Mathematical Foundations of Machine Learning

(growing lecture notes)

Michael M. Wolf

July 17, 2016

Contents

Introduction

What follows are notes on the lecture course "Mathematical Foundations of Machine Learning" given at TUM in summer 2016. The notes (hopefully) grow as the lecture course advances. If you spot errors or typos, let me know...

What is machine learning and what is this course about?

Machine learning can be considered as part of the field of artificial intelligence, which in turn may largely be regarded as a subfield of computer science. The aim of machine learning is to exploit data in order to devise complex models or algorithms in an automated way. So machine learning is typically used whenever large amounts of data are available and when one aims at a computer program that is (too) difficult to program 'directly'. Standard examples are programs that recognize faces, handwriting or speech, drive cars or play Go. These are hard to program from scratch so that one uses machine learning algorithms that produce such programs from large amounts of data.

Two main branches of the field are *supervised learning* and *unsupervised learning*. In supervised learning a learner is a device that receives 'labeled training data' as input and outputs a program that predicts the label for unseen instances and thus generalizes beyond the training data. Examples of sets of labeled data are emails that are labeled 'spam' or 'no spam' and medical histories that are labeled with the occurrence or absence of a certain disease. In these cases the learner's output would be a spam filter and a diagnostic program, respectively.

In contrast, in unsupervised learning there is no additional label attached to the data and the task is to identify and model hidden patterns in the data. Hence, unsupervised learning is primarily descriptive whereas supervised learning is more predictive. In this course, we will exclusively deal with the latter.

A first coarse classification of supervised learners is in terms of the chosen *representation*, which determines the basic structure of the generated programs. Common ones are:

- Decision trees
- k-nearest neighbors
- Neural networks
- Support vector machines

We will focus on the latter two, which are arguably the most sophisticated and most powerful classes of representations used today.

The lecture will be a first course concentrating on the mathematical aspects of the subject. We will begin with the framework of statistical learning theory and then have a closer look at neural networks and support vector machines and kernel methods.

Chapter 1

Learning Theory

1.1 Statistical framework

Here we set up the standard statistical framework for supervised learning theory.

Input of the learner is the *training data* that is a finite sequence $S = \big((x_1, y_1), \ldots, (x_n, y_n)\big)$ of pairs from $\mathcal{X} \times \mathcal{Y}$.

Output of the learner is a *hypothesis* $h : \mathcal{X} \to \mathcal{Y}$ that aims at predicting $y \in \mathcal{Y}$ for arbitrary $x \in \mathcal{X}$, especially for those not contained in the training data. Formally, the learner can thus be seen as a map $\cup_{n\in\mathbb{N}}(\mathcal{X}\times\mathcal{Y})^n \to \mathcal{Y}^{\mathcal{X}} : S \mapsto h_S$. We will denote its range, i.e., the set of functions that can be output and thus represented by the learner, by $\mathcal{F}$. From a computer science perspective the learner is an algorithm that, upon input of the training data S, outputs a computer program described by $h \in \mathcal{F}$.

Probabilistic assumption. The pairs (x_i, y_i) are treated as values of random variables (X_i, Y_i) that are identically and independently distributed according to some probability measure P over $\mathcal{X} \times \mathcal{Y}$. We will throughout assume that the corresponding σ-algebra is a product of Borel σ-algebras w.r.t. the usual topologies. All considered functions will be assumed to be Borel functions. Expectations w.r.t. P and P^n will be denoted by $\mathbb{E}$ and $\mathbb{E}_S$.

Goal of the learner is to find a good hypothesis h w.r.t. a suitably chosen *loss function* $L : \mathcal{Y} \times \mathcal{Y} \to \mathbb{R}$ that measures how far $h(x)$ is from the respective y. The smaller the average loss, called *risk* or *generalization error* and given by

$$R(h) := \int_{\mathcal{X}\times\mathcal{Y}} L\big(y, h(x)\big) dP(x, y), \tag{1.1}$$

the better the hypothesis. The challenge is, that the probability measure P is unknown, only the training data S is given. Hence, the learner's task is to minimize the risk without being able to evaluate it directly.

Depending on whether $\mathcal{Y}$ is continuous or discrete one distinguishes two types of learning problems with different loss functions:

Regression

Regression deals with continuous $\mathcal{Y}$. The most common loss function in the case $\mathcal{Y} = \mathbb{R}$ is the quadratic loss $L\big(y, h(x)\big) = |y - h(x)|^2$ leading to the L_2-*risk* also known as *mean squared error* $R(h) = \mathbb{E}\left[|Y - h(X)|^2\right]$. For many reasons this is a mathematically convenient choice. One of them is that the function that minimizes the risk can be handled:

Theorem 1.1: Regression function minimizes L_2-risk

In the present context let $h : \mathcal{X} \to \mathcal{Y} = \mathbb{R}$ be a Borel function and assume that $\mathbb{E}\left[Y^2\right]$ and $\mathbb{E}\left[h(X)^2\right]$ are both finite. Define the *regression function* as conditional expectation $r(x) := \mathbb{E}(Y|X = x)$. Then the L_2-risk of h can be written as

$$R(h) = \mathbb{E}\left[|Y - r(X)|^2\right] + \mathbb{E}\left[|h(X) - r(X)|^2\right]. \tag{1.2}$$

Note: The first term on the r.h.s. in Eq.(1.2) vanishes if there is a deterministic relation between x and y, i.e., if $P(y|x) \in \{0, 1\}$. In general, it can be regarded as unavoidable inaccuracy that is due to noise or due to the lack of information content in X about Y. The second term contains the dependence on h and is simply the squared L_2-distance between h and the regression function r. Minimizing the risk thus means minimizing the distance to the regression function.

Proof. (sketch) Consider the real Hilbert space $L_2(\mathcal{X} \times \mathcal{Y}, P)$ with inner product $\langle \psi, \phi \rangle := \mathbb{E}\left[\psi\phi\right]$. h can be considered as an element of the closed subspace of functions that only depend on x and are constant w.r.t. y. The function r also represents an element of that subspace and since the conditional expectation[1] is, by construction, the orthogonal projection into that subspace, we have $\langle y - r, h - r \rangle = 0$. With this, Pythagoras' identity yields the desired result

$$||y - h||^2 = ||y - r||^2 + ||h - r||^2.$$

□

Classification

Classification deals with discrete $\mathcal{Y}$, in which case a function from $\mathcal{X}$ to $\mathcal{Y}$ is also called a *classifier*. The most common loss function in this scenario is $L(y, y') = 1 - \delta_{y,y'}$ so that the corresponding risk is nothing but the error probability $R(h) = \mathbb{P}\left[h(X) \neq Y\right] = \mathbb{E}\left[\mathbb{1}_{h(X) \neq Y}\right]$. We will mostly consider *binary classification* where $\mathcal{Y} = \{-1, 1\}$. The error probability in binary classification is minimized by the *Bayes classifier*

$$b(x) := \operatorname{sgn}\big(\mathbb{E}\left[Y|X = x\right]\big). \tag{1.3}$$

[1] If there is a probability density $p(x, y)$, the conditional expectation is given by $E(Y|X = x) = \int_{\mathcal{Y}} y\, p(x, y)/p(x) dy$, if the marginal $p(x)$ is non-zero. For a general treatment of conditional expectations see for instance [10], Chap.23.

1.2 Error decomposition

How can the learner attempt to minimize the risk/generalization error $R(h)$ over its accessible hypotheses $h \in \mathcal{F}$ without knowing the underlying distribution P? There are two helping hands. The first one is prior knowledge about P that gets encoded in the choice of $\mathcal{F}$ and the way the learner chooses a hypothesis from this class. Second, although $R(h)$ cannot be evaluated directly, the average loss can be evaluated on the data S, which leads to the *empirical risk*

$$\hat{R}(h) := \frac{1}{n} \sum_{i=1}^{n} L\big(y_i, h(x_i)\big). \tag{1.4}$$

The approach of minimizing $\hat{R}$ is called *empirical risk minimization* (ERM). In particular if $|\mathcal{Y}| < \infty$, then there always exists a minimizer $\hat{h} \in \mathcal{F}$ that attains $\inf_{h\in\mathcal{F}} \hat{R}(h) = \hat{R}(\hat{h})$ since the functions are only evaluated at a finite number of points, which effectively restricts $\mathcal{F}$ to a finite space.

If n is sufficiently large, one might hope that $\hat{R}(h)$ is not too far from $R(h)$ so that ERM would come close to minimizing the risk. The formalization and quantification of this hope is essentially the content of the remaining part of this chapter.

To this end, and also for a better understanding of some of the main issues in machine learning, it is useful to look at the following error decompositions.

Let $R_b := \inf_h R(h)$ be the *Bayes risk*, where the infimum is taken over all measurable functions $h : \mathcal{X} \to \mathcal{Y}$, and let $R_{\mathcal{F}} := \inf_{h\in\mathcal{F}} R(h)$ quantify the optimal performance of a learner capable of representing $\mathcal{F}$. Then we can decompose the difference between the risk of a hypothesis and the optimal Bayes risk as

$$R(h) - R_b = \underbrace{\big(R(h) - R_{\mathcal{F}}\big)}_{\text{estimation error}} + \underbrace{\big(R_{\mathcal{F}} - R_b\big)}_{\text{approximation error}}. \tag{1.5}$$

The *approximation error* does neither depend on the hypothesis nor on the data. It quantifies how well the hypothesis class $\mathcal{F}$ is suited for the problem under consideration. The *estimation error* measures how well the hypothesis h performs relative to best hypotheses in $\mathcal{F}$. If a suitable hypothesis is chosen, e.g. via ERM, the estimation error ideally decreases with the size n of the training data.

Assume that empirical risk minimization has led to a hypothesis $\hat{h} \in \mathcal{F}$ for which $\hat{R}(\hat{h}) \leq \hat{R}(h)\ \forall h \in \mathcal{F}$. Then the estimation error can be bounded by:

$$\begin{aligned} R\big(\hat{h}\big) - R_{\mathcal{F}} &= R\big(\hat{h}\big) - \hat{R}\big(\hat{h}\big) + \sup_{h\in\mathcal{F}} \Big(\hat{R}\big(\hat{h}\big) - R(h)\Big) \\ &\leq 2 \sup_{h\in\mathcal{F}} \Big|\hat{R}(h) - R(h)\Big|. \end{aligned} \tag{1.6}$$

Uniform bounds on the difference between the risk and the empirical risk will be derived in the following sections.

Usually, one is faced with a trade-off between the estimation error and the approximation error: while minimizing the approximation error suggests to take a richer hypothesis class $\mathcal{F}$, the data required to keep the estimation error under control unfortunately turns out to grow rapidly with the size of $\mathcal{F}$ (cf. following sections). A closely related trade-off runs under the name *bias-variance trade-off*. It has its origin in a refinement of the decomposition in Thm.1.1:

Theorem 1.2: Noise-bias-variance decomposition

In the setup of Thm.1.1 consider a fixed learner that outputs a hypothesis h_S upon input of $S \in (\mathcal{X}\times\mathcal{Y})^n$. Regard S as a random variable, distributed according to P^n and define $\bar{h}(x) := \mathbb{E}_S[h_S(x)]$ the expected prediction for a fixed x. If the expected risk $\mathbb{E}_S[R(h_S)]$ is finite, then it is equal to

$$\underbrace{\mathbb{E}\left[|Y - r(X)|^2\right]}_{\text{noise}} + \underbrace{\mathbb{E}\left[|\bar{h}(X) - r(X)|^2\right]}_{\text{bias}^2} + \underbrace{\mathbb{E}\left[\mathbb{E}_S\left[|h_S(X) - \bar{h}(X)|^2\right]\right]}_{\text{variance}} \tag{1.7}$$

Proof. We take the expectation $\mathbb{E}_S$ of Eq.(1.2) when applied to h_S and observe that the first term on the r.h.s. is independent of S. For the second term we obtain

$$\begin{aligned}
\mathbb{E}_S\left[\mathbb{E}\left[|h_S(X) - r(X)|^2\right]\right] &= \mathbb{E}\left[\mathbb{E}_S\left[|h_S(X) - \bar{h}(X) + \bar{h}(X) - r(X)|^2\right]\right] \\
&= \mathbb{E}\left[|\bar{h}(X) - r(X)|^2\right] \\
&\quad + \mathbb{E}\left[\mathbb{E}_S\left[|h_S(X) - \bar{h}(X)|^2\right]\right] \\
&\quad + 2\mathbb{E}\left[\mathbb{E}_S\left[\big(h_S(X) - \bar{h}(X)\big)\big(\bar{h}(X) - r(X)\big)\right]\right].
\end{aligned}$$

The term in the last line vanishes since $\big(\bar{h}(X) - r(X)\big)$ is independent of S and $\mathbb{E}_S\left[\big(h_S(X) - \bar{h}(X)\big)\right] = 0$. □

Again, if we increase the size of $\mathcal{F}$, then the squared bias is likely to decrease while the variance will increase (and the noise is unaffected).

There is a third incarnation of the phenomenon behind a dominating variance or estimation error that is called *overfitting*. All together these are consequences of choosing $\mathcal{F}$ too large so that it contains exceedingly complex hypotheses, which might be chosen by the learner. Formally, one defines a hypothesis $h \in \mathcal{F}$ to be *overfitting* if there exists a hypothesis $h' \in \mathcal{F}$ such that $\hat{R}(h) < \hat{R}(h')$, but $R(h) > R(h')$. That is, h overfits the data in the sense that the empirical error is overly optimistic.

The three closely related issues just discussed all ask for a balanced choice of $\mathcal{F}$. In order to achieve this and to get confidence in the quality of the choice many techniques have been developed. First of all, the available labeled data is split into two disjoint sets, *training data* and *test data*. While the former is used to train/learn/optimize and eventually output a hypothesis h_S, the latter is used to evaluate the performance of h_S. There is sometimes a third separate set, the *validation data*, that is used to tune free parameters of the learning

algorithm. In many cases, however, training data is too precious to set aside a separate validation sample and then validation is done on the training data by a technique called *cross-validation*.

In order to prevent the learner from choosing overly complex hypotheses, ERM is often modified in practice. One possibility, called *structural risk minimization*, is to consider a sequence of hypotheses classes $\mathcal{F}_1 \subset \mathcal{F}_2 \subset \mathcal{F}_3 \subset \dots$ of increasing complexity and to optimize the empirical error plus a penalty term that takes into account the complexity of the underlying class. A smoother variant of this idea is called *regularization*, where a large class $\mathcal{F}$ is chosen together with a *regularizer*, i.e., a complexity penalizing function, which is often a norm in cases where $\mathcal{F}$ has a vector space structure. Then one has to minimize the *regularized empirical risk* $\hat{R}(h) + \lambda||h||^2$, where $\lambda \in \mathbb{R}_+$ is a free parameter to be chosen, e.g. by cross-validation.

In the end, however, choosing a good class $\mathcal{F}$ and/or a good way to pick not too complex hypotheses from $\mathcal{F}$ is to some extent an art. It depends on the problem and on experience. In Sec.1.4 we will see a formalization of the fact that there is no a priori best choice.

1.3 PAC learning

In the light of the previous section and of Eq.(1.6) it is desirable to bound $|\hat{R}(h) - R(h)|$ uniformly w.r.t. $h \in \mathcal{F}$. As the empirical risk depends on the training data, which we assume to be a random sample, we have to take into account the possibility to be unlucky with the training data. What we can reasonably hope for, is that, under the right conditions, we obtain guarantees of the form

$$\mathbb{P}_S \left[|\hat{R}(h) - R(h)| \geq \epsilon \right] \leq \delta.$$

Bounds of this form are the heart of the *probably approximately correct* (PAC) learning framework. Their proofs rely on concentration bounds like the one in the following Lemma:

Lemma 1.1 (Hoeffding's inequality)**.** *Let $Z_1, \dots, Z_n$ be real independent random variables whose values are contained in intervals $[a_i, b_i] \supseteq \text{range}[Z_i]$. Then for every $\epsilon > 0$ it holds that*

$$\mathbb{P}\left[\sum_{i=1}^{n} Z_i - \mathbb{E}\left[Z_i\right] \geq \epsilon \right] \leq \exp\left[-\frac{2\epsilon^2}{\sum_{i=1}^{n}(a_i - b_i)^2} \right]. \tag{1.8}$$

Theorem 1.3: PAC bound for countable, weighted hypotheses

Consider a countable hypothesis class $\mathcal{F}$ and a loss function whose values are contained in an interval of length $c \geq 0$. Let p be any probability distribution over $\mathcal{F}$ and $\delta \in (0, 1]$ any confidence parameter. Then with probability at least $(1-\delta)$ w.r.t. repeated sampling of sets of training data

of size n we have

$$\forall h \in \mathcal{F} : \left|\hat{R}(h) - R(h)\right| \leq c\sqrt{\frac{\ln \frac{1}{p(h)} + \ln \frac{2}{\delta}}{2n}}. \tag{1.9}$$

Note: The bound is independent of the underlying probability measure P. If we consider classification with risk function equal to the error probability, then $c = 1$.

Proof. Let us first consider a fixed $h \in \mathcal{F}$ and apply Hoeffding's inequality to the i.i.d. random variables $Z_i := L(Y_i, h(X_i))/n$. Setting $\epsilon := c\sqrt{\left(\ln \frac{2}{p(h)\delta}\right)/2n}$ we obtain

$$\mathbb{P}_S\left[\left|\hat{R}(h) - R(h)\right| \geq \epsilon\right] \leq p(h)\delta. \tag{1.10}$$

In order to bound the probability that for any of the h's the empirical average deviates from the mean, we exploit the union bound and arrive at

$$\mathbb{P}_S\left[\exists h \in \mathcal{F} : \left|\hat{R}(h) - R(h)\right| \geq \epsilon\right] \leq \sum_{h\in\mathcal{F}} \mathbb{P}_S\left[\left|\hat{R}(h) - R(h)\right| \geq \epsilon\right] \leq \sum_{h\in\mathcal{F}} p(h)\delta = \delta.$$

□

The ϵ in Eq.(1.10) depends on the hypothesis h. The smaller the weight $p(h)$, the larger the corresponding ϵ. Hence, effectively, the above derivation provides reasonable bounds only for a finite number of hypotheses. If $\mathcal{F}$ itself is finite, we can choose $p(h) := 1/|\mathcal{F}|$ and rewrite the theorem so that it yields a bound for the size of the training set that is sufficient for PAC learnability:

Corollary 1.2. *Consider a finite hypothesis space $\mathcal{F}$, $\delta \in (0, 1]$, $\epsilon > 0$ and a loss function whose range is contained in an interval of length $c \geq 0$. Then $\forall h \in \mathcal{F} : |\hat{R}(h) - R(h)| \leq \epsilon$ holds with probability at least $1 - \delta$ over repeated sampling of training sets of size n, if*

$$n \;\geq\; \frac{c^2}{2\epsilon^2}\left(\ln|\mathcal{F}| + \ln\frac{2}{\delta}\right). \tag{1.11}$$

Due to Eq.(1.6) this also guarantees that $R(\hat{h}) - R_{\mathcal{F}} \leq 2\epsilon$, providing a quantitative justification of ERM. Moreover, in a deterministic scenario where a function $f : \mathcal{X} \to \mathcal{Y}$ determines the 'true' label, we have $R(\hat{h}) \leq 2\epsilon$, if $f \in \mathcal{F}$.

Unfortunately, for infinite $\mathcal{F}$ the statement of the corollary becomes void — a drawback that will to a large extent be corrected in the following sections.

If $\mathcal{F} = \mathcal{Y}^{\mathcal{X}}$ with all sets finite, then Eq.(1.11) provides a PAC guarantee essentially only if n exceeds $|\mathcal{X}|$. The latter means, however, that the learner has basically already seen all instances in the training data. The next theorem shows that this is indeed necessary for PAC learning if $\mathcal{F} = \mathcal{Y}^{\mathcal{X}}$, i.e., if the hypotheses class is not constrained.

1.4 No free lunch

If we are given part of a sequence, say 2 4 8 16, without further assumption about an underlying structure, we can not infer the next number. As Hume phrased it (first published anonymously in 1739): *there is nothing in any object, consider'd in itself, which can afford us a reason for drawing a conclusion beyond it.* The necessity of prior information in machine learning is put in a nutshell by the 'no-free-lunch theorem', of which one version is the following:

Theorem 1.4: No-free-lunch

Let $\mathcal{X}$ and $\mathcal{Y}$ both be finite and so that $|\mathcal{X}|$ exceeds the size n of the training set S. For any $f : \mathcal{X} \to \mathcal{Y}$ define $R_f(h) := \mathbb{P}\left[h(X) \neq f(X)\right]$ where the probability is taken w.r.t to a uniform distribution of X over $\mathcal{X}$. Then for every learner the expected risk averaged uniformly over all functions $f \in \mathcal{Y}^{\mathcal{X}}$ fulfills

$$\mathbb{E}_f\left[\mathbb{E}_S\left[R_f\big(h_S\big)\right]\right] \geq \left(1 - \frac{1}{|\mathcal{Y}|}\right)\left(1 - \frac{n}{|\mathcal{X}|}\right). \tag{1.12}$$

Note: Here it is understood that f determines the joint distribution $P(x, y) = \delta_{y,f(x)}/|\mathcal{X}|$. Consequently, the training data has the form $\big((x_i, f(x_i))\big)_{i=1}^n$.

Proof. Denote by $\mathcal{X}_S$ the subset of $\mathcal{X}$ appearing in the training data S. If necessary, add further elements to $\mathcal{X}_S$ until $|\mathcal{X}_S| = n$. We can write

$$\begin{aligned}
\mathbb{E}_f\left[\mathbb{E}_S\left[R_f\big(h_S\big)\right]\right] &= \frac{1}{|\mathcal{X}|}\mathbb{E}_f\left[\mathbb{E}_S\left[\sum_{x\in\mathcal{X}} \mathbb{1}_{h_S(x)\neq f(x)}\right]\right] && (1.13)\\
&\geq \frac{1}{|\mathcal{X}|}\mathbb{E}_f\left[\mathbb{E}_S\left[\sum_{x\notin\mathcal{X}_S} \mathbb{1}_{h_S(x)\neq f(x)}\right]\right]. && (1.14)
\end{aligned}$$

While inside $\mathcal{X}_S$ the value of $f(x)$ is determined by S, for $x \notin \mathcal{X}_S$ all $|\mathcal{Y}|$ values are possible and equally likely, so that $h_S(x) \neq f(x)$ holds with probability $1 - 1/|\mathcal{Y}|$ w.r.t. a uniform distribution over f's that are consistent with S. The remaining factor is due to $\sum_{x\notin\mathcal{X}_S} = |\mathcal{X}| - n$. □

Let us compare this with random guessing. The risk, i.e., the average error probability, of random guessing in the above scenario is $1 - 1/|\mathcal{Y}|$. Thm.1.4 only leaves little room for improvement beyond this—an additional factor $(1 - n/|\mathcal{X}|)$. This factor reflects the fact that the learner has already seen the training data, which is at most a fraction $n/|\mathcal{X}|$ of all cases. Regarding the unseen cases, however, all learners are the same on average and perform no better than random guessing. Note that the above proof also allows us to derive an upper bound in addition to the lower bound in Eq.(1.12). To this end, observe that the difference between Eqs.(1.13) and (1.14) is at most $n/|\mathcal{X}|$. Hence, in the limit

$n/|\mathcal{X}| \to 0$ the average error probability is exactly the one for random guessing, irrespective of what learner has been chosen.

