TECHNISCHE UNIVERSITÄT WIEN

DIPLOMARBEIT

Fisher Information Flow in Neural Networks

zur Erlangung des akademischen Grades

Diplom-Ingenieur

im Rahmen des Studiums

Technische Physik

eingereicht von

Maximilian Weimar
Matrikelnummer 01427149

ausgeführt am Institut für Theoretische Physik
der Fakultät für Physik der Technischen Universität Wien

Betreuung
Betreuer: Univ.Prof. Dipl.-Ing. Dr.techn. Stefan Rotter

Wien, 11.09.2022

(Unterschrift Verfasser/in) (Unterschrift Betreuer/in)

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU WIEN Bibliothek Your knowledge hub

Contents

1 Introduction **4**

2 Statistical Estimation Theory and Hypothesis Testing **5**

2.1 Parameter Estimation . . . 5

2.1.1 The Mean Squared Error Loss Function . . . 6

2.1.2 The Fisher Information . . . 7

2.1.3 Example: Fisher Information for Normal Distribution . . . 8

2.1.4 The Fisher Information Inequality . . . 9

2.1.5 The Cramér-Rao Lower Bound . . . 10

2.1.6 The Exponential Family and Efficient Estimators . . . 11

2.2 Regression . . . 12

2.2.1 Estimation of Random Parameters: The Connection between Parameter Estimation and Regression . . . 14

2.3 Hypothesis Testing and Decisions . . . 15

2.3.1 Likelihood Ratio Test and Bayes Error . . . 16

2.3.2 An Inequality for the Bayes Error . . . 18

3 Calculation of the Fisher Information **18**

3.1 Estimation of the Fisher Information with a Monte Carlo Method . . . 18

3.2 Density Estimation . . . 19

3.2.1 The Histogram Method . . . 20

3.2.2 The Maximum Likelihood Method . . . 20

3.2.3 Advanced Methods of Density Estimation . . . 21

3.3 Calculating the Fisher Information from Estimated Densities . . . 22

3.3.1 Calculating the Fisher Information from a Histogram Distribution . . . 22

3.3.2 Calculating the Fisher Information with Maximum Likelihood Estimation . . . 23

3.4 Transformation of Probability Densities . . . 23

3.4.1 Invertible Transformations . . . 23

3.4.2 Non-Invertible Transformations . . . 24

3.4.3 Neural Network Transformations . . . 25

3.5 A Lower Bound of the Fisher Information . . . 25

3.5.1 Linear Fisher Information and Efficiency . . . 26

3.5.2 Example: The Linear Fisher Information of a Univariate Normal Distribution . . . 27

3.5.3 The Lower Bound Maximization Algorithm . . . 27

4 Artificial Neural Networks and Deep Learning **30**

4.1 Deep Neural Networks . . . 30

4.2 The Structure of Deep Neural Networks . . . 30

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

4.3 Training of an ANN . . . 32
4.3.1 Cost Function and Loss Function . . . 32
4.3.2 The Backpropagation Algorithm . . . 33
4.4 Underfitting and Overfitting . . . 33
4.4.1 Training, Validation and Testing . . . 34

5 Fisher Information in Neural Network Regression Models 35
5.1 Approximating a Cramér-Rao Efficient Estimator . . . 35
5.1.1 Calculating the Efficiencies of Neural Network Estimators . . . 36
5.1.2 Transformation of the Training Data: Increasing the Linear Fisher Information . . . 38
5.2 Early Stopping based on Fisher Information . . . 39
5.3 The Propagation of Fisher Information through Deep Neural Networks . . . 40
5.3.1 Fisher Information Inequality in Deep Neural Networks . . . 41
5.3.2 The Fisher Information in Shallow Neural Networks . . . 41
5.3.3 The Linear Fisher Information Approach . . . 42
5.3.4 Tracking the Fisher Information During the Training . . . 44
5.3.5 Error Detection with Fisher Information . . . 45

6 Quantifying the Performance of Neural Network Classification Models 47
6.1 Approximately Achieving the Bayes Error . . . 47
6.2 The Propagation of the Bayes Error in Deep Neural Networks . . . 48
6.2.1 Tracking the Bayes Error During the Training . . . 49
6.2.2 Error Detection with the Bayes Error . . . 50

7 Summary and Outlook 51

8 Acknowledgements 52

Appendices 53

A Estimation Theory 53
A.1 Fisher Information under Reparametrization . . . 53
A.2 Generating Distributions with Arbitrarily Shaped Fisher Information . . . 53
A.3 General Properties of the Fisher Information . . . 54
A.3.1 Chain Rule for the Fisher Information . . . 55
A.4 Invertible Transformations are Sufficient Statistics . . . 56

B Mathematical Properties of Probability Densities 57
B.1 Finite Difference Method . . . 57
B.2 Transformation of Expectations . . . 57
B.3 The Moments under Affine Transformations . . . 58
B.4 Singular Value Decomposition . . . 58

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

B.5 Transformation of Probability Densities 59
B.5.1 Neural Network Transformations 59
B.6 Multivariate Gaussian Densities under Affine Transformations 60

