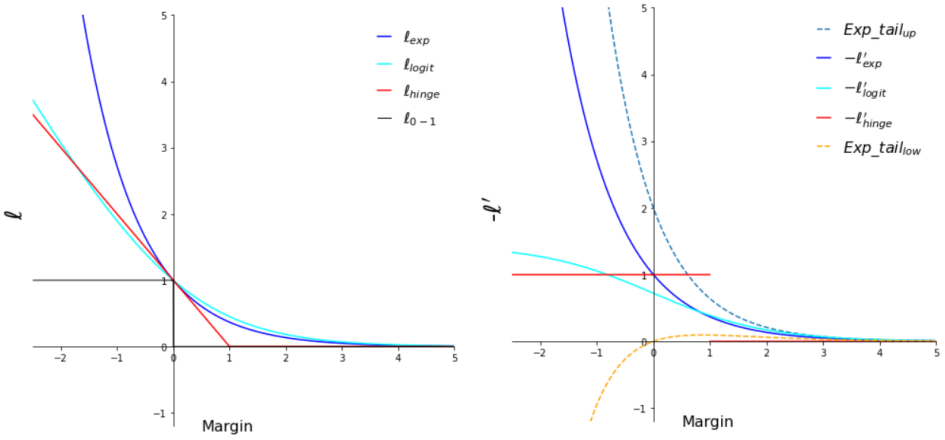
Assumption 2 (β-Smoothness)**.** *The gradient of ℓ is β-Lipschitz:*

$$\forall u, v \in \mathbb{R}, \quad \|\nabla \ell(u) - \nabla \ell(v)\| \leq \beta \|u - v\|.$$

Assumption 3 (Tight Exponential tail)**.** *Generally speaking a function $f : \mathbb{R} \mapsto \mathbb{R}$ is said to have a* tight exponential tail *if there exist positive constants c, a, μ_1, μ_2 and u_0 such that:*

$$\forall u > u_0, \ (1 - e^{-\mu_1 u}) \leq c \ f(u) e^{au} \leq (1 + e^{-\mu_2 u}).$$

In our case we will say that a differentiable loss function ℓ has a tight exponential tail *when its negative derivative $-\ell'$ has a tight exponential tail.*



(a) Losses **(b)** Negative Derivative Losses

Figure 3: Illustration of tight exponential tail property for different common loss functions. We can see that both exponential and logistic loss functions has a tight exponential tail. The hinge loss and 0-1 loss functions have been displayed for reference only.

We consider the following classification problem:

$$\min_{w \in \mathbb{R}^d} \mathcal{L}(w) = \min_{w \in \mathbb{R}^d} \sum_{i=1}^{n} \ell(y_i w^T x_i)$$

where $\forall i \in [\![1, n]\!], (x_i, y_i) \in \mathbb{R}^d \times \{-1, 1\}$ and $\ell : \mathbb{R} \mapsto \mathbb{R}_+^*$ is a surrogate loss function of the 0-1 loss.

We will study the behavior of the solution found by gradient descent using a fixed learning rate η:

$$w_{t+1} = w_t - \eta \nabla \mathcal{L}(w_t) = w_t - \eta \sum_{i=1}^{n} \ell'(y_i w_t^T x_i) y_i x_i \tag{2}$$

Lemma 15. *Let $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^n$ be a linearly separable dataset where $\forall i \in [\![1, n]\!], (x_i, y_i) \in \mathbb{R}^d \times \{-1, 1\}$ and $\ell : \mathbb{R} \mapsto \mathbb{R}_+^*$ a loss function under assumptions 1 and 2. Let w_t be the iterates of gradient descent using learning rate $0 < \eta < \frac{2}{\beta \sigma_{max}^2(X)}$ and any starting point w_0. Then we have:*

(1) $\lim_{t \to \infty} \mathcal{L}(w_t) = 0,$

(2) $\lim_{t\to\infty} \|w_t\| = \infty$,
(3) $\forall i: \quad \lim_{t\to\infty} y_i w_t^T x_i = \infty$,

Proof. As mentioned we use the exponential loss function: $\ell : u \mapsto e^{-u}$, which. Since $\mathcal{D}$ is linearly separable, $\exists w_*$ such that $w_*^T x_i > 0, \forall i$. Then for $w \in \mathbb{R}^d$:

$$w_*^T \nabla\mathcal{L}(w) = \sum_{i=1}^{n} \underbrace{-exp(-y_i w^T x_i)}_{<0} \underbrace{y_i w_*^T x_i}_{>0} < 0.$$

Therefore there is no finite critical points w, for which $\nabla\mathcal{L}(w) = 0$. But gradient descent on a smooth loss with an appropriate learning rate is always guaranteed to converge to a critical point : in other words $\nabla\mathcal{L}(w_t) \to 0$. This necessarily implies that $\|w_t\| \to \infty$, which is (2). It also implies that $\exists t_0$ s.t, $\forall t > t_0, \forall i : y_i w_t^T x_i > 0$ in order to make the exponential term converge to zero, this is (3). But in that case, we also have $\mathcal{L}(w_t) \to 0$, which is (1). □

The norm of the previous solution diverges, but we can normalize it to have norm 1.

Theorem 16. *Let $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^n$ be a linearly separable dataset where $\forall i \in [\![1, n]\!], (x_i, y_i) \in \mathbb{R}^d \times \{-1, 1\}$ and $\ell : \mathbb{R} \mapsto \mathbb{R}_+^*$ a loss function with under assumptions 1, 2 and 3. Let w_t be the iterates of gradient descent using a learning rate η such that $0 < \eta < \frac{2}{\beta\sigma_{max}^2(X)}$ and any starting point w_0. Then we have:*

$$\lim_{t\to\infty} \frac{w_t}{\|w_t\|} = \frac{w_{svm}}{\|w_{svm}\|}$$

where w_{svm} is the solution to the hard margin SVM:

$$w_{svm} = \underset{w\in\mathbb{R}^d}{\operatorname{argmin}} \|w\|^2 \quad s.t. \quad y_i w^T x_i \geq 1, \forall i.$$

Proof sketch. We will just give the main ideas behind the proof of this theorem using the exponential loss function. We will furthermore assume that $w_t/\|w_t\|$ converges to some limit w_∞. For a detailed proof and in the more general case of the loss function having properties 1 to 3 please refer to [23].

