Working Title

University of Oslo

Andreas Godø Lefdalsnes

March 7, 2019

Abstract

Abstract goes here [e.g. 1, page 300].

Contents

Chapter 1

Introduction

Hello, here is some text without a meaning. This text should show what a printed text will look like at this place. If you read this text, you will get no information. Really? Is there no information? Is there a difference between this text and some nonsense like “Huardest gefburn”? Kjift – not at all! A blind text like this gives you information about the selected font, how the letters are written and an impression of the look. This text should contain all letters of the alphabet and it should be written in of the original language. There is no need for special content, but the length of words should match the language.

This is the second paragraph. Hello, here is some text without a meaning. This text should show what a printed text will look like at this place. If you read this text, you will get no information. Really? Is there no information? Is there a difference between this text and some nonsense like “Huardest gefburn”? Kjift – not at all! A blind text like this gives you information about the selected font, how the letters are written and an impression of the look. This text should contain all letters of the alphabet and it should be written in of the original language. There is no need for special content, but the length of words should match the language.

And after the second paragraph follows the third paragraph. Hello, here is some text without a meaning. This text should show what a printed text will look like at this place. If you read this text, you will get no information. Really? Is there no information? Is there a difference between this text and some nonsense like “Huardest gefburn”? Kjift – not at all! A blind text like this gives you information about the selected font, how the letters are written and an impression of the look. This text should contain all letters of the alphabet and it should be written in of the original language. There is no need for special content, but the length of words should match the language.

After this fourth paragraph, we start a new paragraph sequence. Hello, here is some text without a meaning. This text should show what a printed text will look like at this place. If you read this text, you will get no information. Really? Is there no information? Is there a difference between this text and some nonsense like “Huardest gefburn”? Kjift – not at all! A blind text like this gives you information about the selected font, how the letters are written and an impression of the look. This text should contain all letters of the alphabet and it should be written in of the original language. There is no need for special content, but the length of words should match the language.

Hello, here is some text without a meaning. This text should show what a printed text will look like at this place. If you read this text, you will get no information. Really? Is there no information? Is there a difference between this text and some nonsense like “Huardest gefburn”? Kjift – not at all! A blind text like this gives you information about the selected font, how the letters are written and an impression of the look. This text should contain all letters of the alphabet and it should be written in of the original language. There is no need for special content, but the length of words should match the language.

Chapter 2

Quantum Mechanics

The discussion in this chapter follows closely the discussion in [Sakurai — Modern Quantum Mechanics].
In quantum mechanics, a physical state is represented by a *state vector* in a complex vector space. Such a vector is called a *ket*, denoted by $|\alpha\rangle$. The state ket is postulated to contain all information about the physical state. Two kets can be added to produce a new ket:

$$|\alpha\rangle + |\beta\rangle = |\gamma\rangle .$$

They can also be multiplied by a complex number:

$$c\,|\alpha\rangle = |\alpha\rangle\, c = |\delta\rangle .$$

If c is zero the resulting ket is called a *null ket*. If c is non-zero it is postulated that the resulting ket contains the same information.
Observables such as momentum and spin are represented by operators acting on the vector space in question. Operators act on a ket from the left to produce a new ket:

$$A\,|\alpha\rangle = |\delta\rangle .$$

Of particular importance is when the action of an operator on a ket is the same as multiplication:

$$A\,|\alpha\rangle = c\,|\alpha\rangle = |\delta\rangle .$$

These kets are known as *eigenkets* and the corresponding complex numbers are known as *eigenvalues*. The physical state represented by an eigenket is known as an *eigenstate*. Any ket can be written as an expansion of eigenkets $|a'\rangle$:

$$|\alpha\rangle = \sum_{a'} c_{a'} |a'\rangle ,$$

where $c_{a'}$ is a complex coefficient. In principle there are infinitely many linearly indepedent eigenkets, depending on the dimensionality of the vector space. The uniqueness of the expansion can be proven with orthonogality of the eigenkets, which we will simply postulate.

A *bra space* is a vector space "dual" to the ket space. We postulate that for every ket $|\alpha\rangle$ there exists a bra $\langle\alpha|$. The bra space is spanned by eigenbras $\langle a'|$ corresponding to the eigenkets $|a'\rangle$. The ket and bra spaces have a dual correspondence:

$$\begin{gathered}|\alpha\rangle \leftrightarrow \langle\alpha| \\ |\alpha'\rangle , |\alpha''\rangle , \dots \leftrightarrow \langle\alpha'| , \langle\alpha''| , \dots \\ |\alpha\rangle + |\beta\rangle \leftrightarrow \langle\alpha| + \langle\beta| .\end{gathered}$$

The bra dual to $c\,|\alpha\rangle$ is postulated to be $c^*\,|\alpha\rangle$, and more generally:

$$c_\alpha |\alpha\rangle + c_\beta |\beta\rangle \leftrightarrow c_\alpha^* \langle\alpha| + c_\beta^* \langle\beta| .$$