C Fisher Information in Neural Network Regression Models 60
C.1 Analytic Fisher Information in the First Hidden Layer 60
C.2 A Markov Chain Monte Carlo Algorithm 61
C.3 The Calculation of $\partial_\theta \mu$ is a Source of Error 63

D Bayes Error in Neural Network Classification Models 64
D.1 The Bayes Error under Invertible Transformations 64
D.2 Bayes Error Inequality 65

E Specifications of the Neural Networks 66
E.1 Generation of the Datasets used for Training, Validation and Testing 66
E.2 Neural Network Efficiency 67
E.3 Maximising the LFI of the Training Data 68
E.4 Early Stopping 69
E.5 Shallow Neural Networks 69
E.6 Linear Fisher Information Approach 70
E.7 Fisher Information during Training 71
E.8 Error Detection with Fisher Information 71
E.9 Source of Error 72
E.10 Monte Carlo 72
E.11 Bayes Error Propagation 73
E.12 Bayes Error During the Training 73
E.13 Error Detection with Bayes Error 74

References 76

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

1 Introduction

In recent years, the field of Deep Learning (DL) [1, 2] has received massive attention. DL techniques employ Artificial Neural Networks (ANNs) based on a layered structure. During the so-called 'training' of a DL model, the connections between the artificial neurons are adapted, such that the network learns a non-linear relation from data sets. Besides its prominent applications, such as natural language processing and image recognition, DL finds use in natural sciences and physics. Successful applications have been found, e.g., in optics [3], where ANNs can learn to solve inverse scattering problems from speckle patterns that are recorded after coherent light was scattered through a diffusive medium. The inverse problem itself is very challenging, however, a statistical model based on a DNN can capture the relevant statistical properties of the speckle pattern's distribution.

Although DL was proven to be exceptionally successful at solving tasks that are often too complicated for humans to perform, a complete and widely accepted theory explaining the processes happening within the network during the training does not yet exist. ANNs are still considered 'black boxes' that are empirically proven to work. A prominent attempt of peeking into this black box can be found in Ref. [4, 5], where the information theoretic concept of Mutual Information (MI) between the Deep Neural Network's (DNN's) layers and the input and output, respectively, is studied. Despite the success of this information theoretic approach, it is not without debate [6, 7]. One caveat is, for instance, that noise must be present in the ANN, in order to arrive at a meaningful definition of MI.

In this thesis we offer a different framework for analysing ANNs in detail. Based on the mathematical field of estimation theory [8], we characterize the individual layers of a DNN using the quantity of Fisher Information (FI) [8, 9, 10]. Considering a probability distribution that depends on some continuous parameter, the FI determines how precisely this parameter can be estimated from a sample, which was drawn from that distribution. In the example above, where light is scattered through a diffusive medium, the sample corresponds to the recorded speckle patterns. In a basic experimental realization, the continuous parameter could, e.g., be the position of the coherent light source. We show that a large class of models, the so-called 'regression' tasks, can be formulated as parameter estimation problems, which allows us to directly use the FI as a measure of a model's quality. By regarding a DNN as a series of consecutive transformations, the FI can be defined for each layer individually. From the data-processing inequality [11] it follows that, as a function of the layer depth, the FI cannot increase. We interpret the FI as a quantity that flows through the layers of a DNN and associate faulty layers as sinks for the FI. We expect that studying the FI flow at different stages of the network's training helps understanding how ANNs learn. Moreover, a FI can be assigned directly to the data set at hand, independent of the ANN itself. This way, an upper bound for the network's performance, the so-called Cramér-Rao Lower Bound (CRLB) can be estimated. Later in this work, we extend the analysis from regression models to another large group of DL applications, the 'classification' tasks.

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

At first, we give a brief introduction to statistical estimation theory and hypothesis testing (Sec. 2), as these fields provide the mathematical foundation for our analysis of ANNs. In a next step in Sec. 3, we elaborate on technical details concerning the calculation of the FI, which in general is a non-trivial task. After introducing the basics of ANNs and DL in Sec. 4, we dedicate Sec. 5 to applying the concept of FI to DNNs trained to perform regression tasks. In Sec. 6, we try to find an extension of this analysis to classification models. The appendices Sec. A, B, C, and D contain additional mathematical calculations and derivations. Technical details about the ANNs that are used within the scope of this thesis are listed in Sec. E.

2 Statistical Estimation Theory and Hypothesis Testing

In order to provide the mathematical background to analyse Deep Neural Networks from an estimation theoretic [8] point of view, the basics of statistical parameter estimation are introduced in this section. Parameters are calculated from an observation, i.e., a finite sample of noisy data. The data is of random nature, thus it can be regarded as a sample drawn from a probability distribution. As the size of the sample at hand is finite, the estimated quantity itself is inherently noisy. Common parameters that are estimated in statistics are the mean or the variance of a probability distribution. Physical observables can often be interpreted as parameters of distributions as well, as for example the temperature of an ensemble of particles or the position of an emitter of particles/light. Note that estimation theory typically deals with continuous parameters, although the observable can be of either continuous or discrete nature. The corresponding task of finding parameters that can only take on discrete values is part of decision theory [12], which is closely connected to hypothesis testing. Due to the randomness of the estimate itself in both, parameter estimation and decision problems, an important task is to measure the performance of estimation procedures and to quantify their fundamental limitations.