By Lemma 15 we have $\forall i: \ \lim_{t\to\infty} y_i w_t^T x_i = \infty$. As $\frac{w_t}{\|w_t\|}$ converges to w_∞ we can write $w_t = g(t)w_\infty + \rho(t)$ such that $g(t) \to \infty$, $\forall i: \ y_i w_\infty^T x_i > 0$ and $\lim_{t\to\infty} \frac{\rho(t)}{g(t)} = 0$. The gradient can then be written as:

$$-\nabla\mathcal{L}(w_t) = \sum_{i=1}^{n} e^{-y_i w_t^T x_i} x_i = \sum_{i=1}^{n} e^{-g(t) y_i w_\infty^T x_i} \ e^{-y_i \rho(t)^T x_i} x_i \tag{3}$$

We can see that as $g(t) \to \infty$ only the samples with largest exponents in the sum of the right hand side of (3) will contribute to the gradient. But the exponents are maximized for $i \in \mathcal{S} = argmin_i \ y_i w_\infty^T x_i$ which correspond to the samples minimizing the margin: i.e. the support vectors $X_S = \{x_i, i \in \mathcal{S}\}$. The negative gradient $-\nabla\mathcal{L}(w_t)$ would then asymptotically become a non-negative linear combination of support vectors and because $\|w_t\| \to \infty$ (by Lemma 15) the first gradient steps will be negligible and the limit w_∞ will get closer and closer to a non-negative linear combination of support vectors and so will its scaled version $\hat{w} = w_\infty / \min_i y_i w_\infty^T x_i$ (the scaling is done to make the margin of the support vectors equal to 1). We can therefore write:

$$\hat{w} = \sum_{i=1}^{n} \alpha_i x_i \quad with \ \begin{cases} \alpha_i x_i \geq 0 \ and \ y_i \hat{w}^T x_i = 1 \ if \ i \in \mathcal{S} \\ \alpha_i x_i = 0 \ and \ y_i \hat{w}^T x_i > 1 \ if \ i \notin \mathcal{S} \end{cases} \tag{4}$$

We can recognize the KKT conditions for the hard margin SVM problem (see [6] Chapter 7, Section 7.1) and conclude that $\hat{w} = w_{svm}$. Then $\frac{w_\infty}{\|w_\infty\|} = \frac{w_{svm}}{\|w_{svm}\|}$. ∘

Remark 5. *In the proof of Lemma 15 we have seen that $\mathcal{L}(w_t) \to 0$. That means that gradient descent converges to a global minimum.*

Remark 6. *Gradient descent has been suspected to induce a bias towards simple solutions, not only in the previous linear settings, but in deep learning as well, greatly improving generalization performance. It would explain the double descent behavior of deep learning architectures, and recent works such as [11] have been studying the learning dynamics in more complex settings.*

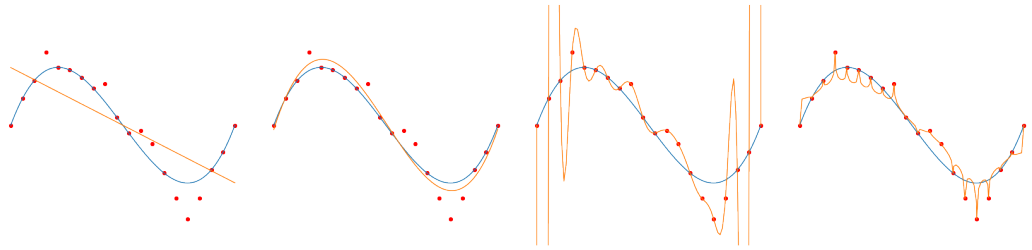
3 The reasons behind double descent

In this section, we consider two settings where double descent can be empirically observed and mathematically justified, in order to give the reader some intuition on the role of inductive biases. We conclude with some references to recent related works studying optimization in the over-parameterized regime, or linking the double descent to a physical phenomenon named *jamming*.

Fully understanding the mechanisms behind this phenomenon in deep learning remains an open question but, as introduced in section 2, inductive biases seem to play a key role.

In the over-parameterized regime, empirical risk minimizers are able to interpolate the data. Intuitively :

- Near the interpolation point, there are very few solutions that fit the training data perfectly. Hence, any noise in the data or model mis-specification will destroy the global structure of the model, leading to an irregular solution that generalizes badly (figure 4c).
- As effective model capacity grows, many more interpolating solutions exist, including some that generalize better and can be selected thanks to the right inductive bias, e.g. smaller norm (figure 4d), or ensemble methods.



(a) $d = 1$ **(b)** $d = 3$ **(c)** $d = 20$ **(d)** $d = 1000$

Figure 4: Fitting degree d Legendre polynomials (orange curve) to $n = 20$ noisy samples (red dots), from a polynomial of degree 3 (blue curve). Gradient descent is used to minimize the squared error, which leads to the smallest norm solution (considering the norm of the vector of coefficients). Taken from [17].