The *inner product* of a bra and a ket is a complex number written as a bra on the left and a ket on the right. It has the fundamental property:

$$\langle\alpha|\beta\rangle = \langle\beta|\alpha\rangle^* ,$$

in other words they are complex conjugates. For this to satisfy the requirements of an inner product we must have

$$\langle\alpha|\alpha\rangle \geq 0,$$

with equality if and only if $|\alpha\rangle$ is a null ket. We define the *norm* of a ket as

$$\sqrt{\langle\alpha|\alpha\rangle},$$

which can be used to form normalized kets

$$|\alpha^{\sim}\rangle \frac{1}{\sqrt{\langle\alpha|\alpha\rangle}} |\alpha\rangle ,$$

with the property

$$\langle\alpha^{\sim}|\alpha^{\sim}\rangle = 1.$$

Two kets are said to be *orthogonal* if

$$\langle\alpha|\beta\rangle = 0.$$

2.0.1 Operators

As we briefly mentioned above, operators act on kets from the left to produce a new ket. Two operators A and B are equal

$$A = B$$

if

$$A\left|\alpha\right\rangle = B\left|\alpha\right\rangle$$

for an arbitrary ket in the relevant ket space. An operator A is said to be the *null operator* if

$$A\left|\alpha\right\rangle = 0.$$

Operators can be added, and addition operations are commutative and associative.

$$X + Y = Y + X,$$

$$(X + Y) + Z = X + (Y + Z).$$

Operators act on bras from the right to produce a new bra

$$\left\langle\alpha\right| A = \left\langle\beta\right|.$$

The ket $A\left|\alpha\right\rangle$ and the bra $\left\langle\alpha\right| A$ are in general not dual to each other. We define the *hermitian adjoint* $A^\dagger$ as

$$A\left|\alpha\right\rangle \leftrightarrow \left\langle\alpha\right| A^\dagger.$$

An operator is said to be *hermitian* if

$$A = A^\dagger.$$

Operators can be multiplied. Multiplication is associative, but non-commutative:

$$XY \neq YX,$$

$$X(YZ) = (XY)Z.$$

The left product of a ket and a bra is known as the *outer product*:

$$\left|\alpha\right\rangle\left\langle\beta\right|.$$

The outer product should be treated as an operator, while the inner product $\langle\alpha|\beta\rangle$ is a complex number.
If an operator is to the left of a ket $|\alpha\rangle A$ or to the right of a bra $A\langle\beta|$ these are illegal products, in other words not defined within the ruleset of quantum mechanics. The associative properties of operators are postulated to hold true as long as we are dealing with legal multiplications among kets, bras and operators. As an example, the outer product acting on a ket:

$$(|\alpha\rangle\langle\beta|)|\gamma\rangle,$$

can be equivalently regarded as scalar multiplication

$$|\alpha\rangle(\langle\alpha|\gamma\rangle) = |\alpha\rangle c = c|\alpha\rangle,$$

where $c = \langle\alpha|\gamma\rangle$ is just a complex number.

2.0.2 Time evolution

In quantum mechanics, time is treated not as an observable, but as a parameter. Relativistic quantum mechanics treats space and time on the same footing, but only by demoting position to a parameter.
Suppose we have a physical system $|\alpha\rangle$ at a time t_0. Denote the ket at a later time $t > t_0$ by

$$|\alpha, t; t_0\rangle.$$

As time is assumed to be a continuous parameter we expect as we evolve the system backward in time

$$\lim_{t\to t_0}|\alpha, t; t_0\rangle = |\alpha\rangle.$$

The kets separated by a time $t - t_0$ are related by the *time-evolution operator* $\mathcal{U}$:

$$|\alpha, t; t_0\rangle = \mathcal{U}(t, t_0)|\alpha, t_0\rangle.$$

If the state ket is normalized to unity at a time t_0, it must remain normalized at a later time:

$$\langle\alpha, t_0|\alpha, t_0\rangle = \langle\alpha, t; t_0|\alpha, t; t_0\rangle = 1.$$

This is guaranteed if the time evolution operator $\mathcal{U}$ is a *unitary* operator:

$$\mathcal{U}^\dagger\mathcal{U} = 1.$$

We also require the composition property:

$$\mathcal{U}(t_2, t_0) = \mathcal{U}(t_2, t_1)\mathcal{U}(t_1, t_0), \ (t_2 > t_1 > t_0).$$

If we consider an infinitesimal time-evolution operator

$$|\alpha, t_0 + dt; t_0\rangle = \mathcal{U}(t_0 + dt, t_0)\, |\alpha, t_0\rangle \, ,$$

it must reduce to the identity operator as the infinitesimal time interval dt goes to zero:

$$\lim_{dt \to 0} \mathcal{U}(t_0 + dt, t_0) = 1,$$

and we expect the difference between the operators to be of first order in dt. These requirements are all satisfied by

$$\mathcal{U}(t_0 + dt, t_0) = 1 - i\Omega dt,$$

where Ω is a Hermitian operator:

$$\Omega^\dagger = \Omega.$$

The operator Ω has the dimension inverse time. Frequency or inverse time is related to energy through the Planck-Einstein relation:

$$E = \hbar\omega.$$

In classical mechanics the Hamiltonian is the generator of time evolution, so we postulate that t Ω is related to the Hamiltonian operator H:

$$\Omega = \frac{H}{\hbar}.$$

The Hamiltonian operator represents the energy of our system, which is a physical observable and must therefore be Hermitian.