2.1 Parameter Estimation

Given a data set $D = (\boldsymbol{x}_1, \boldsymbol{x}_2, ..., \boldsymbol{x}_N)$ of the size N, which can be regarded as a finite sample of a probability distribution $x_i \sim P(\theta)$ $(i = 1, ..., N)$ that depends on some continuous parameter θ, we seek for a function $\hat{\theta}$ that gives an estimation of the true parameter value θ based on the measured data D, i.e., $\hat{\theta}(D) \approx \theta$. As $\hat{\theta}(D)$ is constructed from random variables x_i, it is a random variable itself and follows a probability distribution $P_{\text{estimator}}(\hat{\theta})$. This transformed random variable should have the property that its expectation $\mathrm{E}[\hat{\theta}]$ coincides with the true value θ. An estimator with this attribute is referred to as an unbiased estimator. The bias [8] is defined as $b(\hat{\theta}, \theta) = \mathrm{E}[\hat{\theta}] - \theta$, thus for an unbiased estimator $b = 0$. When an estimator has a small bias one says that it has a high accuracy. The concept of accuracy is depicted in Fig. 1. A finite bias typically arises in one of the following ways:

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

- A bias can be caused by fluctuations in the data that do not average out when the sample size gets infinitely large.

- Bias occurs when the estimator does not properly model the problem at hand, e.g., due to overfitting. An example in the context of machine learning is given in Sec. 4.4.

- For estimators used in practice, a weak version of unbiasedness where the bias is not equal to zero but vanishes in the limit of infinite sample size is sometimes sufficient. Such an estimator is said to be asymptotically unbiased [8].
  However, when an asymptotically unbiased estimator is constructed but a small data set is used for the analysis, the bias might not be small.

These points suggest that the bias of an estimator can be interpreted as a systematic error [8]. In addition to a small bias, the data transformed by the estimator should be strongly concentrated around the true parameter, i.e., its variance $\sigma^2(\hat{\theta}) = \mathrm{E}[(\hat{\theta} - \mathrm{E}[\hat{\theta}])^2]$ is as small as possible. When an estimator has a small variance one also says that it has a high precision (see Fig. 1).

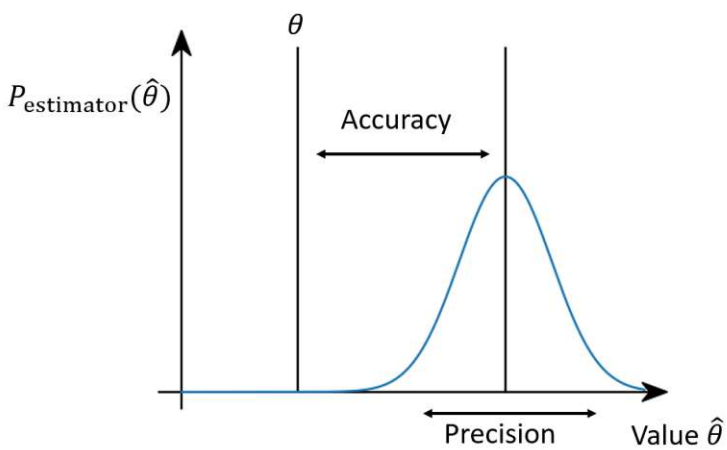

Figure 1: The blue curve depicts the estimator's distribution $P_{\text{estimator}}(\hat{\theta})$. This estimator most likely returns a value close to the peak. The precision is a measure of the uncertainty, as it determines how strongly the estimated value is scattered around the peak. The estimator is clearly biased, i.e., inaccurate, since it systematically predicts values that are different from the actual parameter θ.

2.1.1 The Mean Squared Error Loss Function

In order to quantify the quality of an estimator, a so-called loss function $L[\hat{\theta}, \theta]$ is required. The lower the loss, the better the estimator, thus the construction of an estimator can be formulated as an optimization problem $L[\hat{\theta}, \theta] \rightarrow min$. The choice of the loss depends on the estimation problem at hand. The Mean Squared Error (MSE) [13] $MSE = \mathrm{E}[(\hat{\theta} - \theta)^2]$, for example, is deployed in a large variety of problems. Using the linearity of the expectation

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

and the fact that θ is a constant ($\mathrm{E}[\theta] = \theta$) one can show that the MSE splits into a variance and a bias term

$$MSE = \sigma^2(\hat{\theta}) + b(\hat{\theta}, \theta)^2 \ , \tag{1}$$

which is small when the estimator is both accurate and precise. For unbiased estimators ($b = 0$), minimizing the MSE is equivalent to minimizing the variance, i.e., out of two unbiased estimators the one with the higher precision has a smaller MSE. The unbiased estimator with the smallest possible variance is referred to as the Minimum Variance Unbiased Estimator (MVUE). The MVUE is not necessarily the one with the smallest loss since the presence of a finite bias term can allow for the variance to be even smaller. The fact that both terms need to be taken into account when evaluating the MSE is known as the bias variance trade-off. In Sec. 2.1.5 it is shown that the variance of an estimator is bounded from below. Since the bias term b^2 in the MSE cannot be negative, the lower bound on the variance is a lower bound on the total MSE as well. Evaluating the bound of the variance involves the computation of the so-called Fisher Information, which is introduced in the following section.