3.1 Linear Regression with Gaussian Noise

In this section we consider the family class $(\mathcal{H}_p)_{p\in[\![1,d]\!]}$ of linear functions $h : \mathbb{R}^d \mapsto \mathbb{R}$ where exactly p components are non-zero ($1 \leq p \leq d$). We will study the generalization error obtain using ERM when increasing p (which is regarded as the class complexity). The class of predictors $\mathcal{H}_p$ is defined as follow:

Definition 17. *For $p \in [\![1, d]\!]$, $\mathcal{H}_p$ is the set of functions $h : \mathbb{R}^d \mapsto \mathbb{R}$ of the form:*

$$h(u) = u^T w, \quad \text{for } u \in \mathbb{R}^d$$

With $w \in \mathbb{R}^d$ having exactly p non-zero elements.

Let $(X, \epsilon) \in \mathbb{R}^d \times \mathbb{R}$ be independent random variables with $X \sim \mathcal{N}(0, I)$ and $\epsilon \sim \mathcal{N}(0, \sigma^2)$.

Let $h^* \in \mathcal{H}_d$ and define the random variable $Y = h^*(X) + \sigma\epsilon = X^T w + \sigma\epsilon$, with $\sigma > 0$ with $w \in \mathbb{R}^d$ defined by h^*. We consider $(X_i, Y_i)_{i=1}^n$ n iid copies of (X, Y). We are interested in the following problem:

$$\min_{h \in \mathcal{H}_d} \mathbb{E}[(h(X) - Y)^2] \tag{5}$$

Let $\mathbf{X} \in \mathbb{R}^{n \times d}$ the random matrix which rows are the X_i^T and $\mathbf{Y} = (Y_1, .., Y_n)^T \in \mathbb{R}^n$. In the following we will assume that $\boldsymbol{X}$ is full row rank and that $n \ll d$. Applying empirical risk minimization we can write:

$$\min_{w \in \mathbb{R}^d} \frac{1}{2} \|\boldsymbol{X} w - \boldsymbol{Y}\|^2 \tag{6}$$

Definition 18 (Random p-submatrix/p-subvector[1])**.** *For any $(p, q) \in [\![1, d]\!]^2$ such that $p + q = d$ and matrix $\boldsymbol{A} \in \mathbb{R}^{n \times d}$ and column vector $v \in \mathbb{R}^d$, we will denote by $\boldsymbol{A_{\sim p}}$ (resp. $v_{\sim p}$) the sub-matrix (resp. sub-vector) obtained by randomly selecting a subset of p columns (resp. elements), and by $\boldsymbol{A_{\sim q}} \in \mathbb{R}^{n \times q}$ and $v_{\sim q} \in \mathbb{R}^q$ their discarded counterpart.*

In order to solve (6) we will search for a solution in $\mathcal{H}_p \subset \mathcal{H}_d$ and increase p progressively which is a form of structural empirical risk minimization as $\mathcal{H}_p \subset \mathcal{H}_{p+1}$ for any $p < d$.

Let $p \in [\![1, d]\!]$, we are then interested in the following sub-problem:

$$\min_{w \in \mathbb{R}^p} \frac{1}{2} \|\boldsymbol{X_{\sim p}} w - y\|^2$$

We have seen in proposition 12 of section 2.2.1 that the least norm solution is $\hat{w}_{\sim p} = \boldsymbol{X}^+_{\boldsymbol{\sim p}} y$. If we define $\hat{w}_{\sim q} := 0$ then we will consider as a solution of the global problem (5) $\hat{w} := \phi_p(\hat{w}_{\sim p}, \hat{w}_{\sim q})$ where $\phi_p : \mathbb{R}^p \times \mathbb{R}^q \mapsto \mathbb{R}^d$ is a map rearranging the terms of $\hat{w}_{\sim p}$ and $\hat{w}_{\sim q}$ to match the initial indices of w.

Theorem 19. *Let $(x, \epsilon) \in \mathbb{R}^d \times \mathbb{R}$ independent random variables with $x \sim \mathcal{N}(0, I)$ and $\epsilon \sim \mathcal{N}(0, \sigma^2)$, and $w \in \mathbb{R}^d$. we assume that the response variable y is defined as $y = x^T w + \sigma\epsilon$. Let $(p, q) \in [\![1, d]\!]^2$ such that $p + q = d$, $\boldsymbol{X_{\sim p}}$ the randomly selected p columns sub-matrix of X. Defining $\hat{w} := \phi_p(\hat{w}_{\sim p}, \hat{w}_{\sim q})$ with $\hat{w}_{\sim p} = \boldsymbol{X}^+_{\boldsymbol{\sim p}} y$ and $\hat{w}_{\sim q} = 0$. The risk of the predictor associated to $\hat{w}$ is:*

$$\mathbb{E}[(y - x^T \hat{w})^2] = \begin{cases} (\|w_{\sim q}\|^2 + \sigma^2)(1 + \frac{p}{n-p-1}) & \text{if } p \leq n-2 \\ +\infty & \text{if } n-1 \leq p \leq n+1 \\ \|w_{\sim p}\|^2 (1 - \frac{n}{p}) + (\|w_{\sim q}\|^2 + \sigma^2)(1 + \frac{n}{p-n-1}) & \text{if } p \geq n+2 \end{cases}$$

Proof. Because x is zero mean and identity covariance matrix, and because x and ϵ are independent:

$$\mathbb{E}[(y - x^T \hat{w})^2] = \mathbb{E}[(x^T(w - \hat{w}) + \sigma\epsilon)^2] = \sigma^2 + \mathbb{E}[\|w - \hat{w}\|^2] = \sigma^2 + \mathbb{E}[\|w_{\sim p} - \hat{w}_{\sim p}\|^2] + \mathbb{E}[\|w_{\sim q} - \hat{w}_{\sim q}\|^2]$$

and because $\hat{w}_{\sim q} = 0$, we have: $\mathbb{E}[(y - x^T \hat{w})^2] = \sigma^2 + \mathbb{E}[\|w_{\sim p} - \hat{w}_{\sim p}\|^2] + \|w_{\sim q}\|^2$

[1] The notation used for the random p-submatrix and random p-subvector is not common and is introduced for clarity.