This sobering result also implies that there is no order among learners. If one learner beats another on some functions, the converse has to hold on other functions. This result, as well as similar ones, has to be put into perspective, however, since not all functions are equally relevant.

The no-free-lunch theorem should not come as a surprise. In fact, it is little more than a formalization of a rather obvious claim within our framework: if one is given n values of a sequence of independently, identically and uniformly distributed random variables, then predicting the value of the $(n+1)$'st can not be better than random guessing. If prediction is to be better than chance, then additional structure is required. The inevitable a priori information about this structure can be incorporated into machine learning in different ways. In the approach we focus on in this course, the a priori information is reflected in the choice of the hypotheses class $\mathcal{F}$. In addition, hypotheses in $\mathcal{F}$ may effectively be given different weight, for instance resulting from SRM, regularization or a Bayesian prior distribution over $\mathcal{F}$. At the same time, this approach is *distribution-independent* in the sense that it makes no assumption about the distribution P that governs the data. An alternative approach (which we will not follow) would be to put prior information into P, for instance by assuming a parametric model for P.

1.5 Growth function

Starting in this section, we aim at generalizing the PAC bound derived in Sec.1.3 to beyond finite hypotheses classes. The first approach we will discuss essentially replaces the cardinality of $\mathcal{F}$ by the corresponding *growth function*.

Definition 1.3 (Growth function)**.** *Let $\mathcal{F} \subseteq \mathcal{Y}^{\mathcal{X}}$ be a class of functions with finite target space $\mathcal{Y}$. For every subset $\Xi \subseteq \mathcal{X}$ define the restriction of $\mathcal{F}$ to Ξ as $\mathcal{F}|_\Xi := \{f \in \mathcal{Y}^\Xi \mid \exists F \in \mathcal{F}\ \forall x \in \Xi : f(x) = F(x)\}$. The* growth function *$\Gamma : \mathbb{N} \to \mathbb{N}$ assigned to $\mathcal{F}$ is then defined as*

$$\Gamma(n) := \max_{\Xi \subseteq \mathcal{X} : |\Xi| = n} \big| \mathcal{F}|_\Xi \big|.$$

That is, the growth function describes the maximal size of $\mathcal{F}$ when restricted to a domain of n points.

Example 1.1 (Threshold functions). Consider the set of all threshold functions $\mathcal{F} \subseteq \{-1,1\}^{\mathbb{R}}$ defined by $\mathcal{F} := \{x \mapsto \mathrm{sgn}[x-b]\}_{b\in\mathbb{R}}$. Given a set of distinct points $\{x_1, \ldots, x_n\} = \Xi \subseteq \mathbb{R}$, there are $n+1$ functions in $\mathcal{F}|_\Xi$ corresponding to $n+1$ possible ways of placing b relative to the x_i's. Hence, in this case $\Gamma(n) = n+1$.

Theorem 1.5: PAC bound via growth function

Consider a hypothesis class $\mathcal{F}$ with finite target space $\mathcal{Y}$ and a loss func-

tion whose range is contained in an interval $[0, c]$. Let $\delta \in (0, 1]$. With probability at least $(1 - \delta)$ w.r.t. repeated sampling of training data of size n we have

$$\forall h \in \mathcal{F} : \ |R(h) - \hat{R}(h)| \leq c \sqrt{\frac{8 \ln \left(\Gamma(2n) \frac{4}{\delta}\right)}{n}}. \tag{1.15}$$

Note: this implies a non-trivial bound if the growth function grows sub-exponentially. This is true if $|\mathcal{F}| < \infty$, since $\forall n : \Gamma(n) \leq |\mathcal{F}|$, but it does not require a finite hypotheses class as already seen in example 1.1.

Proof. Let S and S' be i.i.d. random variables with values in $(\mathcal{X} \times \mathcal{Y})^n$ distributed according to some product probability measure P^n. For every value of S' denote by $\hat{R}'(h)$ the corresponding empirical risk of a hypothesis $h \in \mathcal{F}$. By virtue of the triangle inequality, if $|R(h) - \hat{R}(h)| > \epsilon$ and $|R(h) - \hat{R}'(h)| < \frac{\epsilon}{2}$, then $|\hat{R}'(h) - \hat{R}(h)| > \frac{\epsilon}{2}$. Expressed in terms of indicator functions this is

$$\mathbb{1}_{|R(h)-\hat{R}(h)|>\epsilon} \ \mathbb{1}_{|R(h)-\hat{R}'(h)|<\frac{\epsilon}{2}} \ \leq \ \mathbb{1}_{|\hat{R}'(h)-\hat{R}(h)|>\frac{\epsilon}{2}}. \tag{1.16}$$

Let us assume that $n \geq 4c^2\epsilon^{-2} \ln 2$, which will be justified later by a particular choice of ϵ. Taking the expectation value w.r.t. S' in Eq.(1.16) affects the second and third term. The former can be bounded using Hoeffding's inequality together with the assumption on n, which leads to

$$\mathbb{E}_{S'} \left[\mathbb{1}_{|R(h)-\hat{R}'(h)|<\frac{\epsilon}{2}} \right] \geq 1 - 2 \exp \left[-\frac{\epsilon^2 n}{2c^2} \right] \geq \frac{1}{2}.$$

For the expectation value of the last term in Eq.(1.16) we use

$$\mathbb{E}_{S'} \left[\mathbb{1}_{|\hat{R}'(h)-\hat{R}(h)|>\frac{\epsilon}{2}} \right] \leq \mathbb{P}_{S'} \left[\exists h \in \mathcal{F} : |\hat{R}'(h) - \hat{R}(h)| > \frac{\epsilon}{2} \right].$$

Inserting both bounds into Eq.(1.16) gives

$$\mathbb{1}_{|R(h)-\hat{R}(h)|>\epsilon} \ \leq 2 \ \mathbb{P}_{S'} \left[\exists h \in \mathcal{F} : |\hat{R}'(h) - \hat{R}(h)| > \frac{\epsilon}{2} \right].$$

As this holds for all $h \in \mathcal{F}$, we can replace the left hand side by $\mathbb{1}_{\exists h \in \mathcal{F} : |R(h)-\hat{R}(h)|>\epsilon}$. Taking the expectation w.r.t. S then leads to

$$\mathbb{P}_S \left[\exists h \in \mathcal{F} : |R(h) - \hat{R}(h)| > \epsilon \right] \leq 2 \mathbb{P}_{S,S'} \left[\exists h \in \mathcal{F} : |\hat{R}'(h) - \hat{R}(h)| > \frac{\epsilon}{2} \right].$$

Note that the r.h.s. involves only empirical quantities. This implies that every function h is only evaluated on at most $2n$ points, namely those appearing in S and S'. Since restricted to $2n$ points there are at most $\Gamma(2n)$ functions, our aim is now to exploit this together with the union bound and to bound the

remaining factor with Hoeffding's inequality. To this end, observe that we can write

$$2\mathbb{P}_{S,S'}\left[\exists h \in \mathcal{F} : |\hat{R}'(h) - \hat{R}(h)| > \frac{\epsilon}{2}\right] \; = \tag{1.17}$$
$$2\mathbb{E}_{SS'}\left[\mathbb{P}_\sigma\left[\exists h \in \mathcal{F} \; : \; \frac{1}{n}\Big|\sum_{i=1}^{n}\Big(L\big(h(X_i),Y_i\big) - L\big(h(X_i'),Y_i'\big)\Big)\sigma_i\Big| > \frac{\epsilon}{2}\right]\right],$$

where $\mathbb{P}_\sigma$ denotes the probability w.r.t. uniformly distributed $\sigma \in \{-1,1\}^n$. Eq.(1.17) is based on the fact that multiplication with $\sigma_i = -1$ amounts to interchanging $(X_i, Y_i) \leftrightarrow (X_i', Y_i')$, which has no effect since the random variables are independently and identically distributed. Now we can indeed use the union bound together with Hoeffding's inequality and arrive at

$$\begin{aligned}\mathbb{P}_S\left[\exists h \in \mathcal{F} : |R(h) - \hat{R}(h)| > \epsilon\right] &\leq 4\mathbb{E}_{SS'}\left[\big|\mathcal{F}|_{S\cup S'}\big|\right]\exp\left[-\frac{n\epsilon^2}{8c^2}\right] \\ &\leq 4\Gamma(2n)\exp\left[-\frac{n\epsilon^2}{8c^2}\right] \end{aligned} \tag{1.18}$$

The result then follows by setting the final expression in Eq.(1.18) equal to δ and solving for ϵ. The previously made assumption on n then becomes equivalent to $\delta \leq 2\sqrt{2}\Gamma(2n)$, which is always fulfilled as $\delta \in (0,1]$. □

Note that we have proven a slightly stronger result, in which the growth function $\Gamma(2n)$ is replaced by $\mathbb{E}_{SS'}\left[\big|\mathcal{F}|_{S\cup S'}\big|\right]$. The logarithm of this expectation value is called *VC-entropy*. The VC-entropy, however, depends on the underlying probability distribution P and is thus difficult to estimate in general.

For later use, let us discuss the behavior of the growth function w.r.t. compositions:

Lemma 1.4 (Growth functions under compositions)**.** *Consider function classes $\mathcal{F}_1 \subseteq \mathcal{Y}^\mathcal{X}, \mathcal{F}_2 \subseteq \mathcal{Z}^\mathcal{Y}$ and $\mathcal{F} := \mathcal{F}_2 \circ \mathcal{F}_1$. The respective growth functions then satisfy*

$$\Gamma(n) \leq \Gamma_1(n)\Gamma_2(n).$$

Proof. Fix an arbitrary subset $\Xi \subseteq \mathcal{X}$ of cardinality $|\Xi| = n$. With $\mathcal{G} := \mathcal{F}_1|_\Xi$ we can write $\mathcal{F}|_\Xi = \bigcup_{g\in\mathcal{G}}\{f \circ g \mid f \in \mathcal{F}_2\}$. So

$$\begin{aligned}\big|\mathcal{F}|_\Xi\big| &\leq \big|\mathcal{F}_1|_\Xi\big| \max_{g\in\mathcal{G}}\big|\{f\circ g | f \in \mathcal{F}_2\}\big| \\ &\leq \Gamma_1(n)\max_{g\in\mathcal{G}}\big|\mathcal{F}_2|_{g(\Xi)}\big| \\ &\leq \Gamma_1(n)\Gamma_2(n).\end{aligned}$$

□

1.6 VC-dimension

In the case of binary target space ($|\mathcal{Y}| = 2$) there is a peculiar dichotomy in the behavior of the growth function $\Gamma(n)$. It grows at maximal rate, i.e., exponentially and exactly like 2^n, up to some $n = d$ and from then on remains bounded by a polynomial of degree at most d. The number d where this transition occurs, is called the *VC-dimension* of the function class and plays an important role in the theory of binary classification.

Definition 1.5 (Vapnik-Chervonenkis dimension)**.** *The* VC-dimension *of a function class $\mathcal{F} \subseteq \mathcal{Y}^{\mathcal{X}}$ with binary target space $\mathcal{Y}$ is defined as*

$$\mathrm{VCdim}(\mathcal{F}) := \max\left\{n \in \mathbb{N} \;\middle|\; \Gamma(n) = 2^n\right\}$$

if the maximum exists and $\mathrm{VCdim}(\mathcal{F}) = \infty$ *otherwise.*

That is, if $\mathrm{VCdim}(\mathcal{F}) = d$, then there exists a set $A \subseteq \mathcal{X}$ of d points, such that $\mathcal{F}|_A = \mathcal{Y}^A$ and the VC-dimension is the largest such number.

Example 1.2 (Threshold functions). If $\mathcal{F} = \{\mathbb{R} \ni x \mapsto \mathrm{sgn}[x - b]\}_{b\in\mathbb{R}}$, then $\mathrm{VCdim}(\mathcal{F}) = 1$ as we have seen in example 1.1 that $\Gamma(n) = n + 1$. More specifically, if we consider an arbitrary pair of points $x_1 < x_2$, then the assignment $x_1 \mapsto 1 \wedge x_2 \mapsto -1$ is missing in $\mathcal{F}|_{\{x_1,x_2\}}$. Hence, $\mathrm{VCdim}(\mathcal{F}) < 2$.

Theorem 1.6: VC-dichotomy of growth function

Consider a function class $\mathcal{F} \subseteq \mathcal{Y}^{\mathcal{X}}$ with binary target space $\mathcal{Y}$ and VC-dimension d. Then the corresponding growth function satisfies

$$\Gamma(n) \begin{cases} = 2^n, & \text{if } n \leq d. \\ \leq \left(\frac{en}{d}\right)^d, & \text{if } n > d. \end{cases} \tag{1.19}$$

Proof. $\Gamma(n) = 2^n$ for all $n \leq d$ holds by definition of the VC-dimension. In order to arrive at the expression for $n > d$, we show that for every subset $A \subseteq \mathcal{X}$ with $|A| = n$ the following is true:

$$\left|\mathcal{F}|_A\right| \leq \left|\left\{B \subseteq A \;\middle|\; \mathcal{F}|_B = \mathcal{Y}^B\right\}\right|. \tag{1.20}$$

If Eq.(1.20) holds, we can upper bound the r.h.s. by $\left|\{B \subseteq A \middle|\ |B| \leq d\}\right| = \sum_{i=0}^{d} \binom{n}{i}$, which for $n > d$ in turn can be bounded by

$$\begin{aligned} \sum_{i=0}^{d} \binom{n}{i} &\leq \sum_{i=0}^{n} \binom{n}{i} \left(\frac{n}{d}\right)^{d-i} \\ &= \left(\frac{n}{d}\right)^d \left(1 + \frac{d}{n}\right)^n \leq \left(\frac{n}{d}\right)^d e^d, \end{aligned}$$

where the last step follows from $\forall x \in \mathbb{R} : (1 + x) \leq e^x$. Hence, the proof is reduced to showing Eq.(1.20).

This will be done by induction on $|A|$. For $|A| = 1$ it is true (as $B = \emptyset$ always counts). Now assume as induction hypothesis that it holds for all sets of size $n - 1$ and that $|A| = n$. Let a be any element of A and define

$$\mathcal{F}' := \left\{h \in \mathcal{F}|_A \;\middle|\; \exists g \in \mathcal{F}|_A : h(a) \neq g(a) \wedge (h - g)|_{A\setminus a} = 0\right\}, \quad \mathcal{F}_a := \mathcal{F}'|_{A\setminus a}.$$

Then $|\mathcal{F}|_A| = |\mathcal{F}|_{A\setminus a}| + |\mathcal{F}_a|$ and both terms on the r.h.s. can be bounded by the induction hypothesis. For the first term we obtain

$$\left|\mathcal{F}|_{A\setminus a}\right| \leq \left|\left\{B \subseteq A \;\middle|\; \mathcal{F}|_B = \mathcal{Y}^B \wedge a \notin B\right\}\right|. \tag{1.21}$$

The second term can be bounded by

$$\begin{aligned} \left|\mathcal{F}_a\right| &= \left|\mathcal{F}'|_{A\setminus a}\right| \leq \left|\left\{B \subseteq A \setminus a \;\middle|\; \mathcal{F}'|_B = \mathcal{Y}^B\right\}\right| \\ &= \left|\left\{B \subseteq A \setminus a \;\middle|\; \mathcal{F}'|_{B\cup a} = \mathcal{Y}^{B\cup a}\right\}\right| \\ &= \left|\left\{B \subseteq A \;\middle|\; \mathcal{F}'|_B = \mathcal{Y}^B \wedge a \in B\right\}\right| \\ &\leq \left|\left\{B \subseteq A \;\middle|\; \mathcal{F}|_B = \mathcal{Y}^B \wedge a \in B\right\}\right|, \end{aligned} \tag{1.22}$$

where we use the induction hypothesis in the first line and the step to the second line uses the defining property of $\mathcal{F}'$. Adding the bounds of Eq.(1.21) and Eq.(1.22) then yields the result claimed in Eq.(1.20). □

Now we can plug this bound on the growth function into the PAC bound in Thm.1.5. After a couple of elementary manipulations we then arrive at the following result, which, similar to Cor.1.2, provides a bound on the necessary statistics, but with the VC-dimension d now playing the role of $\ln|\mathcal{F}|$.

Corollary 1.6. *Consider a function class $\mathcal{F} \subseteq \mathcal{Y}^{\mathcal{X}}$ with binary target space and VC-dimension d. Let $(\epsilon, \delta) \in (0, 1]^2$ and choose the risk function R to be the error probability. Then $\forall h \in \mathcal{F} : |\hat{R}(h) - R(h)| \leq \epsilon$ holds with probability at least $1 - \delta$ over repeated sampling of training sets of size n, if*

$$n \;\geq\; \frac{32}{\epsilon^2}\left[d \ln\left(\frac{8d}{\epsilon^2}\right) + \ln\frac{6}{\delta}\right]. \tag{1.23}$$

Note: the bound in Eq.(1.23) can be slightly improved. In particular, the first logarithm turns out to be unnecessary, cf. Eq.(1.28).

A useful tool for computing VC-dimensions is the following theorem:

Theorem 1.7: VC-dimension for function vector spaces

Let $\mathcal{G}$ be a real vector space of functions from $\mathcal{X}$ to $\mathbb{R}$. Then $\mathcal{F} := \big\{x \mapsto \operatorname{sgn}[g(x)]\big\}_{g\in\mathcal{G}} \subseteq \{-1,1\}^{\mathcal{X}}$ has $\operatorname{VCdim}(\mathcal{F}) = \dim(\mathcal{G})$.

Proof. Let us first prove $\operatorname{VCdim}(\mathcal{F}) \leq \dim(\mathcal{G})$. We can assume $\dim(\mathcal{G}) < \infty$ and argue by contradiction. Let $k = \dim(\mathcal{G})+1$ and suppose that $\operatorname{VCdim}(\mathcal{F}) \geq k$. Then there is a subset $\Xi = \{x_1,\ldots,x_k\} \subseteq \mathcal{X}$ such that $\mathcal{F}|_\Xi = \{-1,1\}^\Xi$. Define a map $L : \mathcal{G} \to \mathbb{R}^k$ via $L(g) := \big(g(x_1),\ldots,g(x_k)\big)$. L is a linear map whose range has dimension at most $\dim(\mathcal{G})$. Hence, there is a non-zero vector $v \in (\text{range } L)^\perp = \ker \mathrm{L}^*$. This means that for all $g \in \mathcal{G}$:

$$0 = \langle L^*(v), g\rangle = \langle v, L(g)\rangle = \sum_{l=1}^{k} v_l\, g(x_l). \tag{1.24}$$

However, if $\mathcal{F}|_\Xi = \{-1,1\}^\Xi$, we can choose g such that $\operatorname{sgn}[g(x_l)] = \operatorname{sgn}[v_l]$ for all $l \in \{1,\ldots,k\}$, which would imply $\sum_{l=1}^k v_l\, g(x_l) > 0$.

In order to arrive at $\operatorname{VCdim}(\mathcal{F}) \geq \dim(\mathcal{G})$, it suffices to show that for all $d \leq \dim(\mathcal{G})$ there are points $x_1,\ldots,x_d \in \mathcal{X}$ such that for all $y \in \mathbb{R}^d$ there is a $g \in \mathcal{G}$ satisfying $y_j = g(x_j)$ for all j. To this end, consider d linearly independent functions $(g_i)_{i=1}^d$ in $\mathcal{G}$ and define $G(x) := \big(g_1(x),\ldots,g_d(x)\big)$. Then $\operatorname{span}\{G(x)\}_{x\in\mathcal{X}} = \mathbb{R}^d$ so that there have to exist d linearly independent vectors $G(x_1),\ldots,G(x_d)$. Hence, the $d \times d$ matrix with entries $g_i(x_j)$ is invertible and for all $y \in \mathbb{R}^d$ the system of equations $y_j = \sum_{i=1}^d \gamma_i g_i(x_j)$ has a solution $\gamma \in \mathbb{R}^d$. □

Corollary 1.7 (VC-dimension of half spaces)**.** *The set* $\mathcal{F} := \big\{h : \mathbb{R}^d \to \{-1,1\} \mid \exists (v,b) \in \mathbb{R}^d \times \mathbb{R} :\ h(x) = \operatorname{sgn}[\langle v,x\rangle - b]\big\}$*, which corresponds to the set of all half spaces in* $\mathbb{R}^d$*, satisfies*

$$\operatorname{VCdim}(\mathcal{F}) = d+1.$$

Proof. The result follows from the foregoing theorem, when applied to the linear space of functions spanned by $g_i(x) := x_i$ for $i = 1,\ldots,d$ and $g_{d+1}(x) := 1$. □

As in the case of half spaces, we can assign a function $f : \mathbb{R}^d \to \{-1,1\}$ to any subset $C \subseteq \mathbb{R}^d$ and vice versa via $f(x) = 1 \Leftrightarrow x \in C$. In this way we can apply the notion of VC-dimension to classes of Borel subsets of $\mathbb{R}$. Table 1.1 collects some examples.