2.1.2 The Fisher Information

A central quantity in estimation theory is the Fisher Information (FI). Given a parametric family of distributions $P(\theta)$ with the Probability Density Function (PDF) $p(\boldsymbol{x}|\theta)$, the FI

$$I(\theta) := \int d\boldsymbol{x}\, p(\boldsymbol{x}|\theta) \left(\frac{\partial_\theta p(\boldsymbol{x}|\theta)}{p(\boldsymbol{x}|\theta)} \right)^2 = \mathrm{E}\left[\left(\frac{\partial_\theta p(\boldsymbol{x}|\theta)}{p(\boldsymbol{x}|\theta)} \right)^2 \right] \tag{2}$$

determines how sensitively the distribution depends on the parameter θ on average. The PDF $p(\boldsymbol{x}|\theta)$ gives the probability (or "likelihood") of observing the data $\boldsymbol{x}$ for a given parameter value θ, it is therefore also called the likelihood function. The expression

$$S(\theta) = \left(\frac{\partial_\theta p(\boldsymbol{x}|\theta)}{p(\boldsymbol{x}|\theta)} \right) = (\partial_\theta \log(p(\boldsymbol{x}|\theta))) \tag{3}$$

is known as the score function of the likelihood $p(\boldsymbol{x}|\theta)$, measuring the slope of the log likelihood function.
When calculating the FI, the following regularity conditions [8] must hold:

- The variable $\boldsymbol{x}$ that enters the PDF $p(\boldsymbol{x}|\theta)$ is defined in a region (its support) that does not depend on the parameter θ. E.g., for a normal distribution where the support stretches from minus infinity to plus infinity, this condition is fulfilled. The condition does not hold, e.g., for a uniform distribution where the estimated parameter might be the mean or the variance, as the region where the density is non-zero depends on these parameters. The Fisher Information is thus ill-defined in that case.

- As calculating the FI involves evaluating $\partial_\theta p(\boldsymbol{x}|\theta)$, this derivative needs to exist. In particular, $p(\boldsymbol{x}|\theta)$ must be differentiable at least up to third order in θ [8].

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

Various mathematical and descriptive properties of the FI are introduced in Sec. A.3.
Although the focus of this thesis lies on estimating a single parameter, we describe how the concept of FI is generalized to an arbitrary amount of parameters, or equivalently, a vector valued parameter with entries θ_i , $i = 1, ..., K$. The FI then takes the form of a $K \times K$ matrix. Each entry is related to a pair of parameters

$$I_{i,j}(\theta) := \int d\boldsymbol{x}\, p(\boldsymbol{x}|\theta) \frac{\partial_{\theta_i} p(\boldsymbol{x}|\theta) \partial_{\theta_j} p(\boldsymbol{x}|\theta)}{p(\boldsymbol{x}|\theta)^2} = \mathrm{E}\left[\frac{\partial_{\theta_i} p(\boldsymbol{x}|\theta) \partial_{\theta_j} p(\boldsymbol{x}|\theta)}{p(\boldsymbol{x}|\theta)^2}\right] . \tag{4}$$

Equation (4) defines the so called Fisher information matrix (FIM).

2.1.3 Example: Fisher Information for Normal Distribution

We present the FIM $I_{i,j}$ for the general case, where the random variable $\boldsymbol{x}$ follows a multivariate normal distribution $\boldsymbol{x} \sim N(\boldsymbol{\mu}(\theta), \boldsymbol{\Sigma}(\theta))$. $\boldsymbol{\mu}(\theta)$ is the mean of the distribution and $\boldsymbol{\Sigma}(\theta)$ the covariance matrix, which is a generalization of the variance to multiple dimensions. The PDF reads

$$p(\boldsymbol{x}|\boldsymbol{\mu}(\theta), \boldsymbol{\Sigma}(\theta)) = \frac{1}{\sqrt{\det(2\pi\boldsymbol{\Sigma})}} \exp\left(-\frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu})^T \boldsymbol{\Sigma}^{-1} (\boldsymbol{x}-\boldsymbol{\mu})\right) , \tag{5}$$

which is uniquely determined by the mean $\boldsymbol{\mu}$ and the covariance matrix $\boldsymbol{\Sigma}$. We can find an analytic expression [14] of the FIM for vector parameters θ_i , $i = 1, ..., K$

$$I_{i,j}(\theta) = \left(\partial_{\theta_i} \boldsymbol{\mu}^T\right) \boldsymbol{\Sigma}^{-1} \left(\partial_{\theta_j} \boldsymbol{\mu}\right) + \frac{1}{2} \mathrm{tr}\left(\boldsymbol{\Sigma}^{-1} (\partial_{\theta_i} \boldsymbol{\Sigma}) \boldsymbol{\Sigma}^{-1} (\partial_{\theta_j} \boldsymbol{\Sigma})\right) . \tag{6}$$