The classical regime ($p \leq n$) as been treated in [7]. We will then consider the interpolating regime ($p \geq n$). Recall that X is assumed to be of rank n. Let $\eta = y - \boldsymbol{X}_{\sim \boldsymbol{p}} w_{\sim p}$. We can write :

$$w_{\sim p} - \hat{w}_{\sim p} = w_{\sim p} - \boldsymbol{X}^{+}_{\sim \boldsymbol{p}} y = w_{\sim p} - \boldsymbol{X}^{+}_{\sim \boldsymbol{p}}(\eta + \boldsymbol{X}_{\sim \boldsymbol{p}} w_{\sim p}) = (\boldsymbol{I} - \boldsymbol{X}^{+}_{\sim \boldsymbol{p}} \boldsymbol{X}_{\sim \boldsymbol{p}}) w_{\sim p} - \boldsymbol{X}^{+}_{\sim \boldsymbol{p}} \eta$$

It is easy to show (left as an exercise) that $(\boldsymbol{I} - \boldsymbol{X}^{+}_{\sim \boldsymbol{p}} \boldsymbol{X}_{\sim \boldsymbol{p}})$ is the matrix of the orthogonal projection on $\mathrm{Ker}(\boldsymbol{X}_{\sim \boldsymbol{p}})$. Furthermore, $-\boldsymbol{X}^{+}_{\sim \boldsymbol{p}} \eta \in \mathrm{Im}(\boldsymbol{X}^{+}_{\sim \boldsymbol{p}}) = \mathrm{Im}(\boldsymbol{X}^{T}_{\sim \boldsymbol{p}})$. Then $(\boldsymbol{I} - \boldsymbol{X}^{+}_{\sim \boldsymbol{p}} \boldsymbol{X}_{\sim \boldsymbol{p}}) w_{\sim p}$ and $-\boldsymbol{X}^{+}_{\sim \boldsymbol{p}} \eta$ are orthogonal and the Pythagorean theorem gives:

$$\|w_{\sim p} - \hat{w}_{\sim p}\|^2 = \left\|(\boldsymbol{I} - \boldsymbol{X}^{+}_{\sim \boldsymbol{p}} \boldsymbol{X}_{\sim \boldsymbol{p}}) w_{\sim p}\right\|^2 + \left\|\boldsymbol{X}^{+}_{\sim \boldsymbol{p}} \eta\right\|^2$$

We will treat each term of the right hand side of this equality separately.

- $\underline{\left\|(\boldsymbol{I} - \boldsymbol{X}^{+}_{\sim \boldsymbol{p}} \boldsymbol{X}_{\sim \boldsymbol{p}}) w_{\sim p}\right\|^2}$: $\boldsymbol{X}^{+}_{\sim \boldsymbol{p}} \boldsymbol{X}_{\sim \boldsymbol{p}}$ is the matrix of the orthogonal projection on $\mathrm{Im}(\boldsymbol{X}^{T}_{\sim \boldsymbol{p}}) = \mathrm{Im}(\boldsymbol{X}^{+}_{\sim \boldsymbol{p}})$, then using again the Pythagorean theorem gives:

  $$\left\|(\boldsymbol{I} - \boldsymbol{X}^{+}_{\sim \boldsymbol{p}} \boldsymbol{X}_{\sim \boldsymbol{p}}) w_{\sim p}\right\|^2 = \|w_{\sim p}\|^2 - \left\|\boldsymbol{X}^{+}_{\sim \boldsymbol{p}} \boldsymbol{X}_{\sim \boldsymbol{p}} w_{\sim p}\right\|^2$$

  Because $\boldsymbol{X}^{+}_{\sim \boldsymbol{p}} \boldsymbol{X}_{\sim \boldsymbol{p}}$ is the matrix of the orthogonal projection on $\mathrm{Im}(\boldsymbol{X}^{T}_{\sim \boldsymbol{p}})$ we can write $\boldsymbol{X}^{+}_{\sim \boldsymbol{p}} \boldsymbol{X}_{\sim \boldsymbol{p}} w_{\sim p}$ as a linear combination of rows of X_p, then using the fact that the x_i are i.i.d and of standard normal distribution we have:

  $$\mathbb{E}[\left\|\boldsymbol{X}^{+}_{\sim \boldsymbol{p}} \boldsymbol{X}_{\sim \boldsymbol{p}} w_{\sim p}\right\|^2] = \|w_{\sim p}\|^2 \frac{n}{p} \quad \text{then} \quad \mathbb{E}[\left\|(\boldsymbol{I} - \boldsymbol{X}^{+}_{\sim \boldsymbol{p}} \boldsymbol{X}_{\sim \boldsymbol{p}}) w_{\sim p}\right\|^2] = \|w_{\sim p}\|^2 (1 - \frac{n}{p})$$

- $\underline{\left\|\boldsymbol{X}^{+}_{\sim \boldsymbol{p}} \eta\right\|^2}$: The calculation of this term used the "trace trick" and the notion of distribution of inverse-Wishart for pseudo-inverse matrices and is beyond the scope of this tutorial. It can be shown that:

  $$\mathbb{E}[\left\|\boldsymbol{X}^{+}_{\sim \boldsymbol{p}} \eta\right\|^2] = \begin{cases} (\|w_{\sim q}\|^2 + \sigma^2)(\frac{n}{p-n-1}) & \text{if } p \geq n+2 \\ +\infty & \text{if } p \in \{n, n+1\} \end{cases}$$