Example 1.3 (Axes-aligned rectangles). Consider $\mathcal{C} := \{C \subseteq \mathbb{R}^d | \exists a,b \in \mathbb{R}^d :\ C = [a_1,b_1] \times \ldots \times [a_d,b_d]\}$ the set of all axes-aligned rectangles in $\mathbb{R}^d$ and let $\mathcal{F} := \{f : \mathbb{R}^d \to \{0,1\} | \exists C \in \mathcal{C} : f(x) = \mathbb{1}_{x\in\mathcal{C}}\}$ be the corresponding class of indicator-functions. For any set of points $A = \{x_1,\ldots,x_n\} \subseteq \mathbb{R}^d$ there is a unique smallest rectangle $C_{min} \in \mathcal{C}$ so that $A \subseteq C_{min}$. As long as $n > 2d$

	$\mathcal{X}$	VCdim	see
l_2-balls	$\mathbb{R}^d$	$d+1$	[9]
l_∞-balls	$\mathbb{R}^d$	$\lfloor (3d+1)/2 \rfloor$	[8]
half spaces	$\mathbb{R}^d$	$d+1$	Cor.1.7
axes-aligned rectangles	$\mathbb{R}^d$	$2d$	Exp.1.3
convex k-gons	$\mathbb{R}^2$	$2k+1$	[5]
semialgebraic sets $\mathcal{S}_{k,m}$	$\mathbb{R}^d$	$\leq 2k\binom{m+d}{m} \ln\big((k^2+k)\binom{m+d}{m}\big)$	[4]
$\mathcal{S}_{1,m}$	$\mathbb{R}^d$	$\binom{m+d}{m}$	[4]
$\mathrm{Aff}(C)$ for fixed $C \in \mathcal{S}_{k,m}$	$\mathbb{R}^d$	$\mathcal{O}\big(d^2 \ln(dkm)\big)$	[4]
$\{x \mapsto \operatorname{sgn}\sin[\alpha x] \mid \alpha \in \mathbb{R}\}$	$\mathbb{R}$	∞	Exp.1.4

Figure 1.1: VC-dimension of various classes of functions or corresponding geometric objects. A convex k-gon means a polygon in $\mathbb{R}^2$ that is obtained by intersecting k half spaces. $\mathcal{S}_{k,m}$ is the class of subsets of $\mathbb{R}^d$ that can be obtained as Boolean combination of k sets of the form $f_j^{-1}\big((0,\infty)\big)$ where each $f_j : \mathbb{R}^d \mapsto \mathbb{R}$, $j = 1,\ldots,k$ is a polynomial of maximal degree m. $\mathrm{Aff}(C)$ denotes the class of all affine transformations of C.

we can discard points from A without changing C_{min}. Let $\tilde{A} \subseteq A$ be such a reduced set with $|\tilde{A}| \leq 2d$. Then every $f \in \mathcal{F}|_A$ that assigns a value 1 to all elements of $\tilde{A}$ also assigns 1 to all $A \setminus \tilde{A}$, since those lie inside the same box. Hence, if $n > 2d$, then the function $\tilde{f}(x) := \mathbb{1}_{x \in \tilde{A}}$ is not contained in $\mathcal{F}|_A$ and therefore $\mathrm{VCdim}(\mathcal{F}) \leq 2d$.

To prove equality, consider the extreme points of the d-dimensional hyper-octahedron (i.e., the l_1-unit ball), which are given by all the permutations of $(\pm 1, 0, \ldots, 0)$. Denote them by $x_k^{(+)}$ and $x_k^{(-)}$, $k = 1,\ldots,d$, depending on whether the k'th component is $+1$ or -1. Let f be an arbitrary assignment of values 0 or 1 to these $2d$ points. Then

$$b_k := \frac{1}{2} + f\big(x_k^{(+)}\big) \quad \text{and} \quad a_k := -\frac{1}{2} - f\big(x_k^{(-)}\big)$$

define a rectangle $C \in \mathcal{C}$, which is such that $x_k^{(\pm)} \in C \Leftrightarrow f(x_k^{(\pm)}) = 1$. So, restricted to these $2d$ points, $\mathcal{F}$ still contains all functions and thus $\mathrm{VCdim}(\mathcal{F}) \geq 2d$.

In the examples discussed so far, the VC-dimension was essentially equal to the number of parameters that appear in the definition of the considered hypotheses class. That such a relation is not generally true is shown by the following example:

Example 1.4 (Sine-functions). Consider $\mathcal{F} := \{x \mapsto \operatorname{sgn}\sin(x\alpha) \mid \alpha \in \mathbb{R}\}$ and $A := \{2^{-k} \mid k = 1,\ldots,n\}$. Let f be an arbitrary assignment of values ± 1 to the

points $x_k := 2^{-k}$ in A. If we choose

$$\begin{aligned} \alpha &:= \pi\left(1+\sum_{k=1}^{n}\frac{1-f(x_k)}{2}\,2^k\right), \quad \text{we obtain} \\ \alpha\, x_l \bmod 2\pi &= \pi\left(\frac{1-f(x_l)}{2}\right)+\pi\left[2^{-l}+\sum_{k=1}^{l-1}2^{k-l}\left(\frac{1-f(x_k)}{2}\right)\right] \\ &=: \pi\left(\frac{1-f(x_l)}{2}\right)+\pi\, c, \end{aligned} \tag{1.25}$$

where $c \in (0,1)$. Consequently, $\operatorname{sgn}\sin(\alpha x_l) = f(x_l)$ and thus $\mathcal{F}|_A = \{-1,1\}^A$. Since this holds for all n, we have $\mathrm{VCdim}(\mathcal{F}) = \infty$ despite the fact that there is only a single real parameter involved.

Although the VC-dimension is infinite in this example, there are finite sets B for which $\mathcal{F}|_B \neq \{-1,1\}^B$. Consider for instance $B := \{1,2,3,4\}$ and the assignment $f(1) = f(2) = -f(3) = f(4) = -1$. If $\alpha = 2\pi m - \delta$, $m \in \mathbb{N}$ with $\delta \in [0, 2\pi)$ is to reproduce the first three values, then $\delta \in [\pi/3, \pi/2)$. However, this implies that 4δ is in the range where the sine is positive, so that $f(4) = -1$ cannot be matched.

1.7 Fundamental theorem of binary classification

In this section we collect the insights obtained so far and use them to prove what may be called the *fundamental theorem of binary classification*. For its formulation, denote by $\mathrm{poly}\left(\frac{1}{\epsilon},\frac{1}{\delta}\right)$ the set of all function of the form $(0,1]\times(0,1] \ni (\epsilon,\delta) \mapsto \nu(\epsilon,\delta) \in \mathbb{R}_+$ that are polynomial in $\frac{1}{\epsilon}$ and $\frac{1}{\delta}$.

Theorem 1.8: Fundamental theorem of binary classification

Let $\mathcal{F} \subseteq \{-1,1\}^{\mathcal{X}}$ be any hypotheses class and $n = |S|$ the size of the training data set S, which is treated as a random variable, distributed according to some product probability measure P^n. Choose the risk function R to be the error probability. Then the following are equivalent:

1. **(Finite VC-dimension)** $\mathrm{VCdim}(\mathcal{F}) < \infty$.
2. **(Uniform convergence)** There is a $\nu \in \mathrm{poly}\left(\frac{1}{\epsilon},\frac{1}{\delta}\right)$ so that for all $(\epsilon,\delta) \in (0,1]^2$ and all probability measures P we have

$$n \geq \nu(\epsilon,\delta) \quad \Rightarrow \quad \forall h \in \mathcal{F}: \quad \mathbb{P}_S\left[|\hat{R}(h) - R(h)| \geq \epsilon\right] \leq \delta.$$

3. **(PAC learnability)** There is a $\nu \in \mathrm{poly}\left(\frac{1}{\epsilon},\frac{1}{\delta}\right)$ and a learner that maps $S \mapsto h_S \in \mathcal{F}$ so that for all $(\epsilon,\delta) \in (0,1]^2$ and all probability

measures P we have

$$n \geq \nu(\epsilon, \delta) \quad \Rightarrow \quad \mathbb{P}_S\left[|R(h_S) - R_{\mathcal{F}}| \geq \epsilon\right] \leq \delta. \tag{1.26}$$

4. **(PAC learnability via ERM)** There is a $\nu \in \mathrm{poly}\left(\frac{1}{\epsilon}, \frac{1}{\delta}\right)$ so that for all $(\epsilon, \delta) \in (0,1]^2$ and all probability measures P we have

$$n \geq \nu(\epsilon, \delta) \quad \Rightarrow \quad \mathbb{P}_S\left[|R(\hat{h}) - R_{\mathcal{F}}| \geq \epsilon\right] \leq \delta,$$

where $\hat{h} \in \mathcal{F}$ is an arbitrary empirical risk minimizer.

Proof. 1. $\Rightarrow$ 2. is the content of Cor.1.6.

2. $\Rightarrow$ 4.: Assuming uniform convergence, with probability at least $(1-\delta)$ we have that $\forall h \in \mathcal{F}: \ |\hat{R}(h) - R(h)| \leq \frac{\epsilon}{2}$ if $n \geq \nu(\frac{\epsilon}{2}, \delta)$. By Eq.(1.6) this implies $R(\hat{h}) - R_{\mathcal{F}} \leq \epsilon$.

4. $\Rightarrow$ 3. is obvious since the former is a particular instance of the latter.

3. $\Rightarrow$ 1. is proven by contradiction: choose $\epsilon = \delta = 1/4$, $n = \nu(\epsilon, \delta)$ and suppose $\mathrm{VCdim}(\mathcal{F}) = \infty$. Then for any $N \in \mathbb{N}$ there is a subset $\Xi \in \mathcal{X}$ of size $|\Xi| = N$ such that $\mathcal{F}|_\Xi = \{-1,1\}^\Xi$. Applying the no-free-lunch theorem to this space we get that there is an $f: \Xi \to \{-1,1\}$, which defines a probability density $P(x,y) := \mathbb{1}_{x \in \Xi \,\wedge\, f(x)=y}/N$ on $\mathcal{X} \times \{-1,1\}$ with respect to which

$$\mathbb{E}_S\left[R(h_S)\right] \geq \frac{1}{2}\left(1 - \frac{n}{N}\right) \tag{1.27}$$

holds for an arbitrary learner, given by a mapping $S \mapsto h_S$. Using that $R(h_S)$ is itself a probability and thus bounded by one, we can bound

$$\mathbb{E}_S\left[R(h_S)\right] \leq 1 \cdot \mathbb{P}_S\left[R(h_S) \geq \epsilon\right] + \epsilon\big(1 - \mathbb{P}_S\left[R(h_S) \geq \epsilon\right]\big).$$

Together with Eq.(1.27) and $\epsilon = \frac{1}{4}$ this leads to $\mathbb{P}_S\left[R(h_S) \geq \frac{1}{4}\right] \geq \frac{1}{3} - \frac{4n}{6N}$, which for sufficiently large N contradict $\delta = \frac{1}{4}$. □

There is also a quantitative version of this theorem. In fact, the VC-dimension does not only lead to a bound on the necessary statistics, it precisely specifies the optimal scaling of ν. Let us denote by $\nu_{\mathcal{F}}$ the pointwise infimum of all functions ν taken i) over all functions for which the implication in Eq.(1.26) is true for all P and all (ϵ, δ) and ii) over all learners with range $\mathcal{F}$. $\nu_{\mathcal{F}}$ is called the *sample complexity* of $\mathcal{F}$ and it can be shown that

$$\nu_{\mathcal{F}}(\epsilon, \delta) = \Theta\left(\frac{\mathrm{VCdim}(\mathcal{F}) + \ln\frac{1}{\delta}}{\epsilon^2}\right). \tag{1.28}$$

Here, the asymptotic notation symbol Θ means that there are asymptotic upper and lower bounds that differ only by multiplicative constants (that are non-zero and finite).

Note that the scaling in $1/\delta$ is much better than required—logarithmic rather than polynomial. Hence, we could have formulated a stronger version of the

fundamental theorem. However, requiring polynomial scaling is what is typically done in the general definition of PAC learnability.

What about generalizations to cases with $|\mathcal{Y}| > 2$? For both, classification ($\mathcal{Y}$ discrete) and regression ($\mathcal{Y}$ continuous), the concept of VC-dimension has been generalized and there exist various counterparts to the VC-dimension with similar implications. For the case of classification, the *graph dimension* d_G and the *Natarajan dimension* d_N are two useful generalizations that lead to quantitative bounds on the sample complexity of a hypotheses class with the error probability as risk function. In the binary case they both coincide with the VC-dimension, while in general $d_N \leq d_G \leq 4.67 d_N \log_2 |\mathcal{Y}|$ (cf. [3]). Known bounds on the sample complexity $\nu_{\mathcal{F}}$ turn out to have still the form of Eq.(1.28) with the only difference that in the upper and lower bound the role of the VC-dimension is played by d_G and d_N, respectively. The logarithmic gap between the two appears to be relevant and leads to the possibility of good and bad ERM learners (cf. [7]).

In the case of regression, a well-studied counterpart of the VC-dimension is the *fat-shattering dimension*. For particular loss functions (e.g., the squared loss) the above theorem then has a direct analogue, in the sense that under mild assumptions, uniform convergence, finite fat-shattering dimension and PAC learnability are equivalent [2]. In general learning contexts, however, uniform convergence turns out to be a strictly stronger requirement than PAC learnability [12, 7].

1.8 Rademacher complexity

The approaches discussed so far were distribution independent. Growth function and VC-dimension, as well as its various generalizations, depend only on the hypotheses class $\mathcal{F}$ and lead to PAC guarantees that are independent of the probability measures P. In this section we will consider an alternative approach and introduce the *Rademacher complexities*, which depend on both, $\mathcal{F}$ and P.

Definition 1.8 (Rademacher complexity)**.** *Consider a set of real-valued functions $\mathcal{G} \subseteq \mathbb{R}^{\mathcal{Z}}$ and a vector $z \in \mathcal{Z}^n$. The* empirical Rademacher complexity *of $\mathcal{G}$ w.r.t. z is defined as*

$$\hat{\mathcal{R}}(\mathcal{G}) := \mathbb{E}_\sigma \left[\sup_{g \in \mathcal{G}} \frac{1}{n} \sum_{i=1}^{n} \sigma_i g(z_i) \right], \tag{1.29}$$

where $\mathbb{E}_\sigma$ denotes the expectation w.r.t. a uniform distribution of $\sigma \in \{-1, 1\}^n$. If the z_i's are considered values of a vector of i.i.d.random variables $Z := (Z_1, \ldots, Z_n)$, each distributed according to a probability measure P on $\mathcal{Z}$, then the Rademacher complexities *of $\mathcal{G}$ w.r.t. P are given by*

$$\mathcal{R}_n(\mathcal{G}) := \mathbb{E}_Z \left[\hat{\mathcal{R}}(\mathcal{G}) \right]. \tag{1.30}$$

Note: The uniformly distributed σ_i's are called *Rademacher variables*. Whenever we want to emphasize the dependence of $\hat{\mathcal{R}}(\mathcal{G})$ on $z \in \mathcal{Z}^n$, we will write $\hat{\mathcal{R}}_z(\mathcal{G})$. Similarly, we occasionally write $\mathcal{R}_{n,P}(\mathcal{G})$ to make the dependence on P explicit. We will tacitly assume that $\mathcal{G}$ is chosen so that the suprema appearing in the definition lead to measurable functions.

The richer the function class $\mathcal{G}$, the larger the (empirical) Rademacher complexity. If we define $g(z) := \big(g(z_1), \ldots, g(z_n)\big)$ and write

$$\hat{\mathcal{R}}(\mathcal{G}) = \frac{1}{n}\mathbb{E}_\sigma \left[\sup_{g\in\mathcal{G}} \langle \sigma, g(z)\rangle\right],$$

we see that the (empirical) Rademacher complexity measures how well the function class $\mathcal{G}$ can 'match Rademacher noise'. If for a random sign pattern σ there is always a function in $\mathcal{G}$ that is well aligned with σ in the sense that $\langle \sigma, g(z)\rangle$ is large, the Rademacher complexity will be large. Clearly, this will become more and more difficult when the number n of considered points is increased.

With the following Lemma, which is a refinement of Hoeffding's inequality, we can show that the Rademacher complexity is close to its empirical counterpart. This will imply that the Rademacher complexity can be estimated reliably from the data and that no additional knowledge about P is required.

Lemma 1.9 (McDiarmid's inequality)**.** *Let $(Z_1, \ldots, Z_n) = Z$ be a finite sequence of independent random variables, each with values in $\mathcal{Z}$ and $\varphi : \mathcal{Z}^n \to \mathbb{R}$ a measurable function such that $|\varphi(z) - \varphi(z')| \leq \nu_i$ whenever z and z' only differ in the i'th coordinate. Then for every $\epsilon > 0$*

$$\mathbb{P}\big[\varphi(Z) - \mathbb{E}\left[\varphi(Z)\right] \geq \epsilon\big] \leq \exp\left[-\frac{2\epsilon^2}{\sum_{i=1}^n \nu_i^2}\right]. \tag{1.31}$$

Note: the same inequality holds with $\varphi(Z)$ and $\mathbb{E}\left[\varphi(Z)\right]$ interchanged. This can be seen by replacing φ with $-\varphi$.

Lemma 1.10 (Rademacher vs. empirical Rademacher complexity)**.** *Let $\mathcal{G} \subseteq [a,b]^{\mathcal{Z}}$ be a set of real-valued functions. Then for every $\epsilon > 0$ and any product probability measure P^n on $\mathcal{Z}^n$ it holds that*

$$\mathbb{P}_Z\left[\big(\mathcal{R}_n(\mathcal{G}) - \hat{\mathcal{R}}_Z(\mathcal{G})\big) \geq \epsilon\right] \leq \exp -\frac{2n\epsilon^2}{(b-a)^2}. \tag{1.32}$$

Proof. Define $\varphi : \mathcal{Z}^n \to \mathbb{R}$ as $\varphi(z) := \hat{\mathcal{R}}_z(\mathcal{G})$, which implies $\mathbb{E}\left[\varphi(Z)\right] = \mathcal{R}_n(\mathcal{G})$. Let $z, z' \in \mathcal{Z}^n$ be a pair that differs in only one component. Then $\sup_{g\in\mathcal{G}} \sum_i \sigma_i g(z_i)$ changes by at most $|b-a|$ if we replace z by z'. Consequently,

$$|\varphi(z) - \varphi(z')| = |\hat{\mathcal{R}}_z(\mathcal{G}) - \hat{\mathcal{R}}_{z'}(\mathcal{G})| \leq \frac{|b-a|}{n}, \tag{1.33}$$

and we can apply McDiarmid's inequality to obtain the stated result. □

Now we are prepared for the main result of this section and can prove a PAC guarantee based on (empirical) Rademacher complexities:

Theorem 1.9: PAC bound via Rademacher complexities

Consider a hypotheses class $\mathcal{F} \subseteq \mathcal{Y}^{\mathcal{X}}$ with target space $\mathcal{Y} \subseteq \mathbb{R}$, a loss function $L : \mathcal{Y} \times \mathcal{Y} \mapsto [0, c]$ and define $\mathcal{Z} := \mathcal{X} \times \mathcal{Y}$ and $\mathcal{G} := \{(x, y) \mapsto L(y, h(x)) \mid h \in \mathcal{F}\} \subseteq [0, c]^{\mathcal{Z}}$. For any $\delta > 0$, any probability measure P on $\mathcal{Z}$ and any $h \in \mathcal{F}$ we have with probability at least $(1 - \delta)$ w.r.t. repeated sampling of P^n-distributed training data $S \in \mathcal{Z}^n$:

$$R(h) - \hat{R}(h) \quad \leq \quad 2\mathcal{R}_n(\mathcal{G}) + c\sqrt{\frac{\ln \frac{1}{\delta}}{2n}}, \quad \text{and} \tag{1.34}$$

$$R(h) - \hat{R}(h) \quad \leq \quad 2\hat{\mathcal{R}}_S(\mathcal{G}) + 3c\sqrt{\frac{\ln \frac{2}{\delta}}{2n}}. \tag{1.35}$$

Proof. Defining $\varphi : \mathcal{Z}^n \to \mathbb{R}$ as $\varphi(S) := \sup_{h \in \mathcal{F}} \big(R(h) - \hat{R}(h)\big)$, we can apply McDiarmid's inequality to φ with $\nu_i = \frac{c}{n}$ and obtain

$$\mathbb{P}_S \left[\varphi(S) - \mathbb{E}_S \left[\varphi(S)\right] \geq \epsilon\right] \leq e^{-2n\epsilon^2/c^2}.$$

Setting the r.h.s. equal to δ and solving for ϵ then gives that with probability at least $1 - \delta$ we have

$$\sup_{h \in \mathcal{F}} \big(R(h) - \hat{R}(h)\big) \leq \mathbb{E}_S \left[\varphi(S)\right] + c\sqrt{\frac{\ln \frac{1}{\delta}}{2n}}. \tag{1.36}$$

It remains to upper bound the expectation on the right. To this end, we will again introduce a second sample S' that is an i.i.d. copy of S. Then

$$\begin{aligned}
\mathbb{E}_S \left[\varphi(S)\right] &= \mathbb{E}_S \left[\sup_{h \in \mathcal{F}} \frac{1}{n} \sum_{i=1}^{n} \mathbb{E}_{S'} \left[L\big(Y_i', h(X_i')\big) - L\big(Y_i, h(X_i)\big)\right]\right] \\
&\leq \mathbb{E}_{SS'} \left[\sup_{h \in \mathcal{F}} \frac{1}{n} \sum_{i=1}^{n} L\big(Y_i', h(X_i')\big) - L\big(Y_i, h(X_i)\big)\right] \\
&= \mathbb{E}_{SS'} \mathbb{E}_\sigma \left[\sup_{h \in \mathcal{F}} \frac{1}{n} \sum_{i=1}^{n} \sigma_i \Big(L\big(Y_i', h(X_i')\big) - L\big(Y_i, h(X_i)\big)\Big)\right] \\
&\leq 2\, \mathbb{E}_S \mathbb{E}_\sigma \left[\sup_{h \in \mathcal{F}} \frac{1}{n} \sum_{i=1}^{n} \sigma_i L\big(Y_i, h(X_i)\big)\right] \;=\; 2\mathcal{R}_n(\mathcal{G}),
\end{aligned}$$

where between the second and third line we have used that multiplication with $\sigma_i = -1$ amounts to interchanging $(X_i, Y_i) \leftrightarrow (X_i', Y_i')$, which has no effect as these are i.i.d. random variables. This proves Eq.(1.34). In order to obtain Eq.(1.35) note that by Lemma 1.10 with probability at least $1 - \delta/2$ we have

$$\mathcal{R}_n(\mathcal{G}) \leq \hat{\mathcal{R}}(\mathcal{G}) + c\sqrt{\frac{\ln \frac{2}{\delta}}{2n}}.$$

Combining this via the union bound with Eq.(1.34), where the latter is also applied to $\delta/2$ instead of δ, then yields the desired result. □

When applying the previous theorem to the case of binary classification, one can replace the Rademacher complexities of $\mathcal{G}$ by those of the hypotheses class $\mathcal{F}$:

Lemma 1.11 (Rademacher complexities for binary classification)**.** *Consider a hypotheses class $\mathcal{F} \subseteq \{-1,1\}^{\mathcal{X}}$, $L(y,y') := \mathbb{1}_{y\neq y'}$ as loss function and $\mathcal{G} := \{(x,y) \mapsto L(y,h(x)) \mid h \in \mathcal{F}\}$. Denote the restriction of $S = \big((x_i,y_i)\big)_{i=1}^n \in (\mathcal{X}\times\{-1,1\})^n$ to $\mathcal{X}$ by $S_{\mathcal{X}} := (x_i)_{i=1}^n$. For any probability measure P on $\mathcal{X}\times\{-1,1\}$ with marginal p on $\mathcal{X}$ we have*

$$\hat{\mathcal{R}}_S(\mathcal{G}) = \frac{1}{2}\hat{\mathcal{R}}_{S_{\mathcal{X}}}(\mathcal{F}) \quad \textit{and} \quad \mathcal{R}_{n,P}(\mathcal{G}) = \frac{1}{2}\mathcal{R}_{n,p}(\mathcal{F}). \tag{1.37}$$

Proof. The second equation is obtained from the first by taking the expectation value. The first is obtained by exploiting that $L(y,h(x)) = (1-yh(x))/2$. Then

$$\begin{aligned}\hat{\mathcal{R}}_S(\mathcal{G}) &= \mathbb{E}_\sigma\left[\sup_{h\in\mathcal{F}}\frac{1}{n}\sum_{i=1}^n \sigma_i\big(1-y_ih(x_i)\big)/2\right] \\ &= \frac{1}{2}\mathbb{E}_\sigma\left[\sup_{h\in\mathcal{F}}\frac{1}{n}\sum_{i=1}^n \sigma_i h(x_i)\right] = \frac{1}{2}\hat{\mathcal{R}}_{S_{\mathcal{X}}}(\mathcal{F}),\end{aligned}$$

where we have used that $\mathbb{E}_\sigma[\sigma_i] = 0$ and that the distributions of $-\sigma_i y_i$ and σ_i are the same. □

If, similar to the last part of the proof, we use that σ_i and $-\sigma_i$ are equally distributed, we can write

$$\hat{\mathcal{R}}_{S_{\mathcal{X}}}(\mathcal{F}) = \mathbb{E}_\sigma\left[\sup_{h\in\mathcal{F}}\frac{1}{n}\sum_{i=1}^n -\sigma_i h(x_i)\right] = -\mathbb{E}_\sigma\left[\inf_{h\in\mathcal{F}}\frac{1}{n}\sum_{i=1}^n \sigma_i h(x_i)\right].$$

Hence, computing the empirical Rademacher complexity is an optimization problem similar to empirical risk minimization—so it may be hard. The Rademacher complexity $\mathcal{R}_n$ itself depends on an unknown distribution and is therefore difficult to estimate, as well. However, it can be bounded for instance in the binary case in terms of the growth function or the VC-dimension. More specifically,

$$\mathcal{R}_n(\mathcal{F}) \leq \sqrt{\frac{2\ln\Gamma(n)}{n}} \quad \text{and} \quad \mathcal{R}_n(\mathcal{F}) \leq C\sqrt{\frac{\mathrm{VCdim}(\mathcal{F})}{n}}, \tag{1.38}$$

for some universal constant C.