The first term in Eq. (6) accounts for the dependence of the mean and the second term for the dependence of the covariance matrix on the parameter. The parameter can then be chosen arbitrarily, for example $\boldsymbol{\Sigma}(\theta) = const$. The FIM is then given by

$$I_{i,j}(\theta) = \left(\partial_{\theta_i} \boldsymbol{\mu}^T\right) \boldsymbol{\Sigma}^{-1} \left(\partial_{\theta_j} \boldsymbol{\mu}\right) , \tag{7}$$

as the trace term in Eq. (6) vanishes. Consider the univariate case where the covariance matrix is reduced to a single number $\boldsymbol{\Sigma} = \sigma^2$. Here, Eq. (7) takes on the simple form

$$I(\theta) = \left(\frac{\partial_\theta \mu(\theta)}{\sigma}\right)^2 . \tag{8}$$

Equation (8) shows that, in this situation, the information is determined by the parameter dependence of $\mu(\theta)$. If $\mu(\theta)$ is independent of θ, the FI vanishes.
It may also be interesting to study a situation where the first term in Eq. (6) vanishes but the trace term is finite. A one dimensional example is $\boldsymbol{\Sigma}(\theta) = \sigma^2(\theta) = \theta$, $\boldsymbol{\mu}(\theta) = const$. This results in

$$I(\theta) = \frac{1}{2\sigma^4} . \tag{9}$$

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

A parametrization that is of particular interest in this work is given by $\boldsymbol{\mu}(\theta) = (\theta, \theta, ..., \theta)^T$ and $\boldsymbol{\Sigma}(\theta) = const.$, where the multivariate Gaussian is determined by the scalar parameter θ and the FIM reduces to a single number. The FI reads

$$I(\theta) = \sum_{i,j} \left(\boldsymbol{\Sigma}^{-1}\right)_{i,j} \ . \tag{10}$$

Compact analytical expressions of the FI cannot be obtained in most cases, when the distribution is non-Gaussian. In general one has to rely on numerical algorithms for approximating the FI, some of which we introduce in Sec. 3. These methods, however, often involve analytic expressions of the FI for very basic distributions, e.g., Gaussians.

2.1.4 The Fisher Information Inequality

In Sec. 5.3 the propagation of Fisher Information through the layers of artificial neural networks is studied. The layers of a network can be regarded as a series of non-linear transformations of the data at hand. Understanding how FI changes under data transformations is crucial for the later chapters of this thesis. Mathematically, this transformation behaviour is given by the so-called Fisher Information Inequality (FII), which states that a transformation cannot increase the FI.
Let

$$\boldsymbol{y} = f(\boldsymbol{x}) \tag{11}$$

be a random variable that is related to the random variable $\boldsymbol{x}$ by an arbitrary deterministic transformation f. When $\boldsymbol{x}$ follows a distribution with the density $p_{\boldsymbol{x}}(\boldsymbol{x}|\theta)$, then $\boldsymbol{y}$ follows a transformed distribution $p_{\boldsymbol{y}}(\boldsymbol{y}|\theta)$ (see Sec. 3.4). The FII states that the FI of the transformed random variable $I_{\boldsymbol{y}}(\theta) = \int d\boldsymbol{y}\, p_{\boldsymbol{y}}(\boldsymbol{y}|\theta) \left(\partial_\theta \log(p_{\boldsymbol{y}}(\boldsymbol{y}|\theta))\right)^2$ cannot be larger than the FI $I_{\boldsymbol{x}}(\theta) = \int d\boldsymbol{x}\, p_{\boldsymbol{x}}(\boldsymbol{x}|\theta) \left(\partial_\theta \log(p_{\boldsymbol{x}}(\boldsymbol{x}|\theta))\right)^2$ of the random variable $\boldsymbol{x}$ itself

$$I_{\boldsymbol{y}}(\theta) \leq I_{\boldsymbol{x}}(\theta) \ . \tag{12}$$

The complicated proof of Eq. (12) is not given in this thesis (see [11] for a detailed derivation). A transformation $f(\boldsymbol{x})$ for which the equality in Eq. (12) holds is referred to as a sufficient statistic[1] of $\boldsymbol{x}$ [10].
For any Random Variable (RV) there exist multiple sufficient statistics, such as the trivial case of f being the identity operation $f(\boldsymbol{x}) = \boldsymbol{x}$, or a linear transformation of the form $f(\boldsymbol{x}) = a \times \boldsymbol{x}$, with some constant a. In Sec. A.4, we show that any bijective transformation yields a sufficient statistic. While finding an arbitrary sufficient statistic is trivial, a non-invertible function that preserves the FI can be difficult to construct. In this thesis, we show

[1] R. A. Fisher defined sufficiency as "A statistic satisfies the criterion of sufficiency when no other statistic which can be calculated from the same sample provides any additional information as to the value of the parameter to be estimated" [10]. Although this seems to be equivalent to Eq. (12) becoming an equality, it is possible to construct counterexamples of insufficient statistics that preserve the FI [15]. Within the scope of this thesis, however, we assume that conservation of FI implies sufficiency.

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

that well-trained artificial neural networks automatically establish a non-invertible, close-to-sufficient statistic, even when this is not the primary goal in the training procedure (see Sec. 5).