□

Corollary 1. *Let T be a uniformly random subset of $[\![1, d]\!]$ of cardinality p. Under the setting of Theorem 19 and taking the expectation with respect to T. The risk of the predictor associated to $\hat{w}$ is:*

$$\mathbb{E}[(Y - X^T \hat{w})^2] = \begin{cases} \left((1 - \frac{p}{d})\|w\|^2 + \sigma^2\right)(1 + \frac{p}{n-p-1}) & \text{if } p \leq n-2 \\ \|w\|^2 \left(1 - \frac{n}{d}(2 - \frac{d-n-1}{p-n-1})\right) + \sigma^2(1 + \frac{n}{p-n-1}) & \text{if } p \geq n+2 \end{cases}$$

Proof. Since T is a uniformly random subset of $[\![1, d]\!]$ of cardinality p:

$$\mathbb{E}[\|w_{\sim p}\|^2] = \mathbb{E}[\sum_{i \in T} w_i^2] = \mathbb{E}[\sum_{i=1}^{d} w_i^2 \mathbb{1}_T(i)] = \sum_{i=1}^{d} w_i^2 \mathbb{E}[\mathbb{1}_T(i)] = \sum_{i=1}^{d} w_i^2 \mathbb{P}[i \in T] = \|w\|^2 \frac{p}{d}$$

and, similarly:

$$\mathbb{E}[\|w_{\sim q}\|^2] = \|w\|^2 (1 - \frac{p}{d})$$

Plugging into Theorem 19 ends the proof. □

3.2 Random Fourier Features

In this section we consider the RFF model family [22] as our class of predictors $\mathcal{H}_N$.

Definition 20. *We call* Random Fourier Features (RFF) *model any function* $h : \mathbb{R}^d \to \mathbb{R}$ *of the form :*

$$h(x) = \beta^T z(x)$$

With $z(x) = \sqrt{\frac{2}{N}} \begin{bmatrix} \cos\left(\omega_1^T x + b_1\right) \\ \vdots \\ \cos\left(\omega_N^T x + b_N\right) \end{bmatrix}$, $\beta \in \mathbb{R}^N$ *the parameters of the model and* $\forall i \in [\![1, N]\!] \begin{cases} \omega_i \sim \mathcal{N}(0, \sigma^2 I_d) \\ b_i \sim \mathcal{U}([0, 2\pi]) \end{cases}$. *The vectors* ω_i *and the scalars* b_i *are sampled before fitting the model, and* z *is called a* randomized map.

The RFF family is a popular class of models that are linear w.r.t. the parameters β but non-linear w.r.t. the input x, and can be seen as two-layer neural networks with fixed weights in the first layer. In a classification setting, using these models with the hinge loss amounts to fitting a linear SVM to n feature vectors (of dimension N). RFF models are typically used to approximate the Gaussian kernel and reduce the computational cost when $N \ll n$ (e.g. kernel ridge regression when using the squared loss and a l_2 regularization term). In our case however, we will go beyond $N = n$ to observe the double descent phenomenon.

Remark 7. *Clearly, we have* $\mathcal{H}_N \subset \mathcal{H}_{N+1}$ *for any* $N \geq 0$.

Proposition 21 (Approximation of the Gaussian Kernel, informal)**.** *Let* $k : (x, y) \to e^{-\frac{1}{2\sigma^2}||x-y||^2}$ *be the Gaussian kernel on* $\mathbb{R}^d$*, and let* $\mathcal{H}_\infty$ *be a class of predictors where empirical risk minimizers on* $\mathcal{D}_n = \{(x_1, y_1), \ldots, (x_n, y_n)\}$ *can be expressed as* $h : x \to \sum_{k=1}^n \alpha_k k(x_k, x)$*. Then, as* $N \to \infty$*,* $\mathcal{H}_N$ *becomes a closer and closer approximation of* $\mathcal{H}_\infty$.

Proof sketch. For any $x, y \in \mathbb{R}^d$, with the vectors $\omega_k \in \mathbb{R}^d$ sampled from $\mathcal{N}(0, \sigma^2 I_d)$:

$$\begin{aligned} k(x, y) = e^{-\frac{1}{2\sigma^2}(x-y)^T(x-y)} &\overset{(1)}{=} \mathbb{E}_{\omega \sim \mathcal{N}(0, \sigma^2 I_d)}[e^{i\omega^T(x-y)}] \overset{(2)}{=} \mathbb{E}_{\omega \sim \mathcal{N}(0, \sigma^2 I_d)}[\cos\left(\omega^T(x-y)\right)] \\ &\approx \frac{1}{N}\sum_{k=1}^N \cos\left(\omega_k^T(x-y)\right) = \frac{1}{N}\sum_{k=1}^N 2\cos\left(\omega_k^T x + b_k\right)\cos\left(\omega_k^T y + b_k\right) \overset{(3)}{=} z(x)^T z(y) \end{aligned}$$

Where (1) and (3) are left as an exercise, with indications in [13] if needed, and (2) is because $k(x, y) \in \mathbb{R}$.

Hence, for $h \in \mathcal{H}_\infty$: $h(x) = \sum_{k=1}^n \alpha_k k(x_n, x) \approx \underbrace{\left(\sum_{k=1}^N \alpha_k z(x_k)\right)^T}_{\beta} z(x)$ ∘

A complete definition is outside of the scope of this tutorial, but $\mathcal{H}_\infty$ is actually the *Reproducing Kernel Hilbert Space (RKHS)* corresponding to the Gaussian kernel, for which RFF models are a good approximation when sampling the random vectors ω_i from a normal distribution.