2.0.3 The Schrodinger equation

By exploiting the composition property of the time-evolution operator we obtain:

$$\mathcal{U}(t + dt, t_0) = \mathcal{U}(t + dt, t)\mathcal{U}(t, t_0) = (1 - \frac{iHdt}{\hbar})\mathcal{U}(t, t_0),$$

where the time difference $t - t_0$ is not required to be infinitesimal. This implies

$$\mathcal{U}(t+dt, t_0) - \mathcal{U}(t, t_0) = -i(\frac{H}{\hbar})dt\mathcal{U}(t, t_0),$$

and taking the limit $dt \to 0$:

$$i\hbar\frac{\partial}{\partial dt}\mathcal{U}(t, t_0) = H\mathcal{U}(t, t_0).$$

This is known as the Schrodinger equation for the time-evolution operator. Multiplying both sides by a ket $|\alpha, t_0\rangle$ leads to the Schrodinger equation:

$$i\hbar\frac{\partial}{\partial dt}\mathcal{U}(t, t_0)\left|\alpha, t_0\right\rangle = H\mathcal{U}(t, t_0)\left|\alpha, t_0\right\rangle.$$

As the ket does not depend on time this is the same as

$$i\hbar\frac{\partial}{\partial dt}\left|\alpha, t_0\right\rangle = H\left|\alpha, t_0\right\rangle.$$

Chapter 3

Manybody Quantum Mechanics

This chapter will give a brief overview of the Hartree-Fock and Density Functional Theory methods, which will be our primary workhorses for generating quantum mechanical trajectories.

Something something Born-Oppenheimer nucleonic degrees of freedom.

3.0.1 Hartree-Fock

The theory in this section is adapted from the lectures notes for the course FYS4480 - Many-body Physics at the University of Oslo [ref]. The Hartree-Fock method is an approximate method for solving a many-body system of fermions, with a Hamiltonian which can be written as the sum of a onebody part and a twobody interaction:

$$\hat{H} = \hat{H}_1 + \hat{H}_2 = \sum_i \hat{h}_1(\boldsymbol{r}_i) + \sum_{i<j} \hat{v}(r_{ij}).$$

The onebody part is typically chosen as a potential for which we have an exact solution, such as the harmonic oscillator or the coulomb attraction of an electron to the nucleus.

Hartree-Fock is a method for finding the best possible approximation to the ground state wavefunction Ψ assuming it can be written as a *Slater determinant* Φ of orthonormal spin orbitals ψ:

$$\Phi(\boldsymbol{r}_1, \ldots, \boldsymbol{r}_N, \alpha, \beta, \ldots, \sigma) = \frac{1}{\sqrt{N}} \begin{vmatrix} & x_0 & x_0^2 & \ldots & x_0^n \\ 1 & x_1 & x_1^2 & \ldots & x_1^n \\ \hdotsfor{5} \\ 1 & x_n & x_n^2 & \ldots & x_n^n \end{vmatrix} \tag{3.1}$$

Chapter 4

Molecular Dynamics

From quantum mechanics to molecular dynamics.

4.0.1 Molecular dynamics simulations

4.0.2 Molecular dynamics potentials

Chapter 5

Machine learning

Machine learning is the study of algorithms and statistical models employed by computing systems capable of performing tasks without explicit instruction. While traditional algorithms rely on some specified input and a ruleset for determining the output, machine learning is instead concerned with a set of generic algorithms which can find patterns in a broad class of data sets. This section will give a brief overview of machine learning, and more specifically the class of algorithms known as neural networks, and will follow closely the review by [Mehta et. al.] which the reader is encouraged to seek out.

Examples of machine learning problems include identifying objects in images, transcribing text from audio and making film recommendations to viewers based on their watch history. Machine learning problems are often subdivided into estimation and prediction problems. In both cases, we choose some observable $\boldsymbol{x}$ (e.g. the period of a pendulum) related to some parameters $\boldsymbol{\theta}$ (e.g. the length and the gravitational constant) through a model $p(\boldsymbol{x}|\boldsymbol{\theta})$ that describes the probability of observing $\boldsymbol{x}$ given $\boldsymbol{\theta}$. Subsequently we perform an experiment to obtain a dataset $\boldsymbol{X}$ and use these data to fit the model. Fitting the model means finding the parameters $\hat{\boldsymbol{\theta}}$ that provide the best explanation for the data. *Estimation* problems are concerned with the accuracy of $\hat{\boldsymbol{\theta}}$, whereas prediction problems are concerned with the ability of the model $p(\boldsymbol{x}|\boldsymbol{\theta})$ to make new predictions. Physics has traditionally been more concerned with the estimation of model parameters, while in this thesis we will be focused on the accuracy of the model.

Many problems in machine learning are defined by the same set of ingredients. The first is the dataset $\mathcal{D} = (\boldsymbol{X}, \boldsymbol{Y})$, where $\boldsymbol{X}$ is a matrix containing

observations of the independent variables $\boldsymbol{x}$, and $\boldsymbol{Y}$ is a matrix containing observations of dependent variables. Second is a model $\boldsymbol{F} : \boldsymbol{x} \rightarrow \boldsymbol{y}$ which is a function of the parameters $\boldsymbol{\theta}$. Finally we have a cost function $\mathcal{C}\left(\boldsymbol{Y}, \boldsymbol{F}\left(\boldsymbol{X}; \boldsymbol{\theta}\right)\right)$ that judges the performance of our model at generating predictions.
In the case of linear regression we consider a set of independent observations $\boldsymbol{X} = \begin{bmatrix} \boldsymbol{x}_1 & \boldsymbol{x}_2 & \dots & \boldsymbol{x}_N \end{bmatrix}$ related to a set of dependent observations $\boldsymbol{y} = (y_1, y_2, \dots, y_N)$ through a linear model $f(\boldsymbol{x}; \boldsymbol{\theta}) = x_1 \cdot w_1 + x_2 \cdot w_2 + \dots + x_P \cdot w_P$, with parameters $\boldsymbol{\theta} = (w_1, w_2, \dots, w_P)$. The cost function is the well known sum of least squares $\mathcal{C}(\boldsymbol{y}, f(\boldsymbol{X}; \boldsymbol{\theta})) = \sum_i^N (y_i - f(\boldsymbol{x}_i; \boldsymbol{\theta}))^2$ and the best fit is chosen as the set of parameters which minimize this cost function: $\hat{\boldsymbol{\theta}} = \underset{\boldsymbol{\theta}}{\operatorname{argmin}}\, \mathcal{C}(\boldsymbol{Y}, f(\boldsymbol{X}; \boldsymbol{\theta}))$.