2.1.5 The Cramér-Rao Lower Bound

The importance of Fisher Information in statistical estimation theory largely stems from its appearance in the Cramér-Rao inequality [16] (CRI). The CRI provides a lower bound, the so-called Cramér-Rao Lower Bound (CRLB), for an unbiased estimator's precision that is independent of the estimator itself. With the notation introduced in Sec.2.1.1 and the FI $I(\theta)$, the CRI is given by

$$\sigma(\hat{\theta})^2 \geq \frac{1}{I(\theta)} \ , \tag{13}$$

i.e., the variance of an unbiased estimator $\hat{\theta}$ is always larger than the inverse FI. The right hand side of Eq. (13) is independent of $\hat{\theta}$, which means that the bound holds for arbitrary choices of the unbiased estimator. This inequality is based on the following assumptions:

- The Fisher information is well defined. Therefore the conditions listed in Sec. 2.1.2 are expected to hold.
- The estimator is unbiased for all values of the parameter $b(\hat{\theta}, \theta) = 0$.

The last condition is not a fundamental limitation but instead allows for a modified version of the CRI [2]

$$\sigma(\hat{\theta})^2 \geq \frac{(\partial_\theta \mu(\hat{\theta}))^2}{I(\theta)} \ , \tag{14}$$

with $\mu(\hat{\theta}) := \mathrm{E}[\hat{\theta}]$ being the expectation value of $\hat{\theta}$. A proof of Eq. (13) and Eq. (14) is given in [16]. Note that $\mu(\hat{\theta})$ is related to the estimator's bias via

$$b(\hat{\theta}, \theta) = \mu(\hat{\theta}) - \theta \ . \tag{15}$$

Equation (14) allows for variances smaller than the inverse Fisher information. This however, requires a biased estimator, which is inaccurate, possibly even in the limit of infinitely large samples. As a trivial example of an estimator that achieves a variance below the inverse FI (i.e., $\sigma^2 < 1/I$), consider a function that maps every input $\boldsymbol{x}$ to the same output $\hat{\theta}(\boldsymbol{x}) = \theta_0 = const$. The variance of this estimator vanishes. Thus also the right hand side in Eq. (14) must be equal to 0, which holds since $\partial_\theta \mu(\hat{\theta}) = \partial_\theta \mu(\theta_0) = 0$.

The (bias corrected) CRI can be generalized to multiple parameters, i.e., vector valued $\boldsymbol{\theta}$. Recall that this results in the FI being a matrix (Eq. (4)), rather than a single number. The variance in Eq. (14) needs to be replaced by the covariance matrix $\mathrm{Cov}(\hat{\boldsymbol{\theta}})$ and the derivative

[2] In the literature the term CRI often refers to Eq. (13). In the remainder of this thesis we define the CRI as Eq. (14).

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

of the expected parameter $\partial_{\boldsymbol{\theta}}\boldsymbol{\mu}(\hat{\boldsymbol{\theta}})$ has the shape of a matrix as well, since the gradient $\partial_{\boldsymbol{\theta}}$ is a vector operation and $\boldsymbol{\mu}(\hat{\boldsymbol{\theta}})$ is vector valued as well. The Cramér-Rao inequality in the multi-parameter case [8] is given by

$$\mathrm{Cov}(\hat{\boldsymbol{\theta}}) \geq \left(\partial_{\boldsymbol{\theta}}\boldsymbol{\mu}(\hat{\boldsymbol{\theta}})\right) I(\boldsymbol{\theta})^{-1} \left(\partial_{\boldsymbol{\theta}}\boldsymbol{\mu}(\hat{\theta})\right)^T . \tag{16}$$

For unbiased estimators Eq. (16) reduces to

$$\mathrm{Cov}(\hat{\boldsymbol{\theta}}) \geq I(\boldsymbol{\theta})^{-1} , \tag{17}$$

thus the covariance matrix of the estimator cannot be smaller[3] than the inverse FIM.

2.1.6 The Exponential Family and Efficient Estimators

The Cramér-Rao inequality determines how precise an estimator can possibly be. The question arises, whether an estimator can be found that allows for equality in Eq. (16). It turns out that the answer to this question is in general no [8, 16], although it is possible for some distributions. The classes of distributions for which the equality can be achieved are the so called "exponential families" and the corresponding estimators are referred to as "efficient estimators". Exponential families have a probability density of the form [8]

$$p(\boldsymbol{x}|\theta) = \boldsymbol{A}(\theta)\boldsymbol{B}(\boldsymbol{x}) \exp(\boldsymbol{C}^T(\theta)\boldsymbol{D}(\boldsymbol{x})) . \tag{18}$$

The term "family" refers to the dependence on the parameter, i.e., for distinct values of θ_1 and θ_2, $p(\boldsymbol{x}|\theta_1)$ and $p(\boldsymbol{x}|\theta_2)$ are in general different distributions. In Eq. (18), $\boldsymbol{A}(\theta)$ and $\boldsymbol{B}(\boldsymbol{x})$ are scalars while $\boldsymbol{C}(\theta)$ and $\boldsymbol{D}(\boldsymbol{x})$ are vector valued quantities. Note that $\boldsymbol{A}(\theta)$ and $\boldsymbol{C}(\theta)$ depend only on the parameter θ but not on the variable $\boldsymbol{x}$ while $\boldsymbol{B}(\boldsymbol{x})$ and $\boldsymbol{D}(\boldsymbol{x})$ depend only on $\boldsymbol{x}$ but not on θ. A standard example of an exponential family is the multivariate normal distribution with the density [8, 17]