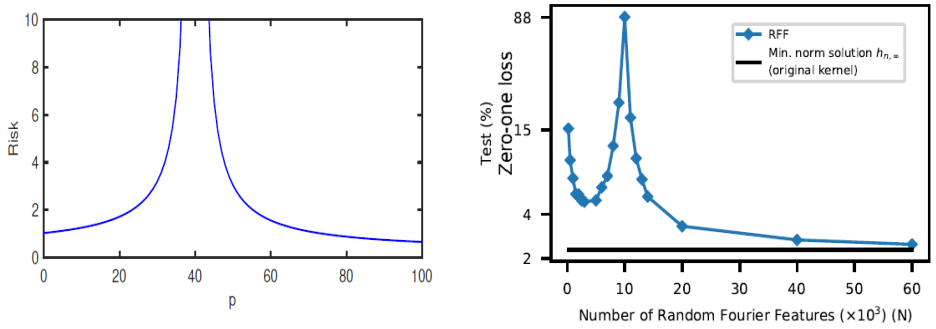
We use ERM to find the predictor $h_{n,N} \in \mathcal{H}_N$ and, in the interpolation regime where multiple minimizers exist, we choose the one whose parameters $\beta \in \mathbb{R}^N$ have the smallest l_2 norm. This training procedure allows us to observe a model-wise double descent (figure 5b). Indeed, in the under-parameterized regime, statistical analyses suggest choosing $N \propto \sqrt{n}\log(n)$ for good test risk guarantees [22]. And as we approach the interpolation point (around $N = n$), we observe that the test risk increases then decreases again.

In the over-parameterized regime ($N \geq n$), multiple predictors are able to fit perfectly the training data. As $\mathcal{H}_N \in \mathcal{H}_{N+1}$, increasing N leads to richer model classes and allows constructing interpolating predictors that are more regular, with smaller norm (eventually converging to $h_{n,\infty}$ obtained from $\mathcal{H}_\infty$). As detailed in theorem 22 (in a noiseless setting), the small norm inductive bias is indeed powerful to ensure small generalization error.

Theorem 22 (Belkin et al. [4]). *Fix any $h^* \in \mathcal{H}_\infty$. Let $(X_1, Y_1), \ldots, (X_n, Y_n)$ be i.i.d. random variables, where X_i is drawn uniformly at random from a compact cube $\Omega \in \mathbb{R}^d$, and $Y_i = h^*(X_i)$ for all i. There exists constants $A, B > 0$ such that, for any interpolating $h \in \mathcal{H}_\infty$ (i.e., $h(X_i) = Y_i$ for all i), so that with high probability :*

$$\sup_{x \in \Omega} |h(x) - h^*(x)| < Ae^{-B(n/logn)^{1/d}}(||h^*||_{\mathcal{H}_\infty} + ||h||_{\mathcal{H}_\infty})$$

Proof. We refer the reader directly to [4] for the proof. □



(a) Plot of risk $\mathbb{E}[(y - x^T\hat{w})^2]$ as a function of p, under the random selection model of the subset of p features. Here $\|w\|^2 = 1$, $\sigma^2 = 1/25$, $d = 100$ and $n = 40$. Taken from [5]

(b) Model-wise double descent risk curve for RFF model on a subset of MNIST ($n = 10^4$, 10 classes), *choosing the smallest norm predictor* $h_{n,N}$ when $N > n$. The interpolation threshold is achieved at $N = 10^4$. Taken from [4], which uses an equivalent but slightly different definition of RFF models.

Figure 5: Risk curves as a function of model capacity.

3.3 Related works

3.3.1 Optimization in the over-parameterized regime

For reasons that are still under investigation, overparameterization seems beneficial not only in the statistical learning framework, but from an optimization standpoint as well as it facilitates convergence to global minima, in particular with the gradient descent procedures.

The optimization problem can be framed as minimizing a certain loss function $\mathcal{L}(w)$ with respect to its parameters $w \in \mathbb{R}^N$, such as the square loss $\mathcal{L}(w) = \frac{1}{2}\sum_{i=1}^{n}(f(x_i, w) - y_i)^2$ where $\{(x_i, y_i)\}_{i=1}^n$ is our given training dataset and $f : (\mathbb{R}^d \times \mathbb{R}^N) \to \mathbb{R}$ is our model.

Exercise 2. *Assume that $\ell : \mathcal{Y} \to \mathbb{R}$ is convex and $f : \mathcal{X} \to \mathcal{Y}$ is linear. Show that $\ell \circ f$ is convex.*

When f is non-linear however (which is habitually the case in deep learning) the landscape of the loss function is generally non-convex. Therefore, first order methods such as GD or SGD are likely to converge and get trapped in spurious local minima, depending on the initialization. Yet, in the over-parameterized regime where there are multiple global minima interpolating almost perfectly the data, it seems that SGD has no problem converging to these solutions, despite the highly non-convex setting. Recent works are trying to explain this phenomenon.

For instance, [20] shows that, for one-hidden layer neural networks that (1) have smooth activation functions, (2) are over-parameterized, i.e. $N \geq Cn^2$ where C depends on the distribution of the data and (3) are initialized with i.i.d. $\mathcal{N}(0, 1)$ entries, then with high probability GD converges quickly to a global optimum. Similar results also hold for ReLU activation functions and for SGD.

In [14], the authors show that sufficiently over-parameterized systems, including wide neural networks, generally satisfy a condition that allows gradient descent to converge efficiently, for a broad class of problems. They use the PL-condition (from Polyak and Lojasiewicz [21]) which does not require convexity but is sufficient for efficient minimization by GD. One key point is that the loss function $\mathcal{L}(w)$ is generally non-convex in the neighborhood of minimizers. Due to the over-parameterization, the Hessian matrices $\nabla^2\mathcal{L}(w)$ are positive semi-definite but not positive definite in these neighborhoods, which is incompatible with convexity for non-linear sets of solutions. This is in contrast to the under-parameterized landscape which generally has multiple isolated local minima with positive definite Hessian matrices. Figure 6 illustrates this.