5.0.1 Basics of statistical learning

Statistical learning theory is a field of statistics dealing with the problem of making predictions from data. We begin with an unknown function
$y = f(x)$ and our goal is to develop a function $h(x)$ such that $h \sim f$. We fix a hypothesis set $\mathcal{H}$ that the algorithm is willing to consider. The expected error of a particular h over all possible inputs x and outputs y is:

$$E[h] = \int_{X \times Y} \mathcal{C}(h(x), y) \rho(x, y) dx dy,$$

where $\mathcal{C}$ is a cost function and $\rho(x, y)$ is the joint probability distribution for x and y. This is known as the *expected error*. Since this is impossible to compute without knowledge of the probability distribution ρ, we instead turn to the *empirical error*. Given n data points the empirical error is given as:

$$E_S[h] = \frac{1}{n} \sum_i^n \mathcal{C}(h(x_i), y_i).$$

The *generalization error* is defined as the difference between the expected and empirical errors:

$$G = E[h] - E_S[h].$$

We say an algorithm is able to learn from data or *generalize* if

$$\lim_{n \to \infty} G = 0.$$

We are in general unable to compute the expected error, and therefore unable to compute the generalization error. The most common approach known as *cross-validation* is to estimate the generalization error by subdividing our dataset into a *training* set and a *test* set. The value of the cost function on the training set is called the *in-sample* error and the value of the cost function on the test set the *out-of-sample* error. Assuming the dataset is sufficiently large and representative of f, and the subsampling into train and test datasets is unbiased, the in-sample error can serve as an appropriate proxy for the generalization error.
In figure 5.1 we show the typical evolution of the errors as the number of data points increase. It is assumed that the function being learned is sufficiently complicated that we cannot learn it exactly, and that we have a sizeable number of data points available. The in-sample error will decrease monotonically, as our model is not able to learn the underlying data exactly. In contrast, the out-of-sample error will decrease, as the sampling noise decreases and the training data set becomes more representative of the underlying probability distribution. In the limit, these errors both approach same value, which is known the model *bias*. The bias represents the best our model could do in the infinite data limit. The out-of-sample error produced from the sampling noise is known as *variance*, and will vanish completely given an infinite representative data set.

In figure 5.2 we show the typical evolution of the out-of-sample error as the model *complexity* increases. Model complexity is a measure of the degrees of freedom in the model space, for example the number of coefficients in a polynomial regression. In the figure we can see that bias decreases monotonically as model complexity increases, as the model is able to fit a larger space of functions. However, the variance will also increase as the model becomes more susceptible to sampling noise. In general the lowest out-of-sample error, and therefore generalization error, is achieved at an intermediate model complexity. We also find that as model complexity increases, a larger amount of data points is required to be able to reasonably fit the true function.

5.0.2 Bias-variance decomposition

Consider a dataset $\mathcal{D}(\boldsymbol{X}, \boldsymbol{y})$ of n pairs of independent and dependent variables. Assume the true data is generated from a noisy model:

$$y = f(\boldsymbol{x}) + \epsilon,$$

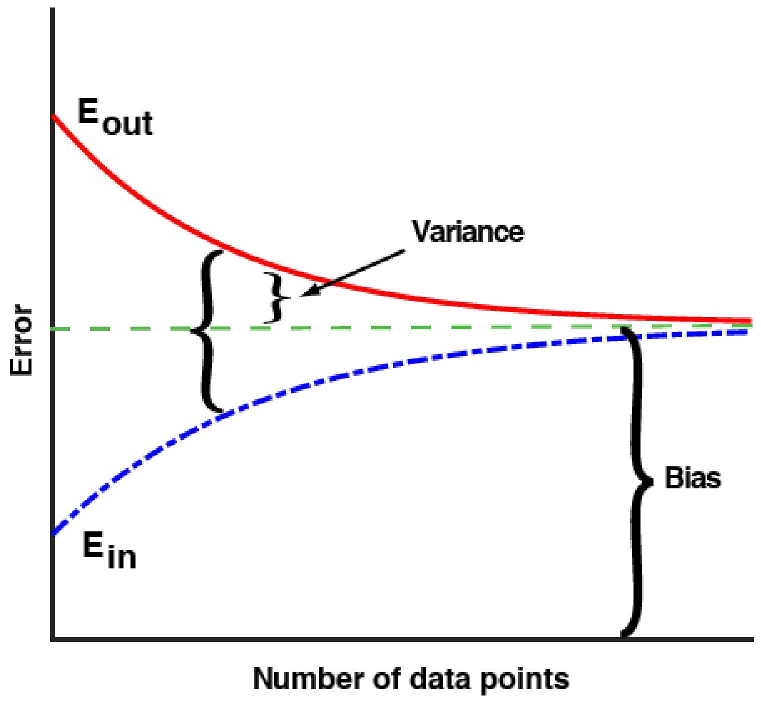

Figure 5.1: Typical in-sample and out-of-sample error as a function of the number of data points. It is assumed that the number of data points is not small, and that the true function cannot be exactly fit.