$$p(\boldsymbol{x}|\boldsymbol{\mu}(\theta), \boldsymbol{\Sigma}) = \frac{1}{\sqrt{\det(2\pi\boldsymbol{\Sigma})}} \exp\left(-\frac{1}{2}(\boldsymbol{x} - \boldsymbol{\mu}(\theta))^T \boldsymbol{\Sigma}^{-1}(\boldsymbol{x} - \boldsymbol{\mu}(\theta))\right) , \tag{19}$$

where we assume for the sake of simplicity that the covariance matrix $\boldsymbol{\Sigma}$ is independent of the parameter θ. This density function can be brought to the form of Eq. (18) by choosing [8]

$$\boldsymbol{A}(\theta) = \frac{1}{\sqrt{\det(2\pi\boldsymbol{\Sigma})}} \exp\left(-\frac{1}{2}\boldsymbol{\mu}(\theta)^T \boldsymbol{\Sigma}^{-1}\boldsymbol{\mu}(\theta)\right) , \tag{20}$$

$$\boldsymbol{B}(\boldsymbol{x}) = \exp\left(-\frac{1}{2}\boldsymbol{x}^T \boldsymbol{\Sigma}^{-1}\boldsymbol{x}\right) , \tag{21}$$

[3] The inequality in Eq. (16) and Eq. (17) does not hold for each entry of the matrices. For two matrices A and B, we denote with the inequality $A \geq B$ that the difference $A - B$ is positive semidefinite [8].

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

$$\boldsymbol{C}(\theta) = \boldsymbol{\Sigma}^{-1}\boldsymbol{\mu}(\theta) \tag{22}$$

and

$$\boldsymbol{D}(\boldsymbol{x}) = \boldsymbol{x} \ . \tag{23}$$

Other examples [8] of distributions that form exponential families, given a suitable parametrization, are the Poisson distribution, the gamma distribution and the multinomial distribution.

In the following we elaborate on how to find the best possible estimator for exponential family distributions, such that its precision exactly matches the Cramér-Rao bound. It can be shown that such an efficient estimator is necessarily unbiased [8]. The condition for an estimator to be efficient reads

$$\partial_\theta \log(p(\boldsymbol{x}|\theta)) = I(\theta)\left(\tau(\boldsymbol{x}) - \theta\right) \ , \tag{24}$$

where $\tau(\boldsymbol{x})$ is the estimator $\hat{\theta}$ evaluated at $\boldsymbol{x}$ [4]. Solving this equation for $\tau(\boldsymbol{x})$ directly yields an expression for an efficient estimator (if it exists)

$$\tau(\boldsymbol{x}) = \frac{\partial_\theta \log(p(\boldsymbol{x}|\theta))}{I(\theta)} + \theta \ . \tag{25}$$

Finding an efficient estimator with Eq. (25) requires that the density function is known explicitly. Note that the left hand side of Eq. (25) is independent of θ itself, hence the right hand side must be too. For an estimator to be efficient its functional form $\tau(\boldsymbol{x})$ must be independent of the estimated parameter. As a measure of how close an estimator comes to achieving the equality sign in the CRI we introduce the efficiency ϵ

$$\epsilon = \frac{\left(\partial_\theta \mu_{\text{estimator}}(\hat{\theta})\right)^2}{\sigma^2_{\text{estimator}}(\hat{\theta}) I(\theta)} \ , \ 0 \leq \epsilon \leq 1 \ , \tag{26}$$

which is equal to one for an efficient estimator. In the literature the term efficiency is often defined with respect to a specific loss function, however, our definition will be convenient in the later sections of this thesis, where we analyse whether estimators that are represented by neural networks achieve precisions close to the Cramér-Rao bound.

2.2 Regression

To this point, we dealt with the problem of estimating quantities that enter the underlying probability distribution of the data as parameters. In many applications of statistics one has to deal with a task that is similar but not equivalent. Given a finite data set, the aim

[4] We use this notation to highlight the difference between the estimator $\hat{\theta}$ which is a function of the data and the transformation $\tau(\boldsymbol{x})$ which is the same function evaluated at the continuous variable $\boldsymbol{x}$.

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

of regression is to model a functional relation between the random variables that enter the data distribution. The difference to estimation problems lies in the way the data is at hand[5]:

- Parameter estimation problems: The data consists of multiple data sets, each corresponding to a specific parameter value. The data is modelled as a parametric family of probability densities which allow for an analysis according to Sec. 2.1. When the distribution is unknown[6], it is necessary to have enough data available for several different parameter values, such that the parametric density can be modelled with sufficient accuracy.

- Regression problems: If it is not possible to assign a sufficiently large subset of the data to several values of the parameter, either because there is not enough data available or if the data is wide-spread along the parameter's direction, it might not be feasible to model the distribution with a parametric family of densities. Rather than being a parameter in the estimation theoretic sense, the quantity of interest should then be considered as a component of the random vector. Hence, this quantity is of stochastic nature.