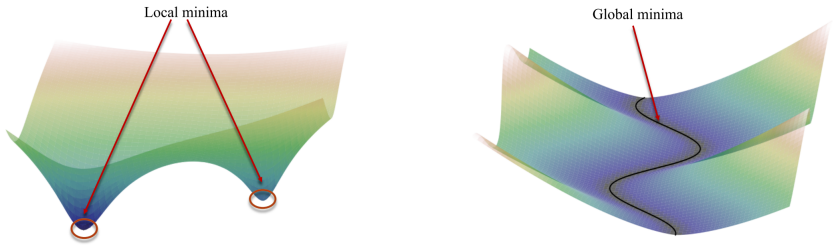


Figure 6: *Left :* Loss landscape of under-parameterized models, locally convex at local minima. *Right :* Loss landscape of over-parameterized models, incompatible with local convexity. Taken from [14].

In addition to better convergence guarantees, over-parameterization can even accelerate optimization. By working with *linear* neural networks (hence fixed expressiveness), [1] finds

that increasing depth has an implicit effect on gradient descent, combining certain forms of *momentum* and *adaptive learning rates* (two well-known tools in the field of optimization). They observe the acceleration for non-linear networks as well (replacing weight matrices by a product of matrices, for fixed expressiveness), and even when using explicit acceleration methods such as Adam.

3.3.2 Neural networks as a physical system : the jamming transition

In order to study the loss landscape, [24] make an analogy between neural networks and complex physical systems with non-convex energy landscape, called glassy systems. Indeed, the loss function can be interpreted as the potential energy of the system f, with a large number of parameters N (degrees of freedom). By considering the hinge loss, the minimization of $\mathcal{L}(w; \mathcal{D}_n)$ actually amounts to a constraint-classification problem (with n constraints, N continuous degrees of freedom), already studied in physics.

Using this analogy, they show that the behavior of deep networks near the interpolation point is similar to the behavior of some granular systems, that undergo a critical *jamming transition* when their density increases such that they are forced to be in contact one another. In the under-parameterized regime, not all the training examples can be classified correctly, which leads to unsatisfied constraints. But in the over-parameterized regime, there is no stable local minima : the network reaches a global minima zero training loss.

As illustrated in figure 7, the authors are able to quantify the location of the jamming transition in the (n, N) plane (considering N as the *effective* number of parameters of the network). Considering a fully-connected network with arbitrary depth, ReLU activation functions and a dataset of size n, they give a linear upper bound on the critical number of parameters N^* characterizing the jamming transition : $N^* \leq \frac{1}{C_0} n$ where C_0 is a constant. In their experiments, it seems that the bound is tight for random data but that N^* increases sub-linearly with n for structured data (e.g. MNIST), as illustrated on figure 7.

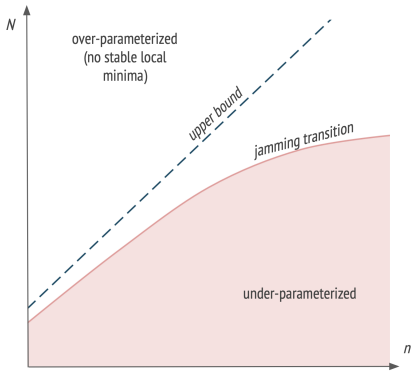


Figure 7: N : degrees of freedom, n : training examples. Inspired from [24]

Similarly to other works, they observe a peak in test error at the jamming transition. In [10], using the same setting of fixed-depth fully-connected networks, they argue that this may be due to $||f||$ diverging near the interpolation point $N = N^*$. Interestingly, they also observe that near-optimal generalization can be obtained using an ensemble average of networks with N slightly beyond N^*.

4 Conclusion

From a statistical learning point of view, deep learning is a challenging setting to study and some recent empirical successes are not yet well understood. The double descent phenomenon, arising from well-chosen inductive biases in the over-parameterized regime, has been studied in linear settings and observed with deep networks [18].

In addition to the references presented in section 3.3, other lines of work seem promising. Notably, [11][19][23][12] are working towards a better understanding of the implicit bias induced by optimization algorithms. Finally, we refer the reader to subsequent works of Belkin *et al.* such as [9], that finds *multiple descent* curves with an arbitrary number of peaks, due to the interaction between the properties of the data and the inductive biases of learning algorithms.

References

[1] S. Arora, N. Cohen, and E. Hazan. On the optimization of deep networks: Implicit acceleration by overparameterization. *35th International Conference on Machine Learning, ICML 2018*, 1:372–389, 2018. URL https://arxiv.org/pdf/1802.06509.pdf. 18

[2] P. W. Battaglia, J. B. Hamrick, V. Bapst, A. Sanchez-Gonzalez, V. Zambaldi, M. Malinowski, A. Tacchetti, D. Raposo, A. Santoro, R. Faulkner, C. Gulcehre, F. Song, A. Ballard, J. Gilmer, G. Dahl, A. Vaswani, K. Allen, C. Nash, V. Langston, C. Dyer, N. Heess, D. Wierstra, P. Kohli, M. Botvinick, O. Vinyals, Y. Li, and R. Pascanu. Relational inductive biases, deep learning, and graph networks. *arXiv:1806.01261 [cs, stat]*, Oct. 2018. URL http://arxiv.org/abs/1806.01261. arXiv: 1806.01261. 6

[3] 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 of the United States of America*, 116(32):15849–15854, 2019. ISSN 10916490. doi: 10.1073/pnas.1903070116. 4, 5

[4] M. Belkin, D. Hsu, S. Ma, and S. Mandal. Reconciling modern machine learning practice and the bias-variance trade-off. *arXiv:1812.11118 [cs, stat]*, Sept. 2019. URL http://arxiv.org/abs/1812.11118. arXiv: 1812.11118. 1, 6, 17