where ϵ is normally distributed with mean μ and standard deviation σ. Assume that we have an estimator $h(\boldsymbol{x};\boldsymbol{\theta})$ trained by minimizing a cost function $\mathcal{C}(\boldsymbol{y}, h(\boldsymbol{x}))$ which we take to be the sum of squared errors:

$$\mathcal{C}(\boldsymbol{y}, h(\boldsymbol{x})) = \sum_{i}^{n} (y_i - h(\boldsymbol{x}_i;\boldsymbol{\theta}))^2.$$

Our best estimate for the model parameters:

$$\boldsymbol{\theta}_{\mathcal{D}} = \arg\min_{\boldsymbol{\theta}} \mathcal{C}(\boldsymbol{y}, h(\boldsymbol{x};\boldsymbol{\theta})),$$

is a function of the dataset $\mathcal{D}$. If we imagine we have a set of datasets $\mathcal{D}_j = (\boldsymbol{y}_j, \boldsymbol{X}_j)$, each with n samples, we would like to calculate the expectation value of the cost function over all these datasets $E_{\mathcal{D},\epsilon}$. We would also like to

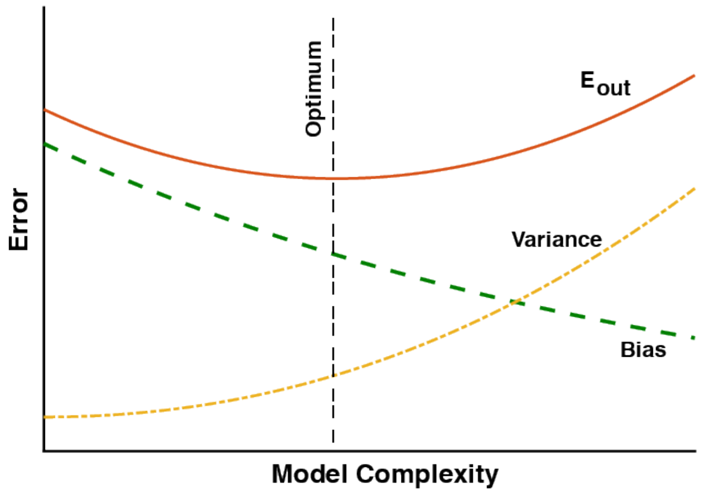

Figure 5.2: Typical out-of-sample error as a function of model complexity for a fixed dataset. Bias decreases monotonically with model complexity, while variance increases as a result of sampling noise.

calculate the expectation value over different instances of the noise ϵ. The expected generalization error can be decomposed as:

$$\begin{aligned} E_{\mathcal{D},\epsilon}[\mathcal{C}(\boldsymbol{y}, h(\boldsymbol{X};\boldsymbol{\theta}_{\mathcal{D}}))] &= E\left[\sum_i (y_i - h(\boldsymbol{x}_i;\boldsymbol{\theta}_{\mathcal{D}}))^2\right] \\ &= \sum_i \sigma_\epsilon^2 + E_{\mathcal{D}}[(f(\boldsymbol{x}_i) - f(\boldsymbol{x}_i;\boldsymbol{\theta}_{\mathcal{D}}))^2]. \end{aligned} \tag{5.1}$$

The second term can be further decomposed as

$$\begin{aligned} &E_{\mathcal{D}}[(f(\boldsymbol{x}_i) - f(\boldsymbol{x}_i;\boldsymbol{\theta}_{\mathcal{D}}))^2] \\ &= (f(\boldsymbol{x}_i) - E_{\mathcal{D}}[h(\boldsymbol{x}_i;\boldsymbol{\theta}_{\mathcal{D}})])^2 + E[(h(\boldsymbol{x}_i;\boldsymbol{\theta}_{\mathcal{D}}) - E[h(\boldsymbol{x}_i;\boldsymbol{\theta}_{\mathcal{D}}])^2] \end{aligned} \tag{5.2}$$

The first term is what we have referred to as the bias:

$$\text{Bias}^2 = \sum_i (f(\boldsymbol{x}_i) - E_{\mathcal{D}}[h(\boldsymbol{x}_i;\boldsymbol{\theta}_{\mathcal{D}})])^2. \tag{5.3}$$

The bias measures the expectation value of the deviation of our model from the true function, i.e. the best we can do in the infinite data limit.

The second term is what we have referred to as the variance:

$$\mathrm{Var} = \sum_i E[(h(\boldsymbol{x}_i;\boldsymbol{\theta}_{\mathcal{D}}) - E[h(\boldsymbol{x}_i;\boldsymbol{\theta}_{\mathcal{D}}])^2] \tag{5.4}$$

The variance measures the deviation of our model due to finite-sampling effects. Combining these effects we can decompose the out-of-sample error into:

$$E_{\mathrm{out}} = \mathrm{Bias}^2 + \mathrm{Var} + \mathrm{Noise}, \tag{5.5}$$

with Noise $= \sum_i \sigma_\epsilon^2$.
In general it can be much more difficult to obtain sufficient good data than to train a very complex model. Therefore it is often useful in practice to use a less complex model with higher bias, because it is less susceptible to finite-sampling effects.