Let us finally collect some properties of the Rademacher complexities, which turn out to be useful for their application.

Theorem 1.10: Properties of Rademacher complexities

Let $\mathcal{G}, \mathcal{G}_1, \mathcal{G}_2 \subseteq \mathbb{R}^{\mathcal{Z}}$ be classes of real-valued functions on $\mathcal{Z}$ and $z \in \mathcal{Z}^n$. The following holds for the empirical Rademacher complexities w.r.t. z:

1. If $c \in \mathbb{R}$, then $\hat{\mathcal{R}}(c\mathcal{G}) = |c|\hat{\mathcal{R}}(\mathcal{G})$.
2. $\mathcal{G}_1 \subseteq \mathcal{G}_2$ implies $\hat{\mathcal{R}}(\mathcal{G}_1) \leq \hat{\mathcal{R}}(\mathcal{G}_2)$.
3. $\hat{\mathcal{R}}(\mathcal{G}_1 + \mathcal{G}_2) = \hat{\mathcal{R}}(\mathcal{G}_1) + \hat{\mathcal{R}}(\mathcal{G}_2)$.
4. $\hat{\mathcal{R}}(\mathcal{G}) = \hat{\mathcal{R}}(\text{conv } \mathcal{G})$, where conv denotes the convex hull.
5. If $\varphi : \mathbb{R} \to \mathbb{R}$ is $L-$Lipschitz, then $\hat{\mathcal{R}}(\varphi \circ \mathcal{G}) \leq L\, \hat{\mathcal{R}}(\mathcal{G})$.

Proof. (sketch) 1.-3. follow immediately from the definition.

4. follows from the simple observation that

$$\sup_{\lambda \in \mathbb{R}^m_+, ||\lambda||_1 = 1} \sum_{i=1}^{n} \sum_{l=1}^{m} \lambda_l \sigma_i g_l(z_i) = \max_{l \in \{1,\ldots,m\}} \sum_{i=1}^{n} \sigma_i g_l(z_i).$$

5. Define $V := \{v \in \mathbb{R}^n \mid \exists g \in \mathcal{G}\ \forall i :\ v_i = g(z_i)\}$. Then

$$\begin{aligned} n\hat{\mathcal{R}}(\varphi \circ \mathcal{G}) &= \mathbb{E}_\sigma \left[\sup_{v \in V} \sum_{i=1}^{n} \sigma_i \varphi(v_i) \right] && (1.39) \\ &= \frac{1}{2} \mathbb{E}_{\sigma_2,\ldots,\sigma_n} \left[\sup_{v,v' \in V} \varphi(v_1) - \varphi(v_1') + \sum_{i=2}^{n} \sigma_i \big(\varphi(v_i) + \varphi(v_i')\big) \right] \\ &\leq \frac{1}{2} \mathbb{E}_{\sigma_2,\ldots,\sigma_n} \left[\sup_{v,v' \in V} L|v_1 - v_1'| + \sum_{i=2}^{n} \sigma_i \big(\varphi(v_i) + \varphi(v_i')\big) \right] \\ &= \mathbb{E}_\sigma \left[\sup_{v \in V} L\sigma_1 v_1 + \sum_{i=2}^{n} \sigma_i \varphi(v_i) \right], && (1.40) \end{aligned}$$

where in the last step we used that the absolute value can be dropped since the expression is invariant w.r.t. interchanging $v \leftrightarrow v'$. Repeating the above steps for the other $n-1$ components then leads to the claimed result. □

Remark: sometimes the definition of the (empirical) Rademacher complexity in the literature differs from the one in Eqs.(1.29, 1.30) and the absolute value is taken, i.e., the empirical quantity is defined as $\mathbb{E}_\sigma \left[\sup_g \left| \sum_i \sigma_i g(x_i) \right|\right]$ instead. In this case Thm.1.10 essentially still holds with small variations: then 3. becomes an inequality '$\leq$' and 5. requires in addition that $\varphi(-z) = -\varphi(z)$ (see [1]).

Lemma 1.12 (Massart's Lemma)**.** *Let A be a finite subset of $\mathbb{R}^m$ that is contained in a Euclidean ball of radius r. Then*

$$\mathbb{E}_\sigma \left[\max_{a \in A} \sum_{i=1}^{m} \sigma_i a_i \right] \leq r\sqrt{2 \ln |A|}, \qquad (1.41)$$

where the expectation value is w.r.t. uniformly distributed Rademacher variables $\sigma \in \{-1,1\}^m$.

Proof. W.l.o.g. we can assume that the center of the ball is at the origin since Eq.(1.41) is unaffected by a translation. We introduce a parameter $\lambda > 0$ to be chosen later and first compute an upper bound for the rescaled set λA:

$$\mathbb{E}_\sigma\left[\max_{a\in\lambda A}\sum_{i=1}^m \sigma_i a_i\right] \;\leq\; \mathbb{E}_\sigma\left[\ln\sum_{a\in\lambda A} e^{\sigma\cdot a}\right] \;\leq\; \ln\mathbb{E}_\sigma\left[\sum_{a\in\lambda A} e^{\sigma\cdot a}\right] \tag{1.42}$$

$$\leq\; \ln\sum_{a\in\lambda A}\prod_{i=1}^m \frac{e^{a_i}+e^{-a_i}}{2} \tag{1.43}$$

$$\leq\; \ln\sum_{a\in\lambda A} e^{||a||_2^2/2} \;\leq\; \frac{1}{2}r^2\lambda^2 + \ln|A|. \tag{1.44}$$

Here, the first step is most easily understood when taking the exponential on both sides of the inequality for a fixed value of σ. Then the first inequality in Eq.(1.42) reduces to the statement that the maximum over positive numbers can be upper bounded by their sum. The second inequality uses concavity of the logarithm together with Jensen's inequality. Eq. (1.43) uses that the σ_i's are independently and uniformly distributed. The step to Eq.(1.44) exploits that $e^x + e^{-x} \leq 2e^{x^2/2}$ holds for all $x \in \mathbb{R}$. The final inequality then bounds the sum by its maximal element multiplied by the number of terms.

We then obtain the claimed result by inserting $\lambda = \sqrt{2\ln|A|}/r$ into

$$\mathbb{E}_\sigma\left[\max_{a\in A}\sum_{i=1}^m \sigma_i a_i\right] \;\leq\; \left(\frac{1}{2}r^2\lambda^2 + \ln|A|\right)/\lambda.$$

□

1.9 AdaBoost

Ensemble methods are meta-algorithms that combine several machine learners or predictors to form a more powerful one. A famous example for the success of ensemble methods is the winner of the $1M Netflix prize in 2009, which substantially improved Netflix' recommender system. All the top submissions in that competition were combinations of combinations of ... hundreds of predictors.

We will describe one of the most common ensemble methods for binary classification, the *AdaBoost* (short for *Ada*ptive *Boost*ing). The starting point of this method is a hypotheses class $\mathcal{F} \subseteq \{-1,1\}^{\mathcal{X}}$ whose elements are called *base hypotheses*. AdaBoost is an iterative method that when stopped after the T'th iteration returns a hypothesis of the form

$$f := \operatorname{sgn}\left(\sum_{t=1}^T w_t h_t\right), \quad w_t \in \mathbb{R}, \quad h_t \in \mathcal{F}. \tag{1.45}$$

That is, f is constructed so that its prediction is a weighted majority vote of the predictions of T base hypotheses. Note that $f \notin \mathcal{F}$, unless $\mathcal{F}$ is incidentally closed under such operations. In every iteration of AdaBoost an ERM algorithm for $\mathcal{F}$ is called as a subroutine, which returns one of the h_t's. The key idea is that the empirical risk that is minimized within this subroutine assigns different weights to the training data instances. The algorithm puts more weight on those instances that appear to be hard in the sense that they were misclassified by the previous h_t's. Suppose the training data S consists of n pairs $(x_i, y_i) \in \mathcal{X} \times \{-1, 1\}$. Let $p^{(t)}$ be a yet to be constructed probability distribution over S that is used in the t'th iteration. Define by

$$\epsilon_t := \sum_{i=1}^{n} p_i^{(t)} \mathbb{1}_{h_t(x_i) \neq y_i} \tag{1.46}$$

the $p^{(t)}$-weighted empirical risk of h_t, i.e., the error probability of h_t on S when the entries in S are weighted according to $p^{(t)}$. Given $p^{(t)}$, the hypothesis h_t is ideally chosen so that it minimizes this weighted empirical risk. We will, however, treat the selection of h_t as a black box and do not require that h_t really minimizes the weighted risk. ϵ_t is simply defined as in Eq.(1.46), whether this is optimal or not. From here define

$$w_t := \frac{1}{2} \ln \left(\frac{1 - \epsilon_t}{\epsilon_t} \right). \tag{1.47}$$

The details of this choice will become clear later. For now, observe that w_t increases with decreasing ϵ_t and that $w_t \geq 0$ whenever $\epsilon_t \leq \frac{1}{2}$, i.e., whenever the hypothesis h_t performs at least as good as random guessing. The update rule for the probability distribution then reads

$$\begin{aligned} p_i^{(t+1)} &:= \frac{p_i^{(t)}}{Z_t} \times \begin{cases} e^{-w_t} & \text{if } h_t(x_i) = y_i \\ e^{w_t} & \text{if } h_t(x_i) \neq y_i \end{cases} \\ &= p_i^{(t)} e^{-w_t y_i h_t(x_i)} / Z_t, \end{aligned}$$

where Z_t is a normalization factor chosen so that $\sum_{i=1}^{n} p_i^{(t+1)} = 1$. Note that the step from $p^{(t)}$ to $p^{(t+1)}$ aims at increasing the weight that corresponds to (x_i, y_i) if x_i has been misclassified by h_t (in case h_t performs better than random guessing).

Upon input of the training data S, AdaBoost starts with a uniform distribution $p^{(1)}$ and iterates the above procedure, where in each iteration ϵ_t, w_t, h_t and $p^{(t+1)}$ are computed. The number T of iterations is a free parameter which essentially allows to balance between the estimation error and the approximation error. If the class $\mathcal{F}$ of base hypotheses is simple, then small T may lead to large approximation error, whereas choosing T very large makes it more likely that overly complex hypotheses are returned.

The following theorem shows that the empirical risk can decrease rapidly with increasing T:

Theorem 1.11: Empirical risk bound for AdaBoost

Let f be the hypothesis that is returned after T iterations of AdaBoost that led to intermediate weighted empirical risks $\epsilon \in [0,1]^T$. Then the error probability of f on the training data set is bounded by

$$\hat{R}(f) \leq \prod_{t=1}^{T} 2\sqrt{\epsilon_t(1-\epsilon_t)}. \tag{1.48}$$

With $\gamma := \min\{|\epsilon_t - 1/2|\}_{t=1}^T$ this implies in particular $\hat{R}(f) \leq \exp[-2\gamma^2 T]$.

Proof. Define $F := \sum_{t=1}^T w_t h_t$ and observe that with $p_i^{(1)} = 1/n$ we can write

$$\begin{aligned} p_i^{(T+1)} &= p_i^{(1)} \times \frac{e^{-w_1 y_i h_1(x_i)}}{Z_1} \times \cdots \times \frac{e^{-w_T y_i h_T(x_i)}}{Z_T} \\ &= \frac{e^{-y_i F(x_i)}}{n \prod_{t=1}^T Z_t}. \end{aligned} \tag{1.49}$$

If $f(x_i) \neq y_i$, then $y_i F(x_i) \leq 0$, which implies $e^{-y_i F(x_i)} \geq 1$. Therefore,

$$\hat{R}(f) = \frac{1}{n}\sum_{i=1}^n \mathbb{1}_{f(x_i)\neq y_i} \leq \frac{1}{n}\sum_{i=1}^n e^{-y_i F(x_i)} = \prod_{t=1}^T Z_t, \tag{1.50}$$

where the last step uses Eq.(1.49) together with the fact that the $p_i^{(T+1)}$'s sum up to 1. Next, we write the normalization factors Z_t in a more suitable form:

$$\begin{aligned} Z_t &= \sum_{i=1}^n p_i^{(t)} e^{-w_t y_i h_t(x_i)} = \sum_{i:h_t(x_i)\neq y_i} p_i^{(t)} e^{w_t} + \sum_{i:h_t(x_i)=y_i} p_i^{(t)} e^{-w_t} \\ &= \epsilon_t e^{w_t} + (1-\epsilon_t)e^{-w_t} = 2\sqrt{\epsilon_t(1-\epsilon_t)}, \end{aligned} \tag{1.51}$$

where we have inserted w_t from Eq.(1.47). This completes the proof of Eq.(1.48). In order to arrive at the second claim of the theorem, we use that $1 - x \leq e^{-x}$ holds for all $x \in \mathbb{R}$, which allows us to bound

$$2\sqrt{\epsilon_t(1-\epsilon_t)} = \sqrt{1 - 4(\epsilon_t - 1/2)^2} \leq \exp\big[-2(\epsilon_t - 1/2)^2\big].$$

□

The proof reveals two more things about AdaBoost. First, we can understand the particular choice of the w_t's. Looking at Eq.(1.51) one is tempted to choose them so that they minimize the expression $\epsilon_t e^{w_t} + (1-\epsilon_t)e^{-w_t}$ and, indeed, this is exactly what the choice in Eq.(1.47) does. Second, notice that after inserting all expressions we obtain

$$\sum_{i:h_t(x_i)=y_i} p_i^{(t+1)} = \sum_{i:h_t(x_i)=y_i} \frac{p_i^{(t)}}{Z_t} e^{-w_t} = (1-\epsilon_t)\frac{e^{-w_t}}{Z_t} = \frac{1}{2}.$$

This means that in every iteration the new probability distribution $p^{(t+1)}$ is chosen so that the correctly classified instances all together get total weight one half (and so do the misclassified ones). Hence, $p^{(t+1)}$ can be computed from $p^{(t)}$ by a simple rescaling of the probabilities of these two sets.

Thm.1.11 shows that if we manage to keep the error probabilities ϵ_t a constant γ away from $1/2$ (the performance of flipping a coin), the empirical risk will decrease exponentially in the number T of iterations. More precisely, it is upper bounded by a decreasing exponential—it does in fact not have to decrease monotonically itself.

In order to get a theoretical bound on the risk, i.e., on the performance beyond the training data, we look at the VC-dimension:

Theorem 1.12: VC-dimension of linearly combined classifiers

Let d be the VC-dimension of $\mathcal{F} \subseteq \{-1,1\}^{\mathcal{X}}$ and for $T \in \mathbb{N}$ define $\mathcal{F}_T := \{f = \operatorname{sgn} \sum_{t=1}^{T} w_t h_t \mid w_t \in \mathbb{R},\ h_t \in \mathcal{F}\}$. Then the growth function Γ of $\mathcal{F}_T$ satisfies

$$\Gamma(n) \leq \left(\frac{en}{T}\right)^T \left(\frac{en}{d}\right)^{Td} \quad \text{and} \tag{1.52}$$

$$\operatorname{VCdim}(\mathcal{F}_T) \leq 2T(d+1)\log_2\big(2eT(d+1)\big). \tag{1.53}$$

Proof. In order to bound the growth function we regard $\mathcal{F}_T = \mathcal{G} \circ \mathcal{H}$ as a composition of two function classes

$$\mathcal{G} := \Big\{ g : \mathbb{R}^T \to \{-1,1\} \;\Big|\; g(z) = \operatorname{sgn} \sum_{i=1}^{T} w_i z_i,\ w_i \in \mathbb{R} \Big\},$$

$$\mathcal{H} := \Big\{ h : \mathcal{X} \to \mathbb{R}^T \;\Big|\; h(x) = \big(h_1(x), \ldots, h_T(x)\big),\ h_t \in \mathcal{F} \Big\}.$$

Following Lemma 1.4 we have $\Gamma(n) \leq \Gamma_{\mathcal{G}}(n)\Gamma_{\mathcal{H}}(n)$ where $\Gamma_{\mathcal{G}}$ and $\Gamma_{\mathcal{H}}$ denote the growth functions of $\mathcal{G}$ and $\mathcal{H}$, respectively. Since the VC-dimension of $\mathcal{G}$ is equal to T by Thm.1.7, we can apply Thm.1.6 and obtain $\Gamma_{\mathcal{G}}(n) \leq (en/T)^T$. The product structure of $\mathcal{H}$ implies that $\Gamma_{\mathcal{H}}(n) = \Gamma_{\mathcal{F}}(n)^T$ where $\Gamma_{\mathcal{F}}$ denotes the growth function of $\mathcal{F}$. The latter can by Thm.1.6 be bounded in terms of the VC-diemsnion so that $\Gamma_{\mathcal{F}}(n) \leq (en/d)^d$. Collecting the terms this finally leads to Eq.(1.52).

In order to arrive at a bound for the VC-dimension, note that $D \geq \operatorname{VCdim}(\mathcal{F}_T)$ if $2^D > \Gamma(D)$. Inserting the upper bound on the growth function from Eq.(1.52) this is implied by

$$D > T(d+1)\log_2(eD) - T\log_2 T - dT\log_2 d.$$

Straight forward calculation shows that this is satisfied, if we choose D equal to the r.h.s. of Eq.(1.53). □

Comparing the scaling of this bound for the VC-dimension with the one of the empirical risk in Thm.1.11 is already promising: while the VC-dimension grows

not much faster than linearly with T, the empirical risk ideally decreases exponentially. In practice, AdaBoost has been observed to be remarkably resistant against overfitting as long as the data is not too noisy.

From a purely theoretical perspective, AdaBoost shows that a priori different notions of learnability coincide. Consider a weak and a strong notion of learnability, where the strong notion is the one we discussed in the context of PAC learnability. This requires something for all $\epsilon \in (0, 1]$ where the weak notion would only ask for $\epsilon \in (1/2 - \gamma, 1]$ for some fixed, possibly small $\gamma > 0$. Then AdaBoost can be used to 'boost' learnability from weak to strong and to show that these two notions actually coincide.

Chapter 2

Neural networks

2.1 Information processing in the brain

The human brain contains about 10^{11} *neurons*, which can be regarded as its basic information processing units. A typical neuron consist of a *cell body*, *dendrites*, which receive incoming signals from other neurons and an *axon*, which transmits signals to other neurons. While there are typically several dendrites originating from the cell body and then branching out in the neighborhood of the neuron, there is only one axon, which may have a local branching in the neighborhood of the cell body and a second branching at a distance. This can mean everything from 0.1mm to 2m.

On the macroscopic scale, if we regard the human brain as a whole, we see it covered with a folded outer layer, which is about 3mm thick, and called the *cerebral cortex*. The largest part of the cerebral cortex is also its evolutionary youngest part and for this reason called *neocortex*. The neocortex plays a crucial role in many higher brain functions.

If we look at slices of the brain, we see the cerebral cortex as *grey matter* clearly separated from the *white matter*, which it surrounds. White matter almost exclusively consists of axons that connect more distant parts of the brain. The axons originate from neurons (mainly so-called pyramidal neurons, named after their shape), which are part of the grey matter, then leave the grey matter, traverse parts of the brain in the white matter, which is formed by them, and then reenter the grey matter and connect to other neurons. In this sense, white matter is related to (long distance) communication, whereas information storage and processing happens in the grey matter. The difference in color stems from the myelin, a fatty white substance which covers the axons in the white matter. The main purpose of the myelin sheaths is to increase the speed at which signals travel down the axons. Therefore, only the long distance connections are covered with myelin.

A typical pyramidal neuron in the neocortex forms a highly connected local network in the grey matter where it is connected to about 10^4 of its neigh-

bors that are less then 1mm apart. In addition, via the axon traversing the white matter, the neuron is connected to a similar number of distant neurons. There is evidence that connections between different regions of the neocortex are typically two-way connections.

The neocortex is very homogeneous throughout the brain so that different functions that are assigned to different areas are not obviously reflected physiologically. The assignment of special functions to specific areas clearly depends on which parts or sensory inputs the area is connected to.

The signals between neurons are electrical pulses that originate in a change of the electrical potential of in total about 100mV—the *action potential*. Such a pulse takes about 1ms and travels down the axon where it reaches so-called *synapses* at the axon's branches. A synapse connects the axon of one neuron with the dendrite of another neuron. The signal transmission within most synapses is of chemical nature. The arriving electrical pulse induces a chemical process inside the synapse, which in turn leads to a change of electrical potential in the postsynaptic neuron. The time it takes for a signal to pass a chemical synapse is around 1ms.

In the dendritic part of the postsynaptic neuron all the incoming signals are integrated. If they lead to a joint stimulus above a certain threshold, they will cause an action potential and the neuron will fire. This is an all-or-nothing process and all stimuli above threshold lead to the same pulse with standardized amplitude and duration. While the outgoing signal in this way can be considered digital, the integration/accumulation of the incoming signals appears to be more of analog nature.