[5] M. Belkin, D. Hsu, and J. Xu. Two models of double descent for weak features. *arXiv:1903.07571 [cs, stat]*, Oct. 2020. URL http://arxiv.org/abs/1903.07571. arXiv: 1903.07571. 17

[6] C. M. Bishop. *Pattern recognition and machine learning*. Information science and statistics. Springer, New York, 2006. ISBN 978-0-387-31073-2. 12

[7] L. Breiman and D. Freedman. How Many Variables Should be Entered in a Regression Equation? *Journal of the American Statistical Association*, 78(381):131– 136, 1983. URL http://www.jstor.org/stable/2287119. 15

[8] A. Canziani, A. Paszke, and E. Culurciello. An Analysis of Deep Neural Network Models for Practical Applications. pages 1–7, 2016. URL http://arxiv.org/abs/1605.07678. 4

[9] L. Chen, Y. Min, M. Belkin, and A. Karbasi. Multiple descent: Design your own generalization curve. *arXiv*, 2020. ISSN 23318422. URL https://arxiv.org/pdf/2008.01036.pdf. 20

[10] M. Geiger, A. Jacot, S. Spigler, F. Gabriel, L. Sagun, S. D'Ascoli, G. Biroli, C. Hongler, and M. Wyart. Scaling description of generalization with number of parameters in deep learning. *Journal of Statistical Mechanics: Theory and Experiment*, 2020(2), 2020. ISSN 17425468. doi: 10.1088/1742-5468/ab633c. URL https://arxiv.org/pdf/1901.01608.pdf. 19

[11] D. Gissin, S. Shalev-Shwartz, and A. Daniely. The implicit bias of depth: How incremental learning drives generalization. *arXiv*, (2017):1–25, 2019. ISSN 23318422. URL https://arxiv.org/pdf/1909.12051.pdf. 12, 20

[12] S. Gunasekar, B. Woodworth, S. Bhojanapalli, B. Neyshabur, and N. Srebro. Implicit Regularization in Matrix Factorization. *arXiv:1705.09280 [cs, stat]*, May 2017. URL http://arxiv.org/abs/1705.09280. arXiv: 1705.09280. 20

[13] G. Gundersen. Random fourier features, 2019. URL http://gregorygundersen.com/blog/2019/12/23/random-fourier-features/. 16

[14] C. Liu, L. Zhu, and M. Belkin. Toward a theory of optimization for over-parameterized systems of non-linear equations: the lessons of deep learning. *arXiv*, (1):1–40, 2020. ISSN 23318422. URL https://arxiv.org/pdf/2003.00307.pdf. 18

[15] T. M. Mitchell. The need for biases in learning generalizations. Technical report, Rutgers University, New Brunswick, NJ, 1980. URL http://dml.cs.byu.edu/~cgc/docs/mldm_tools/Reading/Need%20for%20Bias.pdf. 6

[16] P. Nakkiran, G. Kaplun, Y. Bansal, T. Yang, B. Barak, and I. Sutskever. Deep Double Descent: Where Bigger Models and More Data Hurt. pages 1–24, 2019. URL http://arxiv.org/abs/1912.02292. 4, 5

[17] P. Nakkiran, G. Kaplun, Y. Bansal, T. Yang, B. Barak, and I. Sutskever. Windows on theory - deep double descent, 2019. URL https://windowsontheory.org/2019/12/05/deep-double-descent/. 13

[18] P. Nakkiran, G. Kaplun, Y. Bansal, T. Yang, B. Barak, and I. Sutskever. Deep Double Descent: Where Bigger Models and More Data Hurt. *arXiv:1912.02292 [cs, stat]*, Dec. 2019. URL http://arxiv.org/abs/1912.02292. arXiv: 1912.02292. 20

[19] B. Neyshabur, R. Salakhutdinov, and N. Srebro. Path-SGD: Path-normalized optimization in deep neural networks. *Advances in Neural Information Processing Systems*, 2015-January:2422–2430, 2015. ISSN 10495258. URL https://arxiv.org/pdf/1506.02617.pdf. 20

[20] S. Oymak and M. Soltanolkotabi. Towards moderate overparameterization: global convergence guarantees for training shallow neural networks. *arXiv*, 2019. ISSN 23318422. doi: 10.1109/jsait.2020.2991332. URL https://arxiv.org/pdf/1902.04674.pdf. 18

[21] B. T. Polyak. Gradient methods for minimizing functionals. *Zhurnal Vychislitel'noi Matematiki i Matematicheskoi Fiziki*, 3(4):643–653, 1963. 18

[22] A. Rahimi and B. Recht. Random features for large-scale kernel machines. *Advances in Neural Information Processing Systems 20 - Proceedings of the 2007 Conference*, (1):1–8, 2009. URL https://papers.nips.cc/paper/2007/file/013a006f03dbc5392effeb8f18fda755-Paper.pdf. 16, 17

[23] D. Soudry, E. Hoffer, M. S. Nacson, S. Gunasekar, and N. Srebro. The Implicit Bias of Gradient Descent on Separable Data. *arXiv:1710.10345 [cs, stat]*, Dec. 2018. URL http://arxiv.org/abs/1710.10345. arXiv: 1710.10345. 11, 20

[24] S. Spigler, M. Geiger, S. d'Ascoli, L. Sagun, G. Biroli, and M. Wyart. A jamming transition from under- to over-parametrization affects loss landscape and generalization. *J. Phys. A: Math. Theor.*, 52(47):474001, Nov. 2019. ISSN 1751-8113, 1751-8121. doi: 10.1088/1751-8121/ab4c8b. URL http://arxiv.org/abs/1810.09665. arXiv: 1810.09665. 19

[25] V. Vapnik. Principles of risk minimization for learning theory. *Advances in neural information processing systems*, pages 831–838, 1992. URL http://papers.nips.cc/paper/506-principles-of-risk-minimization-for-learning-theory. 2