5.0.3 Neural networks

Artificial Neural Networks (ANN) or Deep Neural Networks (DNN) are supervised learning models vaguely inspired by biological neural networks. The building blocks of neural networks are neurons that take a vector input of d features $\boldsymbol{x} = (x_1, \dots, x_d)$ and produce a scalar output $a(\boldsymbol{x})$. A neural networks consists of layers of these neurons stacked together with the output of one layer serving as input for another. The first layer is typically known as the *input layer*, the middle layers as *hidden layers* and the final layer the *output layer*. The basic architecture is shown in figure 5.3. In almost all cases the output $a_i(\boldsymbol{x})$ of neuron i can be decomposed into a linear operation on the inputs passed through a non-linear activation function:

$$a_i(\boldsymbol{x}) = \sigma_i(z_i),$$

where σ_i is a non-linear function and z_i is the dot product between the inputs $\boldsymbol{x}$ and a set of neuron-specific weights $\boldsymbol{w}_i$:

$$z_i = \boldsymbol{x}^T \boldsymbol{w}_i + b_i.$$

The term b_i is a neuron-specific re-centering of the input.
Typical choices of non-linearities/activation functions include the sigmoid and hyperbolic tangent functions, and Rectified Linear Units (ReLU). When the activation function is non-linear, the neural network with a single hidden

layer can be proven to be a *universal function approximator*, given an arbitrarily large number of neurons. We typically also want functions that are monotonic and smooth with a monotonic derivative.

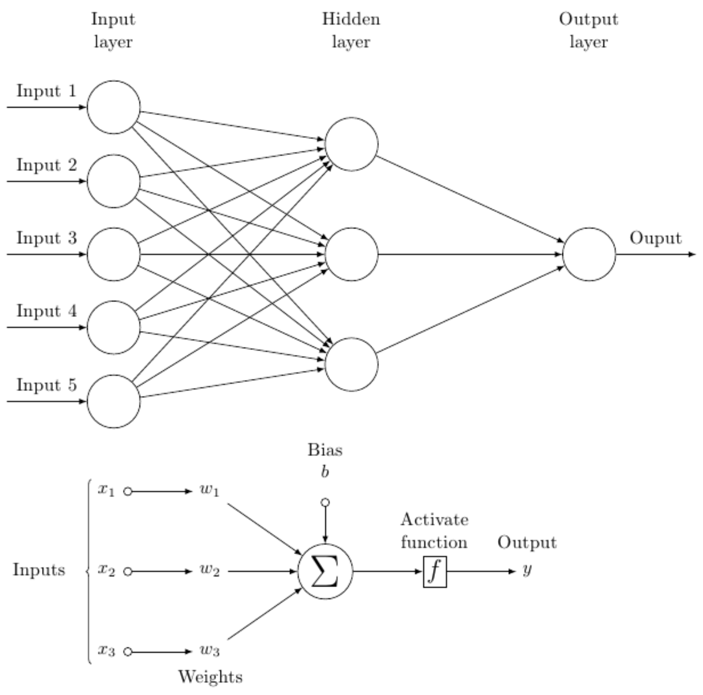

Figure 5.3: Text.

The simplest neural networks are known as *feed-forward* neural networks (FNN). The input layer is the vector $\boldsymbol{x}$ of inputs, while each neuron in the first hidden layer performs a dot product between its weights $\boldsymbol{w}_i$ and the inputs and passes it through a non-linearity σ_i. The activation function is typically shared across one or multiple layers $\sigma_i = \sigma$. The vector of neuron outputs $\boldsymbol{a}_i$ serves as input to the next hidden layer until we reach the final layer. In the final layer the choice of activation function is dependent on the problem we are trying to solve. If we are performing non-linear regression the final activation function is often the identity $\sigma_i(z) = z$, or if we are doing classification the soft-max function is often employed.

Let $\boldsymbol{x}$ be a vector of $d = 1, \ldots, D$ inputs or *features*. Let $a_i^{(h)}$ denote the output of neuron $i = 1, \ldots, N_d$ in layer $h = 1, \ldots, H$. The output of neuron i in the first hidden layer $a_i^{(1)}$ is thus:

$$z_i^{(1)} = \boldsymbol{x}^T \boldsymbol{w}_i^{(1)} + b_i^{(1)}.$$

$$a_i^{(1)} = \sigma_i^{(1)}(z_i^{(1)}),$$

The inputs are iterated through each hidden layer until we reach the final layer:

$$z_i^{(H)} = (\boldsymbol{a}_{H-1})^T \boldsymbol{w}_i^{(H)} + b_i^{(H)}.$$

$$\begin{aligned} a_i^{(H)} &= o_i = \sigma_i^{(H)}(z_i^{(H)}) \\ &= \sigma_i^{(H)}\left((\boldsymbol{a}_{H-1})^T \boldsymbol{w}_i^{(H)} + b_i^{(H)}\right) \\ &= \sigma_i^{(H)}\left(\left((\sigma_1^{(H-1)}, \ldots, \sigma_{N_{H-1}}^{(H-1)})\right)^T \boldsymbol{w}_i^{(H)} + b_i^{(H)}\right). \end{aligned} \tag{5.6}$$

This allows us to compose a complicated function $\boldsymbol{F} : \mathbb{R}^D \to \mathbb{R}^O$, with D the number of inputs and O the number of outputs. The *universal approximation theorem* tells us that this simple architecture can approximate any of a large set of continuous functions given appropriate choice of weights $\boldsymbol{w}_i^h$ and mild assumptions on the activation functions. The theorem requires only a single hidden layer, where the strength of the approximation relies on the number of neurons. In practice it has been found that adding more layers produces faster convergence and higher accuracy, which has given rise to the field of *deep learning*.