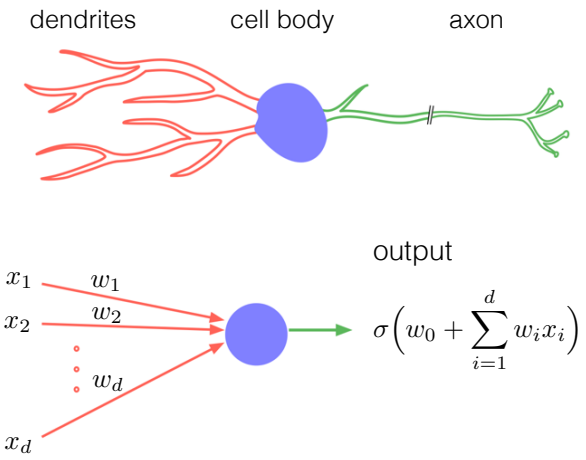
The effect of an incoming pulse on the postsynaptical neuron can vary significantly—over time, in duration and in strength. It may change the potential from milliseconds to minutes and during this period have an excitatory or an inhibitory effect.[1] The variability of the strength of this effect is considered crucial for purposes of learning and memorization.

A neuron can fire several hundred times per second (while the average firing rate is closer to 1Hz). A limiting factor to higher rates is the duration of each pulse and a corresponding *refractory period* of about 1ms after initiation of an action potential during which no stimulus can lead to firing. Within about 4ms after this strict refractory period stimuli still have to be stronger than usual to lead to an action potential. This period is called *relative refractory period*.

Everything said in this section merely describes (in a nutshell) the basic structure and physiology that is thought to be relevant for information processing in the brain. How memory and learning actually work on the basis of the described pieces, is much less understood and has to be left aside here.

Let us finally make a rough comparison between the human brain and present day computers in terms of the basic numbers. The power consumption of the brain is around 20 Watts and thus about the same as the one of a present day laptop. Also the estimated number of neurons (10^{11}) is not too far from the

[1] However, connections between pyramidal neurons, which are believed to correspond to the majority of synapses in the cerebral cortex, are exclusively excitatory.



number of transistors, which is $10^9 - 10^{10}$ on a state-of-the-art chip. Significant differences lie in the connectivity, the frequency and the related logical depth. A transistor is only directly connected to a few others, it runs with a clock rate of several GHz and is involved in computations of enormous logical depth. A neuron in comparison is connected to 10^4 or more others, but operates at frequencies of only a few hundred Hz, which is a factor of 10^7 below the computers clock rates. Since most 'computations' in the brain are nevertheless done within a fraction of a second, they cannot have logical depth significantly beyond 100.

2.2 From Perceptrons to networks

Artifical neurons A simple artificial neuron model that incorporates some of the properties of biological neurons described in the last section is the *Perceptron*, introduced by Rosenblatt in 1958. More specifically, the Perceptron incorporates (i) several inputs whose effects are determined by variable weights, (ii) a single output (which may, however, by copied/fanned out an arbitrary number of times), (iii) integration of input signals and (iv) an all-or-nothing process with adjustable threshold.

Mathematically, each input is characterized by a real number x_i where $i = 1, \ldots, d$ runs over the number of inputs. Each of the input lines gets assigned a weight $w_i \in \mathbb{R}$. The mapping from the inputs to the output is then modeled by

$$x \mapsto \sigma\left(w_0 + \sum_{i=1}^{d} w_i x_i\right), \tag{2.1}$$

where $w_0 \in \mathbb{R}$ plays the role of a threshold value and the *activation function* $\sigma : \mathbb{R} \to \mathbb{R}$ is, in the case of the Perceptron, given by the step-function $\sigma(z) = \mathbb{1}_{z \geq 0}$. It is convenient to regard w_0 as the weight of a constant input $x_0 = 1$. Nowadays, one usually considers generalizations of this model that differ from

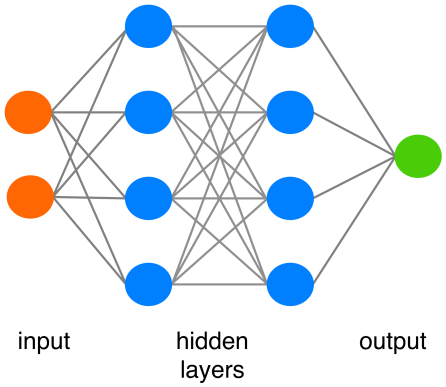


Figure 2.1: Structure of a feedforward neural network with two inputs, two hidden layers with four neurons each and a single output neuron.

the original Perceptron in the choice of the activation function. The main reason for choosing different activation functions is that they enable gradient descent techniques for learning algorithms. Common choices are:

- *Logistic sigmoid*[2] $\sigma(z) = \frac{1}{1+e^{-z}}$,
- *Hyperbolic tangent* $\sigma(z) = \tanh(z)$ (which is an affinely transformed logistic sigmoid),
- *Rectified linear* $\sigma(z) = \max\{0, z\}$. This is sometimes modified to $\max\{\alpha x, x\}$ with some $\alpha \in (0, 1)$.

Note that the logistic sigmoid function and the hyperbolic tangent are smooth versions of the step functions $\mathbb{1}_{z\geq 0}$ and $\operatorname{sgn}(z)$, respectively. In practice, rectified linear activation functions and the tanh seem to be most common.

A multivariate function of the form $\mathbb{R}^d \ni x \mapsto \sigma(w \cdot x)$ with $\sigma : \mathbb{R} \to \mathbb{R}$ is often called a *ridge function* - especially in the context of approximation theory. Note that every ridge function is constant on hyperplanes characterized by $w \cdot x = c$. In the case of the Perceptron with $\sigma(z) = \mathbb{1}_{z\geq 0}$ we have that the set of points in $\mathbb{R}^d$ that are mapped to 1 forms a closed half space. This relation between Perceptrons and half spaces is obviously bijective and often provides a useful geometric depiction.

Neural networks Composing several artificial neurons of the type just introduced by making use of the possibility to fan out their outputs we obtain a *neural network*. This is then described by a weighted directed graph $G = (V, E)$ where vertices correspond to single neurons or input/output nodes, directions mark the flow of signals and the weights are the w_i's assigned to every individual neuron's inputs. For a weighted directed graph to represent a neural network

[2] "Sigmoidal" just means S-shaped.

in the usual sense, there need to be input and output nodes, i.e., vertices with only outgoing and only incoming edges, respectively. In addition, we have to assign an individual threshold value w_0 to every neuron and choose an activation function σ. The latter is often chosen equally for all neurons. A natural exception is when dealing with regression problems where one often omits the activation functions at the output nodes (or, equivalently, chooses the identity as activation function).

The graph underlying a neural network is called the network's *architecture*, which then neglects the values of weights and thresholds. If the graph is an *acyclic* directed graph, meaning it does not contain directed cycles, then the network is called a *feedforward* network. Otherwise, it is called a *recurrent* network. A particular class of feedforward neural networks are *multilayered feedforward neural networks.* In this case the vertives $V = \bigcup_{l=0}^{m} V_l$ are arranged into disjoint layers $\{V_l\}_{l=0}^{m}$ so that connections only exist between neighboring layers, i.e., $E \subseteq \bigcup_{l=0}^{m-1} \{(u, v) | u \in V_l, v \in V_{l+1}\}$. m is then called the *depth* of the network and $m - 1$ is the number of *hidden layers* ("hidden" in the sense of in between input and output). In the following we will focus on multilayered feedforward networks. If there is no further specification, we will always assume that neighboring layers are *fully connected*, i.e., every element of V_l is connected to every element in V_{l+1}.

2.3 Representation and approximation

A neural network with d inputs and d' outputs represents a function $f : \mathbb{R}^d \to \mathbb{R}^{d'}$. In this section we address the question which functions can be represented (exactly or approximately) by a multilayered feedforward neural network, depending on the architecture and on the chosen activation function. We will start with the discrete case.

Representation of Boolean functions As a warm-up consider a single Perceptron with two Boolean inputs. Can it represent basic Boolean functions like AND, OR, NAND or XOR? The simple but crucial observation for answering this question is that the function $f_w : \mathbb{R}^d \to \mathbb{R}$ in Eq.(2.1) that describes a Perceptron is, for every choice of the weights $w \in \mathbb{R}^{d+1}$, constant on hyperplanes that are orthogonal to the vector $(w_1, \ldots, w_d)$. Moreover, due to the special choice of σ as a step function f_w separates half-spaces and by choosing suitable weights any half-space can be separated from its complement.

If we regard the inputs of AND, OR or NAND as points in $\mathbb{R}^2$, then in all three cases the subsets that are mapped to 0 or 1 can be separated from each other by a line. Consequently, AND, OR and NAND can be represented by a single Perceptron. This is already somewhat promising since we know that every Boolean function can be obtained as a composition of many of such building blocks. XOR, on the other hand, cannot be represented by a single Perceptron since in this case the inputs that are mapped to 0 cannot be linearly separated from the ones mapped to 1. This implies that representing an arbitrary Boolean

function by a feedforward neural network requires at least one hidden layer. The following theorem shows that a one hidden layer is already sufficient.

Theorem 2.1: Representation of Boolean functions

Every Boolean function $f : \{0,1\}^d \to \{0,1\}$ can be represented exactly by a feedforward neural network with a single hidden layer containing at most 2^d neurons, if $\sigma(z) = \mathbb{1}_{z\geq 0}$ is used as activation function.

Proof. If $a, b \in \{0,1\}$ are Boolean variables, then $2ab - a - b \leq 0$ with equality iff $a = b$. With this observation we can write $\mathbb{1}_{x=u} = \sigma\big(\sum_{i=1}^d 2x_iu_i - x_i - u_i\big)$ for $x, u \in \{0,1\}^d$. Denoting by $A := f^{-1}\big(\{1\}\big)$ the set of all vectors u for which $f(u) = 1$, we can then represent f as

$$f(x) = \sigma\Big(-1 + \sum_{u\in A} \mathbb{1}_{x=u}\Big) = \sigma\Big(-1 + \sum_{u\in A} \sigma\big(\sum_{i=1}^d 2x_iu_i - x_i - u_i\big)\Big), \quad (2.2)$$

which is the sought representation using a single hidden layer with $|A| \leq 2^d$ neurons. □

We will see in Sec.2.4 that the exponential increase of the number of neurons cannot be avoided.

Binary classification in $\mathbb{R}^d$:

Theorem 2.2: Binary classification of finite sets in $\mathbb{R}^d$

Let $A = \{x_1, \ldots, x_N\}$ be a finite subset of $\mathbb{R}^d$ and $f : A \to \{-1, 1\}$ arbitrary. There is a feedforward neural network that implements a function $F : \mathbb{R}^d \to \{-1, 1\}$ with a single hidden layer containing $m \leq N$ neurons and using $\sigma = \text{sgn}$ as activation function so that $F|_A = f$. If the points in A are in general position (i.e., no hyperplane in $\mathbb{R}^d$ contains more than d of them), then $m \leq 2\lceil N/(2d)\rceil$ neurons suffice.

Proof. Denote by A_+ and A_- the subsets of A that are mapped to 1 and -1, respectively. W.l.o.g. assume that $|A_+| \leq |A_-|$ so that $|A_+| \leq N/2$. For ever $x \in A_+$ we can find a hyperplane $H := \{z \in \mathbb{R}^d | a \cdot z + b = 0\}$ characterized by $a \in \mathbb{R}^d$ and $b \in \mathbb{R}$ so that $A \cap H = \{x\}$. Due to finiteness of A we can now find two hyperplanes that are parallel to H, contain H and thus x in between them, but none of the other points from A. In other words, we can choose $\epsilon \neq 0$ appropriately, so that the map $z \mapsto \sigma(\epsilon + a \cdot z + b) + \sigma(\epsilon - a \cdot z - b)$ takes on the value 2 for $z = x$ but is zero on $A \setminus \{x\}$. Repeating this for all points in A_+ we can finally construct

$$F(z) := \sigma\left(-1 + \sum_{x\in A_+} \sigma\big(\epsilon_x + a_x \cdot z + b_x\big) + \sigma\big(\epsilon_x - a_x \cdot z - b_x\big)\right), \quad (2.3)$$

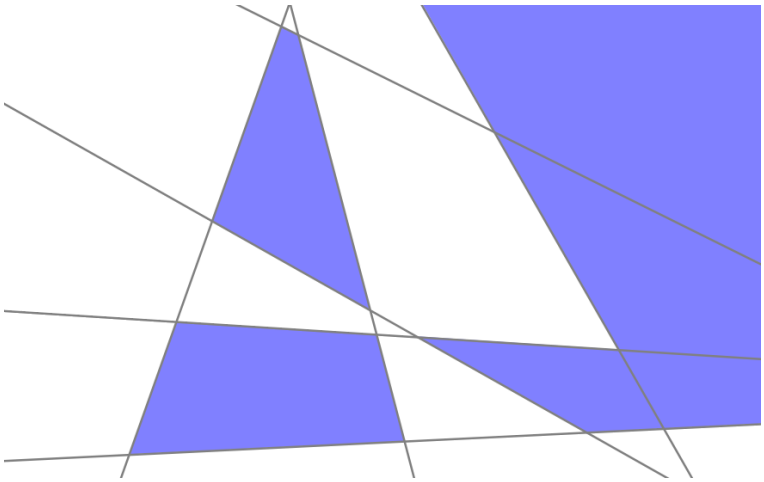


Figure 2.2: The subsets of $\mathbb{R}^d$ that are classified by a neural network with a single hidden layer containing m hidden neurons with $\sigma(z) = \mathbb{1}_{z\geq 0}$ are unions of convex polyhedra that are obtained as intersections of subsets of m half spaces. Here $m = 7$, the hyperplanes that delimit the half spaces are gray lines and the blue regions exemplify $f^{-1}(\{1\})$.

so that $F|_A = f$. Then F has the form of a neural network with a single hidden layer that contains $m = 2|A_+| \leq N$ neurons.

Now assume the points in A are in general position. Then we can in every (but the last) step of the construction choose the hyperplane H so that it contains d points from A_+ and no other point from A. In this way, we reduce the number of terms essentially by a factor d and we get $m \leq 2\lceil N/(2d)\rceil$. □

Let us consider binary classification of subsets of $\mathbb{R}^d$ via neural networks from a more geometric point of view. Consider a network with a single hidden layer with m neurons and $\sigma(z) = \mathbb{1}_{z\geq 0}$ as activation function. As mentioned before, every individual Perceptron can be characterized by a half space, say H_j for the j'th hidden neuron, in such a way that the output of the Perceptron upon input x is given by the value of the indicator function $\mathbb{1}_{x\in H_j}$. In this way we can write the function $f : \mathbb{R}^d \to \{0, 1\}$ that is implemented by the network as

$$f(x) = \sigma\Big(w_0 + \sum_{j=1}^{m} w_j \mathbb{1}_{x\in H_j}\Big).$$

Defining by $\mathcal{A} := \{A \subseteq \{1, \ldots, m\} \mid \sum_{j\in A} w_j \geq -w_0\}$ the set of all subsets of hidden neurons that are capable of activating the output neuron by firing together, we can write

$$f^{-1}(\{1\}) = \bigcup_{A\in\mathcal{A}} \bigcap_{j\in A} H_j. \tag{2.4}$$

Note that $\bigcap_{j\in A} H_j$ is, as an intersection of at most m half spaces, a convex polyhedron with at most m facets. Hence, the set of points that are mapped to 1 by the network are given by a union of convex polyhedra that are obtained as intersections of some of m half spaces.

Approximating real-valued functions We will begin with the one-dimensional case $f : \mathbb{R} \to \mathbb{R}$ and later lift the obtained results to the case of higher dimensional input and output spaces. Let us denote by $\mathcal{F}_{\sigma,m}$ the class of functions representable by a feedforward network with a single hidden layer with m neurons in the hidden layer. That is,

$$\mathcal{F}_{\sigma,m} := \Big\{ f : \mathbb{R} \to \mathbb{R} \;\Big|\; f(x) = \sum_{v=1}^{m} a_v \sigma(w_v + b_v),\ a_v, b_v, w_v \in \mathbb{R} \Big\} \tag{2.5}$$

and define $\mathcal{F}_\sigma := \bigcup_{m\in\mathbb{N}} \mathcal{F}_{\sigma,m}$. Note that we now omit the application of the activation function at the output.

The following approximation theorem is formulated in terms of the *modulus of continuity* of the function f to be approximated. It is defined as

$$\omega(f, \delta) := \sup_{x,y:|x-y|\le\delta} \big|f(x) - f(y)\big|. \tag{2.6}$$

Note that if f is continuous, then $\omega(f, \delta) \to 0$ for $\delta \to 0$. Moreover, if f is $L-$Lipschitz, then $\omega(f, \delta) \le \delta L$.

Theorem 2.3: Approximations using bounded sigmoids

Let $\sigma : \mathbb{R} \to \mathbb{R}$ be any bounded function that satisfies $\lim_{z\to-\infty} \sigma(z) = 0$ and $\lim_{z\to\infty} \sigma(z) = 1$. There is a constant c so that for every $f \in C\big([0, 1]\big)$ and every $m \in \mathbb{N}$ we have

$$\inf_{f_m\in\mathcal{F}_{\sigma,m}} ||f - f_m||_\infty \le c\, \omega(f, 1/m). \tag{2.7}$$

Note: From the proof we get that $c = 2 + 2||\sigma||_\infty$ is a valid constant (where $||\sigma||_\infty = \sup_{z\in\mathbb{R}} |\sigma(z)|$). A more careful analysis shows that $c = ||\sigma||_\infty$ suffices. The assumption that σ is asymptotically either 0 or 1 is convenient for the proof but not really crucial. The same argument works if only the limits $\lim_{z\to\pm\infty} \sigma(z)$ exist in $\mathbb{R}$ and differ from each other. Similarly, the domain $[0, 1]$ can be replaced by any compact subset of $\mathbb{R}$.

Proof. The idea is to first approximate f by a piecewise constant function h_m and then h_m by an appropriate f_m. Define $x_i := i/m$ and $h_m(x)$ so that it takes the value $f(x_i)$ in the interval $x \in \big[x_{i-1}, x_i\big)$ where $i = 1, \dots, m$. By

construction $||f - h_m||_\infty \leq \omega(f, 1/m)$. With $j := \lfloor mx \rfloor$ write

$$\begin{aligned} h_m(x) &= f(x_1) + \sum_{i=1}^{j} \big(f(x_{i+1}) - f(x_i)\big) \quad \text{and define} \\ f_m(x) &:= f(x_1)\sigma(\alpha) + \sum_{i=1}^{m-1} \big(f(x_{i+1}) - f(x_i)\big)\sigma\big(\alpha(mx - i)\big), \end{aligned} \tag{2.8}$$

for some $\alpha \in \mathbb{R}$ to be chosen shortly. Note that f_m is of the desired form. The claim is that f_m approximates h_m well for large α. To bound the distance between the two functions, fix any $\epsilon > 0$ and choose α such that $|\sigma(z) - \mathbb{1}_{z\geq 0}| \leq \epsilon/m$ whenever $|z| \geq \alpha$. This is possible since σ is assumed to be a sigmoidal function. Note that, by the choice of α, we get that if $i \notin \{j, j+1\}$ the term $\sigma\big(\alpha(mx - i)\big)$ is ϵ/m-close to the step function $\mathbb{1}_{i\leq \lfloor mx \rfloor}$. Consequently, we can bound

$$\begin{aligned} |f_m(x) - h_m(x)| \quad \leq \quad & \frac{\epsilon}{m}\Big[|f(x_1)| + (m-2)\omega(f, 1/m)\Big] \\ & + \big|f(x_{j+1}) - f(x_j)\big| \,\Big|1 - \sigma\big(\alpha(mx - j)\big)\Big| \\ & + \big|f(x_{j+2}) - f(x_{j+1})\big| \,\Big|\sigma\big(\alpha(mx - j - 1)\big)\Big|, \end{aligned}$$

where the r.h.s. of the first line can be made arbitrary small by the choice of ϵ and the sum of the last two lines can be bounded by $\omega(f, 1/m)(1 + 2||\sigma||_\infty)$. □

As a consequence, an arbitrary L-Lipschitz function can be approximated uniformly by a feedforward network with a single hidden layer of m neurons so that the approximation error is $\mathcal{O}(L/m)$. The following proposition characterizes the class of continuous activation functions for which similar approximation results can be obtained.

Proposition 2.1 (Universality of all non-polynomial activation functions)**.** *Let $\sigma \in C(\mathbb{R})$. The set of functions $\mathcal{F}_\sigma$ representable by a neural network with a single hidden layer and activation function σ is dense in $C(\mathbb{R})$ w.r.t. the topology of uniform convergence on compacta iff σ is not a polynomial.*

Proof. (sketch) Suppose σ is a polynomial of degree k. Since these form a closed set under linear combinations, $\mathcal{F}_\sigma$ will still only contain polynomials of degree at most k and thus cannot be dense in $C(\mathbb{R})$.

For the converse direction we will restrict ourselves to the case $\sigma \in C^\infty(\mathbb{R})$. The extension of the argument from $C^\infty(\mathbb{R})$ to $C(\mathbb{R})$ can be found in [11]. It is known (for instance as a non-trivial consequence of Baire's category theorem, cf. [6]) that for any $\sigma \in C^\infty(\mathbb{R})$ there is a point z such that $\sigma^{(k)}(z) \neq 0$ for all $k \in \mathbb{N}$. Since $\big[\sigma((\lambda + \delta)x + z) - \sigma(\lambda x + z)\big]/\delta$ represents a function in $\mathcal{F}_\sigma$ for all $\delta \neq 0$, we get that

$$\frac{d}{d\lambda}\sigma(\lambda x + z)\Big|_{\lambda=0} = x\, \sigma^{(1)}(z), \tag{2.9}$$

as a function of x, is contained in the closure of $\mathcal{F}_\sigma$. Similarly, by taking higher derivatives, we can argue that $x \mapsto x^k \sigma^{(k)}(z)$ is in the closure of $\mathcal{F}_\sigma$. Since all derivatives of σ are non-zero at z, all monomials and therefore all polynomials are contained in the closure of $\mathcal{F}_\sigma$. As these are dense in $C(\mathbb{R})$, by Weierstrass' theorem, so is $\mathcal{F}_\sigma$. □

Note that if σ is polynomial, then by the same reasoning no finite number of hidden layers will suffice to obtain an arbitrarily good approximation. If σ is non-polynomial, then depth of the network can be traded with width.

Lemma 2.2 (Approximation by exponentials)**.** *Let $K \subseteq \mathbb{R}^d$ be compact. Then $\mathcal{E} := \operatorname{span}\{f : K \to \mathbb{R} \mid f(x) = \exp \sum_{i=1}^d w_i x_i,\ w_i \in \mathbb{R}\}$ is dense in $\big(C(K), ||\cdot||_\infty\big)$.*

Proof. This is an immediate consequence of the Stone-Weierstrass theorem, which says that $\mathcal{E}$ is dense if (i) $\mathcal{E}$ forms an algebra (i.e., it is closed under multiplication and linear combination), (ii) $\mathcal{E}$ contains a non-zero constant function and (iii) for every distinct pair $x, y \in K$ there is an $f \in \mathcal{E}$ so that $f(x) \neq f(y)$. Here (i) holds by the property of the exponential and the construction of $\mathcal{E}$ as linear span, (ii) holds since $1 \in \mathcal{E}$ and (iii) holds since for $w := (x - y)$ we get $e^{w \cdot x} \neq e^{w \cdot y}$. □

Theorem 2.4: Approximation of multivariate functions

Let $d, d' \in \mathbb{N}$, $K \subseteq \mathbb{R}^d$ be compact and $\sigma : \mathbb{R} \to \mathbb{R}$ any activation function that is (i) continuous and non-polynomial or (ii) bounded and so that the limits $\lim_{z \to \pm\infty} \sigma(z)$ exist in $\mathbb{R}$ and differ from each other. Then the set of functions representable by a feedforward neural network with a single hidden layer of neurons with activation function σ is dense in the space of continuous functions $f : K \to \mathbb{R}^{d'}$ in the topology of uniform convergence.

Note: A norm inducing the topology of uniform convergence would be $||f|| := ||\big(||f_i||_\infty\big)_{i=1}^{d'}||'$ where $||\cdot||'$ is an arbitrary norm on $\mathbb{R}^{d'}$.

Proof. First note that it suffices to show the result for $d' = 1$ by considering the d' components in $\big(f_1(x), \ldots, f_{d'}(x)\big) = f(x)$ separately. If each of the f_i's can be approximated up to ϵ using m neurons in the hidden layer, then the same order of approximation is obtained for f with md' neurons just by stacking the d' hidden layers on top of each other.

In order to prove the statement for the case $f : K \to \mathbb{R}$ we first approximate f by exponentials and then the exponentials by linear combinations of activation functions. According to Lemma 2.2 for every $\epsilon > 0$ there is a $k \in \mathbb{N}$, a set of vectors $v_1, \ldots, v_k \in \mathbb{R}^d$ and signs $s \in \{-1, 1\}^k$ so that $g : \mathbb{R}^d \to \mathbb{R}$, $g(x) := \sum_{i=1}^k s_i e^{v_i \cdot x}$ satisfies $||f - g||_\infty \leq \epsilon/2$.

Define $K_1 := \bigcup_{i=1}^k \{v_i \cdot x \mid x \in K\}$ and note that $K_1 \subseteq \mathbb{R}$ is compact. Following Thm.2.3 and Prop.2.1 there is an $l \in \mathbb{N}$ and a set of real numbers

a_j, w_j, b_j so that $\sup_{y\in K_1} \left|e^y - \sum_{j=1}^{l} a_j\sigma(w_j y - b_j)\right| \leq \epsilon/(2k)$. Combining the two approximations we obtain

$$\begin{aligned} & \left\| f - \sum_{j=1}^{l}\sum_{i=1}^{k} s_i a_j \sigma(w_j\ v_i\cdot x - b_j)\right\|_\infty \\ \leq\ & \left\| f - \sum_{i=1}^{k} s_i e^{v_i\cdot x}\right\|_\infty + \sum_{i=1}^{k}\left\| e^y - \sum_{j=1}^{l} a_j\sigma(w_j y - b_j)\right\|_\infty \leq \epsilon, \end{aligned}$$

where the sup-norms are understood as $\sup_{x\in K}$ and $\sup_{y\in K_1}$, respectively. □

Let us finally make a remark, primarily of historical interest, concerning the relation of the above discussion to Kolmogorov's solution of Hilbert's 13th problem. Hilbert conjectured that a solution of the general equation of degree seven cannot be expressed as a finite superposition of continuous functions of two variables. In 1957 Kolmogorov and his student Arnold disproved this conjecture by showing that every continuous multivariate function can even be represented as a finite superposition of continuous functions of only one variable. This eventually led to the following:

Proposition 2.3 (Kolmogorov's superposition theorem)**.** *For every $n \in \mathbb{N}$ there exist functions $\varphi_j \in C\big([0,1]\big)$, $j = 0,\ldots,2n$ and constants $\lambda_k \in \mathbb{R}_+$, $k = 1,\ldots,n$ such that for every continuous function $f : [0,1]^n \to \mathbb{R}$ there exists $\phi \in C\big([0,1]\big)$ so that*

$$f(x_1,\ldots,x_n) = \sum_{j=0}^{2n} \phi\left(\sum_{k=1}^{n} \lambda_k \varphi_j(x_k)\right). \tag{2.10}$$

This theorem can be extended in various directions. First, one can restrict to increasing continuous functions φ_j and even show that the set of $2n+1$ tuples of such functions that fulfill the proposition is 'fat', in the sense of being of second Baire category. Moreover, one can show that there is a single continuous function φ in terms of which the φ_j's can be expressed as $\varphi_j(x) = c\varphi(aj+x)+bj$ with constants a, b, c.

From the point view of neural networks, Eq.(2.10) can be interpreted as a feedforward network with two hidden layers where the first hidden layer contains $n(2n+1)$ neurons, which use the φ_j's as activation functions, the second hidden layer contains $2n+1$ neurons with linear activation function $\sigma(z) = z$ and the output neuron uses ϕ as activation function. Hence, Eq.(2.10) provides an exact representation using only finitely many neurons, but at the cost of having an activation function, namely ϕ, that depends on f.

2.4 VC dimension of neural networks

Theorem 2.5: VC-dimension for feedforward networks

For arbitrary $n_0, \omega \in \mathbb{N}$ fix an architecture of a multilayerd feedforward neural network with n_0 inputs, a single output and ω weights (including threshold values). Denote by $\mathcal{F}$ the set of all functions $f : \mathbb{R}^{n_0} \to \{-1, 1\}$ that can be implemented by any feedworward network with this architecture when using $\sigma(z) = \mathrm{sgn}(z)$ as activation function. Then

$$\mathrm{VCdim}(\mathcal{F}) \leq 2\omega \log_2(e\omega). \tag{2.11}$$

Note: ω equals the number of edges in the graph that represents the network if we add to every neuron an additional edge that corresponds to the constant input related to the threshold value. Not surprisingly, the bound in Eq.(2.11) also holds (via the same argument) if $\sigma(z) = \mathbb{1}_{z \geq 0}$ and functions into $[0, 1]$ are considered.

Proof. Suppose the considered network has depth m and let n_i be the number of nodes (i.e., neurons or inputs) in the i'th layer with $i = 0, \ldots, m$. Then n_0 is the number of inputs and $n_m = 1$. We can decompose every function f that the considered architecture implements into functions $f_i : \mathbb{R}^{n_{i-1}} \to \{-1, 1\}^{n_i}$ that represent the mappings corresponding to the individual layers. Then $f = f_m \circ \cdots \circ f_1$. Furthermore, we can breakdown every $f_i(x) = \big(f_{i,1}(x), \ldots, f_{i,n_i}(x)\big)$ into its n_i components, each of which describes the action of a single neuron. Denote by $\omega_{i,j}$ the number of free parameters in $f_{i,j}$, i.e., the number of weights (including the threshold value) of the corresponding neuron. Then $\omega = \sum_{i=1}^{m} \sum_{j=1}^{n_i} \omega_{i,j}$ and we can bound the growth function of $\mathcal{F}$ via

$$\begin{aligned} \Gamma(n) &\leq \prod_{i=1}^{m} \Gamma_{f_i}(n) \leq \prod_{i=1}^{m} \prod_{j=1}^{n_i} \Gamma_{f_{i,j}}(n) \\ &\leq \prod_{i=1}^{m} \prod_{j=1}^{n_i} \left(\frac{en}{\omega_{i,j}}\right)^{\omega_{i,j}} \leq (en)^{\omega}. \end{aligned} \tag{2.12}$$

Here, the first inequality is an application of the composition property of the growth function shown in Lemma 1.4. Γ_{f_i} and $\Gamma_{f_{i,j}}$ denote the growth functions of the function classes corresponding to the i'th layer and the j'th neuron in the i'th layer, respectively. The step from the first to the second line in Eq.(2.12) exploits that by Thm.1.7 the VC-dimension of a single neuron is equal to the number of weights $\omega_{i,j}$, which then leads to an upper bound to the growth function following Thm.1.6. Finally, the last inequality simply uses $1/\omega_{i,j} \leq 1$.

From Eq.(2.12) we obtain that $\mathrm{VCdim}(\mathcal{F}) \leq D$, if $2^D \geq (eD)^{\omega}$. This is satisfied by $D = 2\omega \log_2(e\omega)$ for all $\omega > 1$. For $\omega = 1$, however, Eq.(2.11) holds as well since the VC-dimension in this case is at most 1. □

In Thm.2.1 we saw that an arbitrary Boolean function on d inputs can be represented by a neural network with 2^d neurons. As an implication of the above

theorem on the VC-dimension of feedforward neural networks we can now show that an exponential number of neurons is indeed necessary.

Corollary 2.4. *For any $d \in \mathbb{N}$ consider feedforward neural networks with d inputs, a single output and activation function $\sigma(z) = \mathbb{1}_{z\geq 0}$. Within this setting the number of neurons $\nu(d)$ of the smallest architecture that is capable of representing any Boolean function $f : \{0,1\}^d \to \{0,1\}$ satisfies $\nu(d) + d \geq 2^{(d-2)/3}$.*

Proof. The VC-dimension corresponding to the smallest such architecture has to be at least the VC-dimension of the class of Boolean functions, which is 2^d. On the other hand, it is at most $2\omega \log_2(e\omega)$ by Thm.2.5. If we let $G = (V, E)$ be the underlying graph and use that $|E| \leq |V|^2/2$ and thus $\omega \leq |E| + |V| \leq |V|^2$, we can estimate

$$2^d \leq 2\omega \log_2(e\omega) \leq 2|V|^2 \log_2\left(e|V|^2\right) \leq 4|V|^3.$$

With $\nu(d) + d = |V|$ we arrive at the desired result. □

Consider the following family of activation functions:

$$\sigma_c(z) := \frac{1}{1+e^{-z}} + cz^3 e^{-z^2} \sin(z), \quad c \geq 0. \tag{2.13}$$

The members of this family have many of the properties of the standard logistic sigmoid function, which is given by σ_0: the functions are analytic, satisfy $\lim_{z\to\infty} \sigma_c(z) = 1$, $\lim_{z\to-\infty} \sigma_c(z) = 0$ and for sufficiently small but not necessarily vanishing c we have that the second derivative $\sigma''(z)$ is strictly positive for $z < 0$ and strictly negative for $z > 0$. That is, the function is convex/concave in the respective regions.

Proposition 2.5 (Neural network with infinite VC-dimension)**.** *Consider feedforward networks with one input, a single output neuron whose activation function is $z \mapsto \operatorname{sgn}(z)$ and a single hidden layer with two neurons using σ_c with $c > 0$ as activation function. The class of functions $f : \mathbb{R} \to \{-1, 1\}$ representable by such networks has infinite VC-dimension.*

Proof. From Exp.1.4 we know that $\mathcal{F} := \{f : \mathbb{R}_+ \to \mathbb{R} \mid \exists \alpha \in \mathbb{R} : f(x) = \operatorname{sgn}\sin(\alpha x)\}$ has infinite VC-dimension. So the proposition follows by showing that $\mathcal{F}$ is contained in the function class representable by the considered networks. To this end, we choose the weights and threshold such that

$$\begin{aligned} f(x) &= \operatorname{sgn}\left[\sigma_c(\alpha x) + \sigma_c(-\alpha x) - 1\right] \\ &= \operatorname{sgn}\left[2c(\alpha x)^3 e^{-\alpha^2 x^2} \sin(\alpha x)\right] = \operatorname{sgn}\sin(\alpha x). \end{aligned}$$

□

2.5 Rademacher complexity of neural networks

... to be written ...

2.6 Training neural networks via gradient descent

... to be written ...

NP-hardness of empirical risk minimization Consider an arbitrary graph $G = (V, E)$ whose vertices are numbered so that $V = \{1, \dots, d\}$. Assign a set $S_G \in \left\{\{0,1\}^{|V|} \times \{0,1\}\right\}^n$ with $n := |V| + |E| + 1$ to the graph in the following way: denoting by $e_i \in \{0,1\}^{|V|}$ the unit vector whose i'th component is equal to one, we set $S = \{(e_i, 0), (e_i + e_j, 1), (0,1)\}_{i \in V, (i,j) \in E}$.

Recall that G is called 3-*colorable* iff there exists a map $\chi : V \to \{1,2,3\}$ with the property that $(i,j) \in E \Rightarrow \chi(i) \neq \chi(j)$. That is, there is an assignment of 'colors' to vertices such that no pair connected by an edge has the same color.

Proposition 2.6. *Consider feedforward neural networks with d inputs, a single hidden layer with three neurons and a single output neuron. Assume all activation functions are $\sigma(z) = \mathbb{1}_{z \geq 0}$ and that the output neuron has all weights and the threshold fixed so that it acts as $x \mapsto \sigma(\sum_{i=1}^3 (x_i - 1))$. Let $\mathcal{F}_d \subseteq \{0,1\}^{\mathbb{R}^d}$ be the function class that can be represented by such networks. Then for any graph G there is an $h \in \mathcal{F}_d$ which correctly classifies S_G iff G is 3-colorable.*

Proof. Note first that the output neuron is set up so that it fires iff all three hidden neurons do so. Assume G is 3-colorable via $\chi : V \to \{1,2,3\}$. The weight $w_{l,i}$ that connects the i'th input and the l'th hidden neuron is chosen so that $w_{l,i} = -1$ if $\chi(i) = l$ and $w_{l,i} = 1$ otherwise. With this we define $f \in \mathcal{F}_d$ so that

$$f(x) = 1 \;\Leftrightarrow\; \forall l \in \{1,2,3\} : \sum_k w_{l,k} x_k \geq -\frac{1}{2}.$$

Now we have to verify that f correctly classifies S_G. Clearly, $f(0) = 1$. It also holds that $f(e_i) = 0$ since if $\chi(i) = l$, then $w_{l,i} = -1$ so that $\sum_k w_{l,k}(e_i)_k = w_{l,i} \not\geq -1/2$. In order to verify $f(e_i + e_j) = 1$ for all $(i,j) \in E$, note that for any $l \in \{1,2,3\}$ we have $\chi(i) \neq l \vee \chi(j) \neq l$ since χ is a coloring. Therefore $\sum_k w_{l,k}(e_i + e_j)_k = w_{l,i} + w_{l,j}$ is larger than zero for all l.

Let us now show the converse implication and assume that there is an $f \in \mathcal{F}_d$ that correctly classifies S_G. Associating a half space H_l to each of the hidden Perceptrons we can express this assumption as $f^{-1}(\{1\}) = H_1 \cap H_2 \cap H_3 =: H$ where $0 \in H$, $\forall (i,j) \in E : \; e_i + e_j \in H$ and $\forall i \in V : e_i \notin H$. We define $\chi(i) := \min\{l | e_i \notin H_l\}$ and claim that this is a 3-coloring. First note that due to convexity of H and the fact that H contains the origin, we have $(e_i + e_j)/2 \in H$ for every edge $(i,j) \in E$. Suppose, aiming at a contradiction, that there would be an edge for which $\chi(i) = \chi(j) = l$. Then since $e_i, e_j \notin H_l$ this would, again by convexity, imply that $(e_i + e_j)/2 \notin H_l$ – a contradiction. □

2.7 Backpropagation

... to be completed ...

Consider a layered feedforward neural network whose layers will be labeled by an upper or lower index $l \in \{0, \ldots, m\}$. Here, the 0'th layer corresponds to the input and the m'th to the output. N_l will be the number of neurons in the l'th layer. By w^l_{jk} we will denote the weight that corresponds to the connection from the k'th neuron in layer $l-1$ to the j'th neuron in layer l. Similarly, b^l_j will be the threshold value of the j'th neuron in layer l. The vector x^l, whose components x^l_j are the outputs of the neurons of the l'th layer, can then be expressed in matrix/vector notation as $x^l = \sigma(w^l x^{l-1} + b^l)$, where the activation function σ is applied component-wise. We introduce a separate variable $z^l := w^l x^{l-1} + b^l$ to denote the output before application of the activation function.

Consider a function $f : \mathbb{R}^{N_m} \to \mathbb{R}$ that maps the output x^m to a real number—such as the loss function, which acts as $L(y, x^m) =: f(x^m)$ for a fixed pair (x^0, y) of the training data. By expanding x^m in terms of previous layers and the corresponding weights and threshold values, we may interpret f as a function of different kinds of variables. In particular, we will consider the mappings $(w, b) \mapsto f(x^m)$, $x^l \mapsto f(x^m)$ and $z^m \mapsto f(x^m)$. Abusing notation all these mappings will be denoted by f.

Our aim is to compute the partial derivatives of f w.r.t. all weights and threshold values. To this end, we introduce intermediate quantities $\delta^l_j := \frac{\partial f}{\partial z^l_j}$ in terms of which all the sought derivatives will be expressed by use of the chain rule. The latter is also central in computing the δ^l_j's themselves. Beginning with the output layer, we obtain

$$\delta^m_j = \sum_k \frac{\partial f}{\partial x^m_k} \frac{\partial x^m_k}{\partial z^m_j} = \sigma'(z^m_j)\, \frac{\partial f}{\partial x^m_j}, \tag{2.14}$$

where the summation runs over all neurons in the considered layer. Next, we show that δ^l can be expressed in terms of δ^{l+1}, so that all δ's can be computed by going layerwise backwards from the output layer:

$$\begin{aligned} \delta^l_j = \frac{\partial f}{\partial z^l_j} &= \sum_k \frac{\partial f}{\partial z^{l+1}_k} \frac{z^{l+1}_k}{\partial z^l_j} \\ &= \sum_k \delta^{l+1}_k\, w^{l+1}_{kj}\, \sigma'(z^l_j). \end{aligned} \tag{2.15}$$

Finally, we can express the sought derivatives in terms of the δ's:

$$\frac{\partial f}{\partial b^l_j} = \sum_k \frac{\partial f}{\partial z^l_k} \frac{\partial z^l_k}{\partial b^l_j} = \delta^l_j, \tag{2.16}$$

$$\frac{\partial f}{\partial w^l_{jk}} = \sum_i \frac{\partial f}{\partial z^l_i} \frac{\partial z^l_i}{\partial w^l_{jk}} = \delta^l_j\, x^{l-1}_k. \tag{2.17}$$

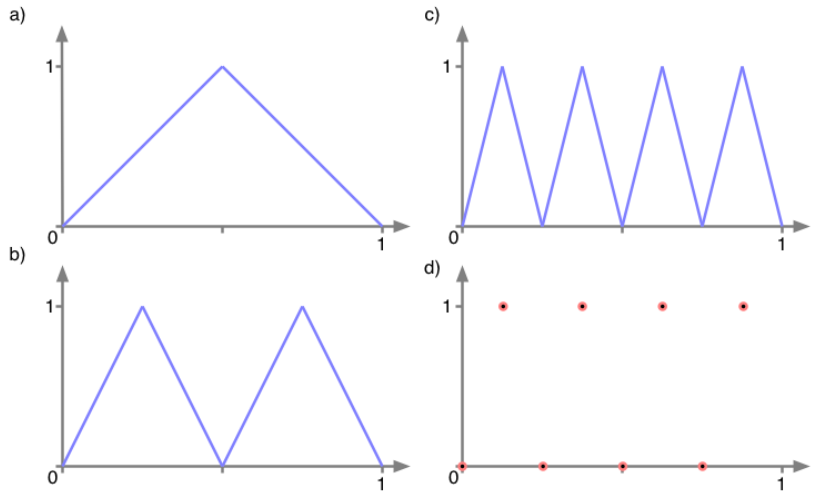


Figure 2.3: Fig. a) - c) show the graphs of g, g^2 and g^3. d) shows the set S for $k = 3$. By construction, it is part of the graph of g^3.

As a result we obtain that in order to compute the partial derivatives of f w.r.t. all weights and threshold values it suffices to run the network once forward (to obtain all x's and z's) and once 'backwards' (to obtain the δ's). This has to be contrasted with the naive approach, where for every individual partial derivative the network had to be evaluated twice already.

2.8 Deep neural nets

... to be written ...

Representation benefits of deep networks Define $\mathcal{F}(m, l) \subseteq \mathbb{R}^{\mathbb{R}}$ as the set of functions that can be represented by a feedforward neural network with l layers, m neurons within every hidden layer, a single neuron at the output and the rectified linear unit $\sigma_R(z) := \max\{0, z\}$ as activation function. In order to make an $f \in \mathcal{F}(m, l)$ into a classifier, define $\tilde{f}(x) := \mathbb{1}_{f(x) \geq 1/2}$ and let $\hat{R}(f) := \frac{1}{|S|} \sum_{(x,y) \in S} \mathbb{1}_{\tilde{f}(x) \neq y}$ be the corresponding empirical risk w.r.t. a training data set S.

Theorem 2.6: Exponential benefit of deep networks

Let $k \in \mathbb{N}$, $n = 2^k$ and $S := \big((x_i, y_i)\big)_{i=0}^{n-1}$ with $x_i := i/n$ and $y_i := i \bmod 2$.

1. There is an $h \in \mathcal{F}(2, 2k)$ for which $\hat{R}(h) = 0$.

2. If $m, l \in \mathbb{N}$ and $m \leq 2^{\frac{k-2}{l}-1}$, then $\hat{R}(f) \geq \frac{1}{6}$ holds for all $f \in \mathcal{F}(m, l)$.

Proof. 1. Define a function $g : \mathbb{R} \to \mathbb{R}$ as

$$g(x) := \begin{cases} 2x, & 0 \leq x \leq 1/2, \\ 2(1-x), & 1/2 < x \leq 1, \\ 0, & \text{otherwise.} \end{cases} \tag{2.18}$$

Since this can be written as $g(x) = \sigma_R\big(2\sigma_R(x) - 4\sigma_R(x - 1/2)\big)$ we have $g \in \mathcal{F}(2, 2)$. If we compose the function k-times with itself, the graph of $h(x) := g^k(x)$ is a saw-tooth with 2^{k-1} 'teeth' (see Fig.2.3). By construction, $h \in \mathcal{F}(2, 2k)$ and $h(x_i) = y_i$ for all $i = 1, \ldots n$. Hence, $\hat{R}(h) = 0$.