5.0.4 Backpropagation

Given a set of datapoints $(\boldsymbol{x}_i, y_i),\ i = 1, \ldots, n$, the value of the cost function is entirely determined by the weights and biases of each neuron in the network. We define learning narrowly as adjusting the parameters of the network in order to minimize the cost function.
Gradient descent is a simple, but powerful method of finding the minima of differentiable functions. Given a function $F : \mathbb{R}^d \to \mathbb{R}$, and an initial value $\boldsymbol{x}_0$ we define an iterative procedure:

$$\boldsymbol{x}_{n+1} = \boldsymbol{x}_n - \eta \nabla F(\boldsymbol{x}_n),$$

where η is known as the *learning rate*. The procedure terminates when the norm $|\nabla F(\boldsymbol{x}_n)|$ or $|x_{n+1} - x_n|$ is appropriately small.
The learning rate is not necessarily fixed throughout the procedure, and proves crucial to the convergence of the method. If f is convex, and η is reasonably small, convergence is guaranteed. Convergence may be very slow however, and if f is not convex you are only guaranteed to find local minima, and this makes the method very sensitive to initial conditions.

In order to train the model, we need to calculate the derivative of the cost function with respect to a very large number of parameters multiple times. However, numerical calculation of gradients is very time consuming. The *backpropagation* algorithm is a clever use of the chain rule that allows us to calculate gradients efficiently.

Assume that there are L layers in our network with $l = 1, 2, ..., L$ indexing the layers, including the output layer and all the hidden layers. Let w^l_{ij} denote the weight for the connection from the i-th neuron in layer $l-1$ to the j-th neuron in layer l. Let b^l_j denote the bias of this j-th neuron.

The activation a^l_j of the j-th neuron in the l-th layer is related to the activities of the neurons in the layer $l-1$ by:

$$a^l_j = f\left(\sum_i w^l_{ij} a^{l-1}_i + b^l_j\right) = f\left(z^l_j\right),$$

where f is some activation function.

The cost function $\mathcal{C}$ depends directly on the activations in the output layer, and indirectly on the activations in all the lower layers. Define the error Δ^L_j of the j-th neuron in the L-th (final) layer as the change in cost function with respect to the weighted input z^L_j:

$$\Delta^L_j = \frac{\partial \mathcal{C}}{\partial z^L_j}.$$

Define analogously the error Δ^l_j of neuron j in the l-th layer as the change in cost function with respect to the weighted input z^l_j:

$$\Delta^l_j = \frac{\partial \mathcal{C}}{\partial z^l_j}.$$

This can also be interpreted as the change in cost function with respect to the bias b^l_j:

$$\Delta_j^l = \frac{\partial \mathcal{C}}{\partial z_j^l} = \frac{\partial \mathcal{C}}{\partial b_j^l}\frac{\partial b_j^l}{\partial z_j^l} = \frac{\partial \mathcal{C}}{\partial b_j^l},$$

since $\partial b_l^j / \partial z_j^l = 1$.

The error depends on neurons in layer l only through the activation of neurons in layer $l+1$, so using the chain rule we can write:

$$\begin{aligned} \Delta_j^l = \frac{\partial \mathcal{C}}{\partial z_j^l} &= \sum_i \frac{\partial \mathcal{C}}{\partial z_i^{l+1}} \frac{\partial z_i^{l+1}}{\partial z_j^l} \\ &= \sum_i \Delta_i^{l+1} \frac{\partial z_i^{l+1}}{\partial z_j^l} \\ &= \sum_i \Delta_i^{l+1} w_{ij}^{l+1} f'(z_j^l) \\ &= \left(\sum_i \Delta_i^{l+1} w_{ij}^{l+1} \right) f'(z_j^l). \end{aligned} \tag{5.7}$$

The sum comes from the fact that any error in neuron j in the l-th layer propagates to all the neurons in the layer $l+1$, so we have to sum up these errors.

This gives us the equations we need to update the weights and biases of our network:

$$\frac{\partial \mathcal{C}}{\partial w_{ij}^l} = \frac{\partial \mathcal{C}}{\partial z_j^l}\frac{\partial z_j^l}{\partial w_{ij}^l} = \Delta_j^l a_i^{l-1}.$$

$$\frac{\partial \mathcal{C}}{\partial b_j^l} = \Delta_j^l.$$

Now, if we have the error of every neuron j at the output layer, Δ_j^L, equation **??** gives us the recipe for calculating the error in the preceding layer until we reach the first hidden layer, and we are done. All we are missing is the error at the output layer:

$$\Delta_j^L = \frac{\partial \mathcal{C}}{\partial z_j^L} = \frac{\partial}{\partial z_j^L} \left(- \sum_c y_c \log f(z_c^L) \right),$$

where f is the softmax function. Taking the derivative of each term:

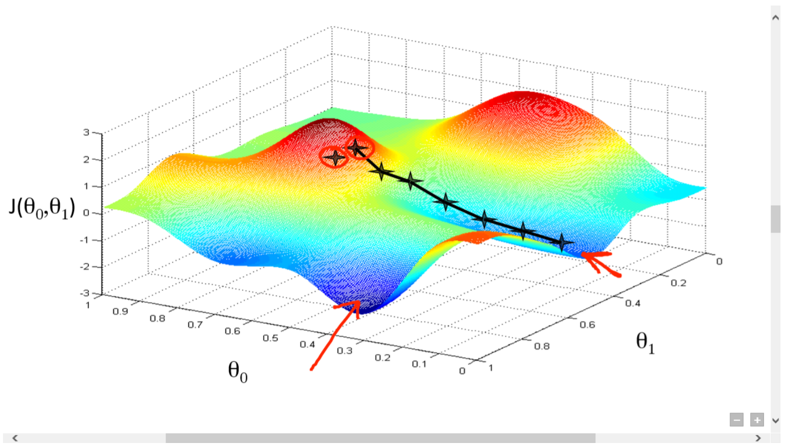

Figure 5.4: Text.