2. Every $f \in \mathcal{F}(m, l)$ is piecewise affine with at most $(2m)^l$ pieces. This is a consequence of the following simple fact: suppose f_1 and f_2 are piecewise affine with t_1 and t_2 pieces, respectively. Then $f_1 + f_2$ and $f_1 \circ f_2$ are again piecewise affine with at most $t_1 + t_2$ and $t_1 t_2$ pieces.

With at most $t = (2m)^l$ affine and thus monotone pieces, the graph of a function $f \in \mathcal{F}(m, l)$ crosses $1/2$ at most $2t - 1$ times: not more than once inside every interval and possibly once from one interval to the next. Therefore, $\tilde{f}$ is piecewise constant with at most $2t$ intervals with values 0 or 1. Let us now consider how the n points, whose values alternate between zero and one, can be distributed over these $2t$ intervals. Clearly, at most $2t$ points can be in intervals that contain no more than one point. The other $n - 2t$ points have to be in intervals that contain more than one point. At least one third of these points are thus misclassified so that the empirical risk can be bounded from below as

$$\hat{R}(f) \geq \frac{n - 2t}{3n} \geq \frac{1}{3} - \frac{2}{3n}(2m)^l,$$

which is at least $1/6$ if $m \leq 2^{(k-2)/l-1}$. □

Convolutional neural nets ... to be written ...

Chapter 3

Support Vector Machines

3.1 Linear maximal margin separators

Separable case. Consider a real Hilbert space $\mathcal{H}$ and a training data set $S = \big((x_i, y_i)_{i=1}^n\big) \in \big(\mathcal{H} \times \{-1, 1\}\big)^n$. Suppose the two subsets of points corresponding to the labels ± 1 can be separated by a hyperplane H. That is, there are $w \in \mathcal{H}$ and $b \in \mathbb{R}$ that characterize the hyperplane via $H = \{x \in \mathcal{H} \mid \langle w, x\rangle + b = 0\}$ so that $\forall i : \operatorname{sgn}\big(\langle w, x_i\rangle + b\big) = y_i$. If there is no point exactly on the hyperplane this is equivalent to

$$y_i\big(\langle w, x_i\rangle + b\big) > 0 \quad \forall i. \tag{3.1}$$

The separating hyperplane is not unique and the question arises, which separating hyperplane to choose. The standard approach in the SVM framework is to choose the one that maximizes the distance to the closest points on both sides. In order to formalize this, we need the following Lemma.

Lemma 3.1 (Distance to a hyperplane)**.** *Let $\mathcal{H}$ be a Hilbert space and $H := \{z \in \mathcal{H} \mid \langle z, w\rangle + b = 0\}$ a hyperplane defined by $w \in \mathcal{H}$ and $b \in \mathbb{R}$. The distance of a point $x \in \mathcal{H}$ to H is given by*

$$d(x, H) := \inf_{z \in H} ||x - z|| = \frac{|\langle x, w\rangle + b|}{||w||}. \tag{3.2}$$

Proof. Let us first determine the distance of an arbitrary hyperplane to the origin: since $\inf_{z \in H} ||z||$ is attained for $z = -bw/||w||^2$ we get that $d(0, H) = |b|/||w||$. Using that translations are isometries, we can rewrite $d(x, H) = d(0, H - x)$ and apply the previous observation to the hyperplane $H - x = \{z | \langle z, w\rangle + b' = 0\}$ with $b' := \langle x, w\rangle + b$. □

Using Lemma 3.1 and Eq.(3.1) we can write the distance between a separating hyperplane and the closest point in S as

$$\rho := \min_i d(x_i, H) = \frac{\min_i y_i\big(\langle w, x_i\rangle + b\big)}{||w||}. \tag{3.3}$$

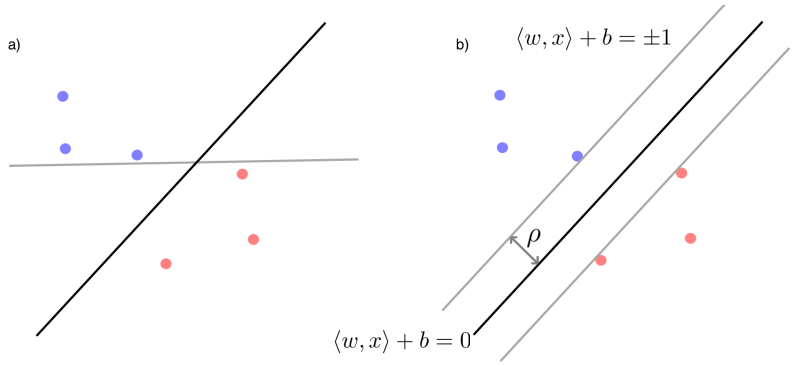


Figure 3.1: a) Red and blue points are separated by both hyperplanes. b) The black hyperplane is the one that maximizes the *margin* ρ. If the two margin hyperplanes are characterized by $\langle w, x\rangle + b = \pm 1$, then $\rho = 1/||w||$.

ρ is called the *margin* of the hyperplane w.r.t. S and the aim is now to determine the hyperplane that maximizes the margin. To this end, note that there is a scalar freedom in the characterization of the hyperplane: if we multiply both w and b by a positive scalar, then the hyperplane is still the same and also the margin does not change. We can now use this freedom to fix either the denominater in Eq.(3.3) or the enumerator and in this way obtain two different albeit equivalent constrained optimization problems. Constraining the denominator for instance leads to

$$\max_{(b,w)} \rho \;=\; \max_{(b,w):||w||\leq 1} \min_i \; y_i\big(\langle w, x_i\rangle + b\big).$$

Assuming that the sets of points that correspond to the two labels are not empty, a maximum is attained since the closed unit ball in a Hilbert space is weakly compact. So writing max instead of sup is indeed justified.

Alternatively, in order to obtain the hyperplane that maximizes the margin, we may use the mentioned scalar freedom to impose a constraint on the enumerator in Eq.(3.3) and minimize the denominator $||w||$ or, for later convenience, $||w||^2/2$, which leads to the same minimizer. That is, the maximal margin hyperplane is the one that achieves the minimum in

$$\min_{(b,w)} \; \frac{1}{2}||w||^2 \quad \text{s.t.} \quad \forall i: \; y_i\big(\langle w, x_i\rangle + b\big) \geq 1. \tag{3.4}$$

This is an optimization problem with strictly convex target function and affine inequality constraints. Due to strict convexity the minimum is unique. We further apply a standard tool from convex optimization:

Proposition 3.2 (Convex KKT)**.** *Let $\mathcal{H}$ be a real Hilbert space, $\{f_i : \mathcal{H} \to \mathbb{R}\}_{i=0}^n$ a set of continuously differentiable convex functions and assume that there is a $z \in \mathcal{H}$ for which $f_i(z) < 0$ holds for all $i = 1, \ldots, n$. Then for every $\tilde{z}$ that satisfies $f_i(\tilde{z}) \leq 0$ for all $i = 1, \ldots, n$ the following are equivalent:*

1. $f_0(\tilde{z}) = \min_{z\in\mathcal{H}}\{f_0(z) \mid f_i(z) \leq 0 \; \forall i = 1,\ldots,n\}$.

2. *There exist* $\lambda_i \leq 0$ *so that*

$$\nabla f_0(\tilde{z}) = \sum_{i=1}^{n} \lambda_i \nabla f_i(\tilde{z}) \text{ and} \tag{3.5}$$

$$\lambda_i f_i(\tilde{z}) = 0 \quad \forall i = 1,\ldots,n. \tag{3.6}$$

Applying this to the optimization problem in Eq.(3.4) leads to the following crucial insight: if $\tilde{w}$ corresponds to the maximal margin hyperplane, then Eq.(3.5) implies $\tilde{w} = \sum_{i=1}^{n} y_i\lambda_i x_i$. That is, the minimizing $\tilde{w}$ is a linear combination of the training data points x_i. In addition, Eq.(3.6), which in our case reads $\lambda_i\big[1 - y_i\big(\langle\tilde{w}, x_i\rangle + b\big)\big] = 0$, implies that only those x_i's contribute for which the i'th constraint is *active*. This means $y_i\big(\langle\tilde{w}, x_i\rangle + b\big) = 1$ so that the corresponding x_i is sitting on one of the two margin hyperplanes. These x_i's are called *support vectors*.

Non-separable case Now we drop the assumption that the data is exactly linearly separable. However, we still seek a predictor that is given in terms of a hyperplane and that in some sense still has maximal margin. The difference to the foregoing discussion is that we now allow for outliers that may either be on the wrong side of the hyperplane or inside the margin. In order to formalize this, one introduces *slack variables* $\xi_i \geq 0$ that measure the extent to which the i'th constraint is violated. In addition, one penalizes these violations in the object function. This leads to the optimization problem

$$\begin{aligned} &\min_{(b,w,\xi)} \; \frac{\lambda}{2}||w||^2 + \frac{1}{n}\sum_{i=1}^{n}\xi_i \\ &\text{s.t.} \;\; y_i\big(\langle w, x_i\rangle + b\big) \geq 1 - \xi_i \quad \wedge \quad \xi_i \geq 0 \;\; \forall i = 1,\ldots n, \end{aligned} \tag{3.7}$$

where $\lambda > 0$ is a free parameter that can be used to adjust the strength of the penalty. There is some arbitrariness in how one penalizes large ξ. In Eq.(3.7) we have essentially chosen the l_1-norm of ξ. Another common choice would be the l_2-norm.

The optimization problem in Eq.(3.7) can be written as ERM problem w.r.t. the so-called *hinge loss* $L_{hinge} : \mathbb{R}\times\mathbb{R} \to \mathbb{R}_+$ that is defined as

$$L_{hinge}(y, y') := \max\{0, 1 - yy'\}.$$

The hinge loss provides an upper bound on the usually taken loss function for binary classification in the sense that if $y \in \{-1, 1\}$, then $\mathbb{1}_{y\neq\mathrm{sgn(h(x))}} \leq L_{hinge}(y, h(x))$. Other noticeable properties are that $y' \mapsto L_{hinge}(y, y')$ is convex and $w \mapsto L_{hinge}(y, \langle w, x\rangle + b)$ is $||x||$-Lipschitz.

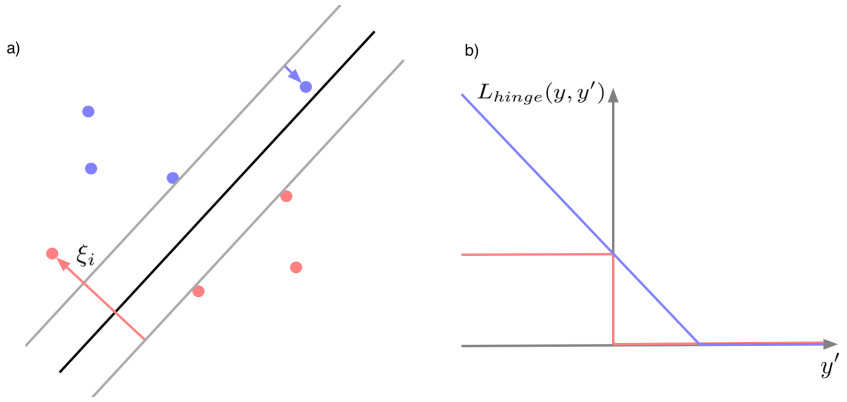


Figure 3.2: a) Outliers (points that are either inside the margin corridor, or on the wrong side) are penalized using *slack variable* ξ_i. b) The *hinge loss* (blue), plotted for the case $y = 1$, is a convex upper bound for the 0-1-loss (red) that is usually used for binary classification.

The optimization problem in Eq.(3.7) can now be written as

$$\begin{aligned} &\min_{(b,w)} \frac{\lambda}{2}||w||^2 + \frac{1}{n}\sum_{i=1}^{n} \max\left\{0, 1 - y_i\big(\langle w, x_i\rangle + b\big)\right\} \\ &= \min_{(b,w)} \frac{\lambda}{2}||w||^2 + \hat{R}_{hinge}(h), \end{aligned} \tag{3.8}$$

where $h(x) := \langle w, x\rangle + b$. Note that Eq.(3.8) is a regularized ERM problem without additional constraints.

Theorem 3.1: Representer theorem

Let $\mathcal{H}$ be a Hilbert space, $g : \mathbb{R} \to \mathbb{R}$ non-decreasing, $f : \mathbb{R}^n \to \mathbb{R}$, $\{x_1, \ldots, x_n\} \subseteq \mathcal{H}$, $\mathcal{H}_x := \text{span}\{x_i\}_{i=1}^n$ and $F : \mathcal{H} \to \mathbb{R}$, $F(w) := g(||w||) + f(\langle w, x_1\rangle, \ldots, \langle w, x_n\rangle)$. Then

$$\inf_{w\in\mathcal{H}} F(w) = \inf_{w\in\mathcal{H}_x} F(w) \tag{3.9}$$

and if g is strictly increasing, then every minimizer of the l.h.s. of Eq.(3.9) is an element of $\mathcal{H}_x$.

Proof. We use that $\mathcal{H} = \mathcal{H}_x \oplus \mathcal{H}_x^\perp$ and that every $w \in \mathcal{H}$ admits a corresponding decomposition of the form $w = w_x + v$ where $w_x \in \mathcal{H}_x$ and $v \in \mathcal{H}_x^\perp$. Then $\langle w, x_i\rangle = \langle w_x, x_i\rangle$ holds for all i and from Pythagoras we obtain

$$g(||w||) = g\Big(\sqrt{||w_x||^2 + ||v||^2}\Big) \geq g(||w_x||).$$

Here, strict inequality holds if g is strictly increasing and $w \neq w_x$. Hence, the claims follow by replacing w by w_x in the argument of F. □

3.2 Positive semidefinite kernels

Definition 3.3 (PSD kernel)**.** *Let $\mathbb{K} \in \{\mathbb{R}, \mathbb{C}\}$ and X be an arbitrary set. A map $K : X \times X \to \mathbb{K}$ is called* positive semidefinite kernel *(PSD kernel) iff for all $n \in \mathbb{N}$ and all $x \in X^n$ the $n \times n$ matrix G with entries $G_{ij} := K(x_i, x_j)$ is positive semidefinite.*

The terminology varies considerably throughout the literature. PSD kernels also run under the names positive definite kernels, positive definite symmetric kernels, kernel functions or just kernels. Recall that a matrix G is positive semidefinite iff G is hermitian, i.e., $G_{ij} = \bar{G}_{ji}$, and G has only non-negative eigenvalues. The latter condition can be replaced with

$$\forall \alpha \in \mathbb{K}^n : \sum_{i,j=1}^{n} \bar{\alpha}_i \alpha_j G_{ij} \geq 0. \tag{3.10}$$

Theorem 3.2: PSD kernels and feature maps

Let X be any set, $\mathbb{K} \in \{\mathbb{R}, \mathbb{C}\}$ and $K : X \times X \to \mathbb{K}$.

1. K is a PSD kernel, if there is an inner product space $\mathcal{H}$ and a map $\phi : \mathcal{H} \to \mathbb{K}$ so that

$$K(x, y) = \langle \phi(x), \phi(y) \rangle \qquad \forall x, y \in X. \tag{3.11}$$

2. Conversely, if K is a PSD kernel, then there exists a Hilbert space $\mathcal{H}$ and a map $\phi : \mathcal{H} \to \mathbb{K}$ so that Eq.(3.11) holds.

Note: the map ϕ is often called *feature map* and the inner product space $\mathcal{H}$ the *feature space*.

Proof. 1. If K is of the form in Eq.(3.11), then for all $\alpha \in \mathbb{K}^n$ and $x \in X^n$ we have $\sum_{i,j=1}^{n} \bar{\alpha}_i \alpha_j \langle \phi(x_j), \phi(x_i) \rangle = \langle \Phi, \Phi \rangle \geq 0$ where $\Phi := \sum_{i=1}^{n} \alpha_i \phi(x_i)$. Hermiticity of the respective matrix follows from hermiticity of the inner product.

2. Assume K to be a PSD kernel and define

$$\mathcal{H}_0 := \operatorname{span} \left\{ k_x : X \to \mathbb{K} \mid \exists x \in X : k_x(y) = K(x, y) \right\} \tag{3.12}$$

the space of all finite $\mathbb{K}$-linear combination of functions of the form $y \mapsto K(x, y)$. We aim at equipping this space with an inner product. For two arbitrary elements of $\mathcal{H}_0$ given by $f(y) := \sum_i \alpha_i K(x_i, y)$ and $g(y) := \sum_j \beta_j K(x_j, y)$ define

$$\begin{aligned} \langle f, g \rangle &:= \sum_{i,j} \alpha_i \bar{\beta}_j K(x_i, x_j) \\ &= \sum_i \alpha_i \overline{g(x_i)} = \sum_j \bar{\beta}_j f(x_j), \end{aligned}$$

where the second line shows that the definition is independent of the particular decomposition of f or g. So $\langle \cdot, \cdot \rangle$ is a well defined hermitian sesquilinear form on $\mathcal{H}_0$. Moreover, since K is a PSD kernel, we have $\langle g, g \rangle \geq 0$ for all $g \in \mathcal{H}_0$. Hence, the Cauchy Schwarz inequality holds. Applying it to

$$f(y) = \sum_i \alpha_i K(x_i, y) = \langle f, k_y \rangle, \tag{3.13}$$

we obtain $|f(y)|^2 = |\langle f, k_y \rangle|^2 \leq \langle k_y, k_y \rangle \langle f, f \rangle$. This shows that $\langle f, f \rangle = 0$ implies $f = 0$ and thus $\langle \cdot, \cdot \rangle$ is indeed an inner product. Note that if we apply Eq.(3.13) to $f = k_x$, we obtain

$$k_x(y) = K(x, y) = \langle k_x, k_y \rangle. \tag{3.14}$$

So if we denote by $\mathcal{H}$ the completion of the inner product space $\mathcal{H}_0$ and define $\phi : X \to \mathcal{H}$ so that $\phi(x)$ is the isometric embedding of k_x into $\mathcal{H}$, then Eq.(3.14) implies $K(x, y) = \langle \phi(x), \phi(y) \rangle$. □

Proposition 3.4 (Building new PSD kernels)**.** *Let $\mathbb{K} \in \{\mathbb{R}, \mathbb{C}\}$, $K_1, K_2, \ldots$ PSD kernels on a set X and $f \in \mathbb{K}^X$. Then*

1. *$K(x, y) := f(x)\overline{f(y)}$ is a PSD kernel.*
2. *$K(x, y) := \lambda K_1(x, y)$ is a PSD kernel for all $\lambda \geq 0$.*
3. *$K(x, y) := K_1(x, y) + K_2(x, y)$ is a PSD kernel.*
4. *$K(x, y) := \lim_{n \to \infty} K_n(x, y)$ is a PSD kernel, if the limits exist in $\mathbb{K}$.*
5. *$K(x, y) := K_1(x, y) K_2(x, y)$ is a PSD kernel.*

Proof. In all cases hermiticity is rather obvious, so we only have a look at positive semidefiniteness. 1. K is PSD since $\sum_{i=1}^n \alpha_i \bar{\alpha}_j f(x_i)\overline{f(x_j)} = \left| \sum_{i=1}^n \alpha_i f(x_i) \right|^2$ is always positive. 2. and 3. are elementary consequences of the definition. 4. is implied by the closedness of the set of PSD matrices, or more explicitly by positivity of $\sum_{i=1}^m \alpha_i \bar{\alpha}_j K(x, y) = \lim_{n \to \infty} \sum_{i=1}^m \alpha_i \bar{\alpha}_j K_n(x, y)$ as a limit of positive numbers. 5. follows from the fact the set of PSD matrices is closed under taking element wise products (called *Schur products* or *Hadamard products*). □

With these tools at hand, many kernels can easily be shown to be PSD. Some of the most common examples are:

Example 3.1 (Polynomial kernels). On $X = \mathbb{R}^d$ any polynomial in $\langle x, y \rangle$ with non-negative coefficients is a PSD kernel as a consequence of 2., 3. and 5. in Prop. 3.4 together with the fact that $(x, y) \to \langle x, y \rangle$ is (the paradigm of) a PSD kernel. In particular, $K(x, y) := (1 + \langle x, y \rangle)^2$ is a PSD kernel. On $\mathbb{R}^2$ this can be obtained from the feature map $\phi(x) : \mathbb{R}^2 \to \mathbb{R}^6$, $\phi(x) := (x_1^2, x_2^2, \sqrt{2}x_1x_2, \sqrt{2}x_1, \sqrt{2}x_2, 1)$. Like in this example, all polynomial kernels have corresponding finite dimensional feature spaces.

Example 3.2 (Exponential kernels). For any $\gamma > 0$, $K(x, y) := \exp[\gamma \langle x, y \rangle]$ is a PSD kernel on $X = \mathbb{R}^d$ since it is a limit of polynomial kernels so that 4. in Prop. 3.4 applies.

Example 3.3 (Gaussian kernels). The Gaussian kernel $K(x,y) := \exp\big[-\frac{\gamma}{2}||x-y||^2\big]$ with the Euclidean norm is a PSD kernel on $X = \mathbb{R}^d$ for all $\gamma > 0$. To see this write

$$\exp\big[-\frac{\gamma}{2}||x-y||^2\big] = \underbrace{\exp[-\gamma||x||^2/2]\ \exp[-\gamma||y||^2/2]}_{f(x)\overline{f(y)}} \exp[\gamma\langle x,y\rangle]$$

and apply 1. and 5. of Prop. 3.4.

Example 3.4 (Binomial kernels). On $X := \{x \in \mathbb{R}^d \mid ||x||_2 < 1\}$ $K(x,y) := \big(1-\langle x,y\rangle\big)^{-p}$ is a PSD kernel for any $p > 0$. This follows again from the previous proposition by noting that for $t \in (-1,1)$ the binomial series $(1-t)^{-p} = \sum_{n=0}^{\infty}(-1)^n\binom{-p}{n}t^n$ has positive coefficients $(-1)^n\binom{-p}{n} = (-1)^n\prod_{i=1}^{n}(1-p-i)/i$.

We will see in Sec.3.4 that, whereas polynomial kernels have finite dimensional feature spaces, exponential, Gaussian and binomial kernels require infinite dimensional feature space.

3.3 Reproducing kernel Hilbert spaces

For a given PSD kernel, the corresponding feature map and feature space are not unique. However, there is a canonical choice for the feature space, a so-called *reproducing kernel Hilbert space*.

Definition 3.5 (Reproducing kernel Hilbert space)**.** *Let X be a set, $\mathbb{K} \in \{\mathbb{R},\mathbb{C}\}$ and $\mathcal{H} \subseteq \mathbb{K}^X$ a $\mathbb{K}$-Hilbert space of functions on X with addition $(f+g)(x) := f(x)+g(x)$ and multiplication $(\lambda f)(x) := \lambda f(x)$. $\mathcal{H}$ is called a* reproducing kernel Hilbert space *(RKHS) on X iff for all $x \in \mathcal{H}$ the linear functional $\delta_x : \mathcal{H} \to \mathbb{K}$, $\delta_x(f) := f(x)$ is bounded (i.e., $\sup_{f\in\mathcal{H}\setminus\{0\}} |f(x)|/||f|| < \infty$).*

Note: Since δ_x is linear, boundedness is equivalent to continuity. That is, the defining property of a RKHS is that evaluation of its functions at arbitrary points is continuous w.r.t. varying the function.

Example 3.5. If X is countable, then $l_2(X) := \{f \in \mathbb{K}^X \mid \sum_{x\in X}|f(x)|^2 < \infty\}$ equipped with $\langle f,g\rangle := \sum_{x\in X} f(x)\overline{g(x)}$ is a RKHS since for all $x \in X$ we have $|f(x)| \le \big(\sum_{y\in X}|f(y)|^2\big)^{1/2} = ||f||$. Hence, δ_x is bounded.

Example 3.6. $L_2([0,1])$ is not a RKHS. Since its elements are equivalence classes of functions that differ on sets of measure zero, $f(x)$ is not defined. Even if we restrict to the subspace of continuous functions, where $f(x)$ is defined, its magnitude is not bounded by imposing $||f|| \le 1$. So δ_x is not bounded.

Since $l_2(\mathbb{N})$ and $L_2([0,1])$ are isomorphic, these two examples show that Hilbert space isomorphisms do not necessarily preserve the RKHS property.

A crucial consequence of the continuity of δ_x in any RKHS is that one can invoke the Riesz representation theorem. This states that every continuous linear functional on a Hilbert space can be represented as inner product with a unique vector. In particular, if $\mathcal{H}$ is a RKHS, then for every $x \in \mathcal{H}$ there is a

$k_x \in \mathcal{H}$ so that $f(x) = \langle f, k_x \rangle$ for all $f \in \mathcal{H}$. Since inner products are always continuous, this can be regarded as equivalent characterization of a RKHS. As k_x is an element of $\mathcal{H}$ and therefore a function on X, we can define $K : X \times X \to \mathbb{K}$, $K(x, y) := k_x(y)$. K is called the *reproducing kernel* of the RKHS $\mathcal{H}$. Using that $k_x(y)$ can itself be expressed in terms of an inner product with some element k_y, we obtain

$$K(x, y) = \langle k_x, k_y \rangle. \tag{3.15}$$

Before we relate reproducing kernel Hilbert spaces to PSD kernels, let us mention some elementary properties:

Proposition 3.6. *Let $\mathcal{H} \subseteq \mathbb{K}^X$ be a RKHS with reproducing kernel K and $k_x(y) = K(x, y)$. Let $f, f_n \in \mathcal{H}$ and $\delta_x(f) := f(x)$ for $x \in X$, $f \in \mathcal{H}$. Then*

1. *For all $x \in \mathcal{H}$ we have $||\delta_x||^2 = K(x, x)$.*
2. *$\lim_{n\to\infty} ||f_n - f|| = 0 \;\Rightarrow\; \forall x \in X : \lim_{n\to\infty} f_n(x) = f(x)$.*
3. *$\mathrm{span}\{k_x \mid x \in X\}$ is dense in $\mathcal{H}$.*

Proof. 1. follows with $f(x) = \langle f, k_x \rangle$ from

$$||\delta_x||^2 = \sup_{f \in \mathcal{H}\setminus\{0\}} \frac{|\langle k_x, f\rangle|^2}{||f||^2} = ||k_x||^2 = \langle k_x, k_x \rangle = K(x, x), \tag{3.16}$$

where the second equality is the one of the Cauchy Schwarz inequality. Similarly, also 2. is obtained from Cauchy Schwarz by noting that

$$\big|f_n(x) - f(x)\big| = \big|\langle k_x, f_n - f\rangle\big| \leq ||k_x||\, ||f_n - f|| \to 0.$$

For 3. it suffices to show that there is no non-zero element that is orthogonal to the considered span. Indeed, suppose $f \in \mathcal{H}$ is orthogonal to all k_x, then for all $x \in X$ we have that $0 = \langle f, k_x \rangle = f(x)$, which means $f = 0$. □

Theorem 3.3: RKHS and PSD kernels

1. If $\mathcal{H}$ is a RKHS on X, then its reproducing kernel $K : X \times X \to \mathbb{K}$ is a PSD kernel.
2. Conversely, if $K : X \times X \to \mathbb{K}$ is a PSD kernel, then there is a unique RKHS $\mathcal{H} \subseteq \mathbb{K}^X$ so that K is its reproducing kernel.

Proof. 1. If K is the reproducing kernel of a RKHS $\mathcal{H}$, then by Eq.(3.15) and the properties of the inner product:

$$\begin{aligned} \forall x, y \in X : \quad K(x, y) &= \langle k_x, k_y \rangle = \overline{\langle k_y, k_x \rangle} = \overline{K(y, x)} \quad \text{and} \\ \sum_{i,j=1}^{n} \alpha_i \bar{\alpha}_j K(x_i, x_j) &= \sum_{i,j=1}^{n} \alpha_i \bar{\alpha}_j \langle k_{x_i}, k_{x_j} \rangle = \Big\| \sum_{i=1}^{n} \alpha_i k_{x_i} \Big\|^2 \geq 0. \end{aligned}$$

2. (sketch) The construction of the sought RKHS is the one in the proof of Thm.3.2. Eqs.(3.13,3.14) show that K fulfills the requirement of a reproducing kernel on $\mathcal{H}_0$. A more careful consideration shows that the relevant properties are indeed preserved when going from $\mathcal{H}_0$ to its completion $\mathcal{H}$.

To address uniqueness suppose $\mathcal{H}_1$ and $\mathcal{H}_2$ are two RKHS with reproducing kernel K. Following 3. in Prop.3.6 the space $\mathcal{H}_0 = \text{span}\{k_x | x \in X\}$ is dense in both $\mathcal{H}_1$ and $\mathcal{H}_2$. Moreover, if $f \in \mathcal{H}_0$ with $f(x) = \sum_i \alpha_i k_{x_i}$, then $||f||_l^2 = \sum_{i,j} \alpha_i \bar{\alpha}_j K(x_i, x_j)$ for $l = 1, 2$. Hence, the norms $||\cdot||_1$ and $||\cdot||_2$ coincide on $\mathcal{H}_0$.

Suppose $f \in \mathcal{H}_1$. Then there are $f_n \in \mathcal{H}_0$ so that $||f_n - f||_1 \to 0$. As $(f_n)_{n\in\mathbb{N}}$ is Cauchy in $\mathcal{H}_1$ it is also Cauchy in $\mathcal{H}_2$ and therefore there exist a $g \in \mathcal{H}_2$ so that $||f_n - g||_2 \to 0$. According to 2. in Prop.3.6 we have $f(x) = \lim_{n\to\infty} f_n(x) = g(x)$ for all $x \in X$. Hence, $f = g \in \mathcal{H}_2$ and consequently $\mathcal{H}_1 = \mathcal{H}_2$. Since the norms, and by polarization also the inner products, coincide on a dense subspace, the do so on its completion. □

3.4 Universal and strictly positive kernels

Definition 3.7 (Universal kernels)**.** *A PSD kernel $K : X \times X \to \mathbb{K}$ on a metric space X is called* universal *iff for all $\epsilon > 0$, all compact subsets $\tilde{X} \subseteq X$ and every continuous function $f : X \to \mathbb{K}$ there exists $g \in \text{span}\{k_x : X \to \mathbb{K} \mid \exists x \in X : k_x(y) = K(x, y)\}$ so that*

$$\left|g(x) - f(x)\right| \leq \epsilon \quad \forall x \in \tilde{X}. \tag{3.17}$$

Note that if $\phi : X \to \mathcal{H}$ is a feature map corresponding to K, then Eq.(3.17) means that there exists a $w \in \mathcal{H}$ so that

$$\left|\langle w, \phi(x)\rangle - f(x)\right| \leq \epsilon \quad \forall x \in \tilde{X}. \tag{3.18}$$

Corollary 3.8 (Universal kernels separate all compact subsets)**.** *Let $\phi : X \to \mathcal{H}$ be a feature map of a universal PSD kernel on a metric space X. For any pair of disjoint compact subsets $A_+, A_- \subseteq X$ there exists a $w \in \mathcal{H}$ so that for all $x \in A_+ \cup A_-$:*

$$\text{sgn}\langle w, \phi(x)\rangle = \begin{cases} +1, & x \in A_+ \\ -1, & x \in A_- \end{cases} \tag{3.19}$$

Proof. As the distance between A_+ and A_- is non-zero, we can extend the function $A_+ \cup A_- \ni x \mapsto \mathbb{1}_{x\in A_+} - \mathbb{1}_{x\in A_-}$ to a continuous function f on X. By universality there exists a $w \in \mathcal{H}$ for each $\epsilon \in (0, 1)$ so that $|\langle w, \phi(x)\rangle - f(x)| \leq \epsilon$ for all $x \in A_+ \cup A_-$. Hence,

$$\langle w, \phi(x)\rangle \begin{cases} \geq 1 - \epsilon, & x \in A_+ \\ \leq \epsilon - 1, & x \in A_- \end{cases} \tag{3.20}$$

Note that in this case the sets are separated with margin $(1 - \epsilon)/||w||$. □

Theorem 3.4: Taylor criterion for universality

Let $f(z) := \sum_{n=0}^{\infty} a_n z^n$ be a power series with radius of convergence $r \in (0, \infty]$ and $X := \{x \in \mathbb{R}^d \mid ||x||_2 < \sqrt{r}\}$. If $a_n > 0$ for all n, then $K : X \times X \to \mathbb{R},\ K(x, y) := f(\langle x, y\rangle)$ is a universal PSD kernel.

Proof. First note that K is well defined since $|\langle x, y\rangle| \leq ||x||_2 ||y||_2 < r$. Using multinomial expansion we can write

$$\begin{aligned} K(x,y) &= \sum_{n=0}^{\infty} a_n \left(\sum_{k=1}^{d} x_k y_k\right)^n = \sum_{n=0}^{\infty} a_n \sum_{\substack{k_1+\cdots+k_d=n \\ k_1,\ldots,k_d \geq 0}} \frac{n!}{k_1!\cdots k_d!} \prod_{i=1}^{d} (x_i y_i)^{k_i} \\ &= \sum_{k_1,\ldots,k_d \geq 0} \underbrace{a_{k_1+\ldots+k_d} \frac{(k_1+\ldots+k_d)!}{k_1!\cdots k_d!}}_{=:c_k} \prod_{i=1}^{d} x_i^{k_i} \prod_{j=1}^{d} y_j^{k_j}. \end{aligned} \tag{3.21}$$

This enables us to introduce a feature map $\phi : X \to l_2(\mathbb{N}_0^d)$ as $\phi_k(x) := \sqrt{c_k} \prod_{i=1}^{d} x_i^{k_i}$ for $k \in \mathbb{N}_0^d$ so that $K(x, y) = \langle \phi(x), \phi(y)\rangle$. Since all a_n's are strictly positive, the same holds true for all c_k's. Consequently, $\text{span}\{\phi_k\}_{k \in \mathbb{N}_0^d}$ is the space of all polynomials and by the Stone-Weierstrass theorem dense in the set of continuous functions on compact domains. The claim then follows from the observation that every finite linear combination of functions of the form $x \mapsto \phi_k(x)$ can be regarded as an inner product $\langle w, \phi(x)\rangle$ for some vector w. Since the latter has only finitely many non-zero components, it is indeed an element of $l_2(\mathbb{N}_0^d)$. □

Corollary 3.9. *On $X = \mathbb{R}^d$ the following are universal PSD kernels:*

1. *Exponential kernel: $K(x, y) := \exp\big(\gamma\langle x, y\rangle\big)$, $\gamma > 0$.*
2. *Gaussian kernel: $K(x, y) := \exp\big(-\frac{\gamma}{2}||x - y||_2^2\big)$, $\gamma > 0$.*

Proof. Universality of the exponential kernel follows directly from Thm.3.4 with $a_n = \tau^n / n!$. This in turn can be used to prove universality of the Gaussian kernel: if $\phi : X \to \mathcal{H}$ is a feature map of the exponential kernel, then $\tilde{\phi} : x \mapsto \phi(x)/||\phi(x)||$ is a feature map of the Gaussian kernel. Now take any compact subset $\tilde{X} \subseteq X$ and define $c := \sup_{x \in \tilde{X}} ||\phi(x)||^{-1}$. By universality of the exponential kernel, for every continuous function $f : X \to \mathbb{R}$ there is a $w \in \mathcal{H}$ so that

$$\Big| f(x)||\phi(x)|| - \langle w, \phi(x)\rangle \Big| \leq \frac{\epsilon}{c} \quad \forall x \in \tilde{X}.$$

Dividing by $||\phi(x)||$ and taking the supremum over $x \in \tilde{X}$ on the resulting r.h.s. leads to $\big|f(x) - \langle w, \tilde{\phi}(x)\rangle\big| \leq \epsilon$ for all $x \in \tilde{X}$. □

Proposition 3.10 (Strict positivity of universal kernels)**.** *Let $K : X \times X \to \mathbb{K}$ be a universal PSD kernel on a metric space X. Then K is strictly positive definite, i.e., for all $n \in \mathbb{N}$, every set of n distinct points $x_1, \ldots, x_n \in X$ and all $\alpha \in \mathbb{K}^n \backslash \{0\}$ we have $\sum_{i,j=1}^{n} \alpha_i \bar{\alpha}_j K(x_i, x_j) > 0$.*

Proof. Assume K is not strictly positive definite, i.e., $\sum_{i,j=1}^n \alpha_i \bar{\alpha}_j K(x_i, x_j) = 0$ for some $\alpha \in \mathbb{K}^n \backslash \{0\}$ and $x \in X^n$. Expressing this in terms of the canonical feature map $\phi : X \to \mathcal{H}$, where $\mathcal{H}$ is the corresponding RKHS, we obtain that $\sum_{i=1}^n \alpha_i \phi(x_i) = 0$ since it has vanishing norm. Now for an arbitrary function induced by the kernel via $g(x) := \sum_{j=1}^m \beta_j \langle \phi(x), \phi(y_j) \rangle$ we obtain $\sum_{i=1}^n \alpha_i g(x_i) = \sum_{i,j} \alpha_i \beta_j \langle \phi(x_i), \phi(y_j) \rangle = 0$. Hence, the set of functions induced by the kernel cannot be dense in the set of continuous functions on the compact set $\tilde{X} := \bigcup_{i=1}^n \{x_i\}$ since any continuous function f for which $\sum_{i=1}^n \alpha_i f(x_i) \neq 0$ cannot be approximated to arbitrary accuracy. So K cannot be universal. □

Proposition 3.11 (Properties of strictly positive definite kernels)**.** *Let $K : X \times X \to \mathbb{K}$ be a strictly positive definite kernel on a set X. That is, for all $n \in \mathbb{N}$, every set of n distinct points $x_1, \ldots, x_n \in X$ and all $\alpha \in \mathbb{K}^n \backslash \{0\}$ we have $\sum_{i,j=1}^n \alpha_i \bar{\alpha}_j K(x_i, x_j) > 0$ and $K(x_i, x_j) = \overline{K(x_j, x_i)}$. Then:*

1. *Every corresponding feature space is infinite dimensional.*
2. *Every corresponding feature map is injective.*
3. *If A_+, A_- are disjoint finite subsets of X and $\phi : X \to \mathcal{H}$ is any feature map corresponding to K, then there is a $w \in \mathcal{H}$ and $b \in \mathbb{R}$ so that*

$$\mathrm{Re}\langle w, \phi(x) \rangle \begin{cases} > b, & \text{if } x \in A_+ \\ < b, & \text{if } x \in A_- \end{cases} \tag{3.22}$$

Proof. 1. If $\phi : X \to \mathcal{H}$ is any feature map for K and $d := \dim(\mathcal{H}) < \infty$, then any set of $n > d$ vectors $\{\phi(x_i)\}_{i=1}^n$ is linearly dependent. Therefore, there is an $\alpha \in \mathbb{K}^n \backslash \{0\}$ so that $0 = \sum_{i,j=1}^n \alpha_i \bar{\alpha}_j \langle \phi(x_i), \phi(x_j) \rangle = \sum_{i,j=1}^n \alpha_i \bar{\alpha}_j K(x_i, x_j)$, which implies that K is not strictly positive definite.

2. As argued in the proof of 1., if $x \neq y$, then $\phi(x)$ and $\phi(y)$ have to be linearly independent. So in particular ϕ is injective.

3. The central observation is again linear independence of the set of vectors $\{\phi(x)\}_{x \in A_+ \cup A_-}$. If we define $C_\pm := \mathrm{conv}\{\phi(x)\}_{x \in A_\pm}$ as the convex hulls of the images of the sets A_+ and A_- under ϕ, then linear independence implies that C_+ and C_- are disjoint sets. Moreover, they are closed and bounded convex subsets contained in finite dimensional subspace so that we can invoke the geometric Hahn-Banach separation theorem for compact convex sets to arrive at Eq.(3.22). □

Theorem 3.5: Translation invariant kernels

Let μ be a finite non-negative Borel measure on $X := \mathbb{R}^d$ and denote by $\chi \in C(X)$ its Fourier transform

$$\chi(x) := \int_X e^{-ix \cdot z} d\mu(z). \tag{3.23}$$

Then $K(x, y) := \chi(x - y)$ is a PSD kernel on X. Moreover, K is strictly

positive definite, if the complement of the largest open set $U \subseteq X$ that satisfies $\mu(U) = 0$ has non-zero Lebesgue measure.

Proof. Consider distinct points $x_1, \ldots, x_n \in X$ and $\alpha \in \mathbb{C}^n \backslash \{0\}$. Then

$$\begin{aligned} \sum_{k,j=1}^{n} \alpha_k \bar{\alpha}_j K(x_k, x_j) &= \sum_{k,j=1}^{n} \alpha_k \bar{\alpha}_j \int_X e^{-i(x_k - x_j)\cdot z} d\mu(z) \\ &= \int_X \underbrace{\left| \sum_{k=1}^{n} \alpha_k e^{-i x_k \cdot z} \right|^2}_{=:\psi(z)} d\mu(z) \geq 0. \end{aligned} \tag{3.24}$$

So K is a PSD kernel. Moreover, strict inequality holds in Eq.(3.24) unless the support of μ is contained in the zero set $\psi^{-1}(\{0\})$. However, $\psi^{-1}(\{0\})$ always has zero Lebesgue measure so that every μ whose support has non-zero Lebesgue measure leads to a strictly positive definite kernel. □

3.5 Rademacher bounds

Theorem 3.6: Rademacher bound for bounded inner products

Let $\rho, r > 0$ be positive constants, $x_1, \ldots, x_n$ points in a real Hilbert space $\mathcal{H}$ so that $||x_i|| \leq r$ for all i and $\mathcal{G} := \{g : \mathcal{H} \to \mathbb{R} \mid g(z) = \langle z, w \rangle,\ ||w||^{-1} \geq \rho\}$. With $G_{ij} := \langle x_i, x_j \rangle$ the empirical Rademacher complexity of $\mathcal{G}$ w.r.t. $\{x_1, \ldots x_n\}$ satisfies

$$\hat{\mathcal{R}}(\mathcal{G}) \leq \frac{\mathrm{tr}[G]^{1/2}}{n\rho} \leq \frac{r}{\rho\sqrt{n}}. \tag{3.25}$$

Proof. The first inequality follows from

$$\begin{aligned} \hat{\mathcal{R}}(\mathcal{G}) &= \frac{1}{n} \mathbb{E}_\sigma \left[\sup_{||w|| \leq 1/\rho} \left\langle \sum_{i=1}^{n} \sigma_i x_i, w \right\rangle \right] \\ &\leq \frac{1}{n\rho} \mathbb{E}_\sigma \left\| \sum_{i=1}^{n} \sigma_i x_i \right\| \leq \frac{1}{n\rho} \left[\mathbb{E}_\sigma \left\| \sum_{i=1}^{n} \sigma_i x_i \right\|^2 \right]^{1/2} \\ &= \frac{1}{n\rho} \left[\mathbb{E}_\sigma \sum_{i,j=1}^{n} \sigma_i \sigma_j \langle x_i, x_j \rangle \right]^{1/2} = \frac{1}{n\rho} \left[\sum_{i=1}^{n} \langle x_i, x_i \rangle \right]^{1/2}. \end{aligned}$$

Here the first inequality is implied by Cauchy-Schwarz and the second by Jensen's inequality (applied to the concave square root function). The last step in the chain follows from the fact that if $i \neq j$, then $\mathbb{E}_\sigma[\sigma_i \sigma_j] = \mathbb{E}_\sigma[\sigma_i]\ \mathbb{E}_\sigma[\sigma_j] = 0$ since the Rademacher variables are independent and uniform.

The second inequality in Eq.(3.25) uses in addition that $\sum_{i=1}^{n} \langle x_i, x_i \rangle \leq nr^2$. □

Bibliography

[1] Amiran Ambroladze, Emilio Parrado-Hernández, and John Shawe-Taylor. Complexity of pattern classes and the Lipschitz property. *Theor. Comput. Sci.*, 382(3):232–246, 2007.

[2] P.L. Bartlett, P.M. Long, and R.C. Williamson. Fat-shattering and the learnability of real-valued functions. *J. Comput. Syst. Sci.*, 52(3):434–452, 1996.

[3] Shai Ben-David, N. Cesa-Bianchi, D. Haussler, and P. Long. Characterization of learnability for classes of n-valued functions. *J. Comput. Syst. Sci.*, 50:74–86, 1995.

[4] Shai Ben-David and Michael Lindenbaum. Lcalization vs. Identification of Semi-Algebraic Sets. *Mach. Learn.*, 32:207–224, 1998.

[5] Anselm Blumer, Andrzej Ehrenfeucht, David Haussler, and Manfred K. Warmuth. Learnability and the Vapnik-Chervonenkis dimension. *J. ACM*, 36(4):929–965, 1989.

[6] Ralph P. Boas and Harold P. Boas. *A Primer of Real Functions*. Cambridge University Press, 1996.

[7] Amit Daniely, Sivan Sabato, Shai Ben-David, and Shai Shalev-Shwartz. Multiclass Learnability and the ERM principle.

[8] Christian J.J. Despres. The Vapnik-Chervonenkis dimension of norm on Rd. 2014.

[9] R. M. Dudley. Balls in Rk do not cut all subsets of k + 2 points. *Adv. Math. (N. Y).*, 31(3):306–308, 1979.

[10] Jean Jacod and Philip Protter. *Probability Essentials*. Springer Berlin / Heidelberg, 2 edition, 2004.

[11] Allan Pinkus. Approximation theory of the MLP model in neural networks. *Acta Numer.*, 8:143, 1999.

[12] Shai Shalev-Shwartz, Ohad Shamir, Nathan Srebro, and Karthik Sridharan. Learnability, Stability and Uniform Convergence. *J. Mach. Learn. Res.*, 11:2635–2670, 2010.