$$\frac{\partial \log f(z_c^L)}{\partial z_j^L} = \frac{\partial}{\partial z_j^L}\left(z_c^L - \log\left(\sum_c \exp\left(z_c^L\right)\right)\right) = \delta_{jc} - f(z_j^L),$$

where δ_{jc} is the Kronecker-Delta. This gives us the final expression we need:

$$\begin{aligned}\Delta_j^L &= -\sum_c y_c\left(\delta_{jc} - f(z_j^L)\right)\\ &= -\sum_c y_c\delta_{jc} + \sum_c y_c f(z_j^L)\\ &= -y_j + f(z_j^L)\sum_c y_c\\ &= f(z_j^L) - y_j\\ &= \hat{y}_j - y_j.\end{aligned} \tag{5.8}$$

5.0.5 Optimization

Gradient descent, momentum, ADAM, initialization, batching, vanishing, normalization (?).

Chapter 6

Conclusion

This is the second paragraph. Hello, here is some text without a meaning. This text should show what a printed text will look like at this place. If you read this text, you will get no information. Really? Is there no information? Is there a difference between this text and some nonsense like "Huardest gefburn"? Kjift – not at all! A blind text like this gives you information about the selected font, how the letters are written and an impression of the look. This text should contain all letters of the alphabet and it should be written in of the original language. There is no need for special content, but the length of words should match the language.

And after the second paragraph follows the third paragraph. Hello, here is some text without a meaning. This text should show what a printed text will look like at this place. If you read this text, you will get no information. Really? Is there no information? Is there a difference between this text and some nonsense like "Huardest gefburn"? Kjift – not at all! A blind text like this gives you information about the selected font, how the letters are written and an impression of the look. This text should contain all letters of the alphabet and it should be written in of the original language. There is no need for special content, but the length of words should match the language.

After this fourth paragraph, we start a new paragraph sequence. Hello, here is some text without a meaning. This text should show what a printed text will look like at this place. If you read this text, you will get no information. Really? Is there no information? Is there a difference between this text and some nonsense like "Huardest gefburn"? Kjift – not at all! A blind text like this gives you information about the selected font, how the letters are written and an impression of the look. This text should contain all letters of the alphabet and it should be written in of the original language. There is no need for special content, but the length of words should match the language.

Hello, here is some text without a meaning. This text should show what a printed text will look like at this place. If you read this text, you will get no information. Really? Is there no information? Is there a difference between this text and some nonsense like “Huardest gefburn”? Kjift – not at all! A blind text like this gives you information about the selected font, how the letters are written and an impression of the look. This text should contain all letters of the alphabet and it should be written in of the original language. There is no need for special content, but the length of words should match the language.

This is the second paragraph. Hello, here is some text without a meaning. This text should show what a printed text will look like at this place. If you read this text, you will get no information. Really? Is there no information? Is there a difference between this text and some nonsense like “Huardest gefburn”? Kjift – not at all! A blind text like this gives you information about the selected font, how the letters are written and an impression of the look. This text should contain all letters of the alphabet and it should be written in of the original language. There is no need for special content, but the length of words should match the language.

Appendix A

Appendix Title

And after the second paragraph follows the third paragraph. Hello, here is some text without a meaning. This text should show what a printed text will look like at this place. If you read this text, you will get no information. Really? Is there no information? Is there a difference between this text and some nonsense like "Huardest gefburn"? Kjift – not at all! A blind text like this gives you information about the selected font, how the letters are written and an impression of the look. This text should contain all letters of the alphabet and it should be written in of the original language. There is no need for special content, but the length of words should match the language.

After this fourth paragraph, we start a new paragraph sequence. Hello, here is some text without a meaning. This text should show what a printed text will look like at this place. If you read this text, you will get no information. Really? Is there no information? Is there a difference between this text and some nonsense like "Huardest gefburn"? Kjift – not at all! A blind text like this gives you information about the selected font, how the letters are written and an impression of the look. This text should contain all letters of the alphabet and it should be written in of the original language. There is no need for special content, but the length of words should match the language.

Hello, here is some text without a meaning. This text should show what a printed text will look like at this place. If you read this text, you will get no information. Really? Is there no information? Is there a difference between this text and some nonsense like "Huardest gefburn"? Kjift – not at all! A blind text like this gives you information about the selected font, how the letters are written and an impression of the look. This text should contain all letters of the alphabet and it should be written in of the original language.

There is no need for special content, but the length of words should match the language.

This is the second paragraph. Hello, here is some text without a meaning. This text should show what a printed text will look like at this place. If you read this text, you will get no information. Really? Is there no information? Is there a difference between this text and some nonsense like “Huardest gefburn”? Kjift – not at all! A blind text like this gives you information about the selected font, how the letters are written and an impression of the look. This text should contain all letters of the alphabet and it should be written in of the original language. There is no need for special content, but the length of words should match the language.

And after the second paragraph follows the third paragraph. Hello, here is some text without a meaning. This text should show what a printed text will look like at this place. If you read this text, you will get no information. Really? Is there no information? Is there a difference between this text and some nonsense like “Huardest gefburn”? Kjift – not at all! A blind text like this gives you information about the selected font, how the letters are written and an impression of the look. This text should contain all letters of the alphabet and it should be written in of the original language. There is no need for special content, but the length of words should match the language.

Bibliography

[1] Albert Einstein. “Zur Elektrodynamik bewegter Körper. (German) [On the electrodynamics of moving bodies]”. In: *Annalen der Physik* 322.10 (1905), pp. 891–921. DOI: `http://dx.doi.org/10.1002/andp.19053221004`.