Given a data set of the size N

$$D = ((\boldsymbol{x}_1, \boldsymbol{y}_1), (\boldsymbol{x}_2, \boldsymbol{y}_2), ..., (\boldsymbol{x}_N, \boldsymbol{y}_N)) \ , \tag{27}$$

a basic regression problem [18, 19] requires to find the functional dependence

$$\boldsymbol{y}(\boldsymbol{x}) = f(\boldsymbol{x}) + \boldsymbol{\epsilon} \ , \tag{28}$$

between the random variables $\boldsymbol{x}$ and $\boldsymbol{y}$. Such an ansatz is called an "additive model" [19] since the noise term $\boldsymbol{\epsilon}$ is added to the fit function $f(\boldsymbol{x})$. A large class of problems can be solved with this assumption. The left hand side $\boldsymbol{y}$ is referred to as the "dependent" random variable, which is modelled by the so called "independent" random variable $\boldsymbol{x}$. The dependent random variable plays a role similar to the parameter in an estimation task. We seek for a model that approximates this relation such that a suitable loss function is minimized. The noise term should vanish on average hence a good fit function $f(\boldsymbol{x})$ does not resolve these fluctuations. Modelling this noise term is a common example of overfitting, which is described in more detail in Sec. 4.4. A basic technique for solving such a problem is linear regression, where a linear ansatz $f(\boldsymbol{x}) = \boldsymbol{K}\,\boldsymbol{x} + \boldsymbol{d}$ is made and the fit parameters $\boldsymbol{K}$ and $\boldsymbol{d}$ are calculated via the method of least squares [18]. When a more elaborate functional connection $\boldsymbol{y} = f(\boldsymbol{x})$ is required to achieve good results, artificial neural networks are convenient tools, as they are designed to be able to approximate arbitrarily complex non-linear functions. Machine learning models particularly excel when $\boldsymbol{x}$ is a very high dimensional random vector.

[5] Our notion of the terms "estimation" and "regression" introduced here is not the standard in the literature but convenient for the purpose of analysing machine learning models.

[6] In practice, a model of the distribution is often available beforehand, however, in this thesis we want to apply concepts from estimation theory to data sets that follow unknown distributions.

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

2.2.1 Estimation of Random Parameters: The Connection between Parameter Estimation and Regression

In this work we primarily focus on parameter estimation problems using artificial neural networks. However, general regression tasks that cannot easily be modelled as parameter estimation tasks, are heavily used in machine learning. We therefore demonstrate a way of connecting our notion of estimation and regression problems, as it was introduced in Sec. 2.2, by making use of Bayesian statistics [12].

As described in the previous section, when there is no analytical model of the data distribution available and there is not enough data that can be assigned to different parameter values, the quantity of interest should be modelled as a random variable instead of a parameter. Mathematically, this results in a joint distribution determined by the density $p_{\boldsymbol{x},\theta}(\boldsymbol{x},\theta)$ instead of a conditional density $p_{\boldsymbol{x}|\theta}(\boldsymbol{x}|\theta)$. Given a joint density, the conditional density is defined as

$$p_{\boldsymbol{x}|\theta}(\boldsymbol{x}|\theta) = \frac{p_{\boldsymbol{x},\theta}(\boldsymbol{x},\theta)}{p_\theta(\theta)} \ , \tag{29}$$

with the marginal distribution

$$p_\theta(\theta) = \int d\boldsymbol{x}\, p_{\boldsymbol{x},\theta}(\boldsymbol{x},\theta) \ . \tag{30}$$

Equation (29) states that when interpreting θ as a parameter instead of a random variable, normalization is required. The normalizing function is exactly the marginal distribution of the random variable θ. From a Bayesian perspective of statistics this is called the prior distribution, which itself carries information about the parameter. For example, if $p_\theta(\theta)$ is a distribution peaked around some value θ_0, data drawn from the joint distribution would also have a sharp peak around θ_0 although the conditional $p_{\boldsymbol{x}|\theta}(\boldsymbol{x}|\theta)$ may not show that feature. Informally, one can think of the joint distribution having more information content than the conditional, since the derivative of its log likelihood, which appears in the calculation of the Fisher information, picks up an additional term

$$\partial_\theta \log(p_{\boldsymbol{x},\theta}(\boldsymbol{x},\theta)) = \partial_\theta \log(p_{\boldsymbol{x}|\theta}(\boldsymbol{x}|\theta)) + \partial_\theta \log(p_\theta(\theta)) \tag{31}$$

determined by the prior distribution. It is possible to extend the notion of Fisher information to Bayesian statistics [12] by adding a term which we will refer to as the prior information I_θ

$$I_{\text{Bayesian}} = \int d\theta\, p_\theta(\theta) I(\theta) + I_\theta \ , \tag{32}$$

with

$$I_\theta = \int d\theta \frac{(\partial_\theta p_\theta(\theta))^2}{p_\theta(\theta)} \ . \tag{33}$$

The prior information is defined in an analogous manner to the FI but the integration is performed over the parameter itself, so I_θ is independent of θ. We refer to I_{Bayesian} as the

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

Bayesian Fisher information (BFI). The conventional FI $I(\theta)$ is replaced by its expectation value with respect to the prior distribution

$$\mathrm{E}_{\theta\sim p_\theta(\theta)}\left[I(\theta)\right] = \int d\theta\, p_\theta(\theta) I(\theta)\ . \tag{34}$$

In Bayesian statistics instead of the Cramér-Rao bound which is a lower bound for an estimator's variance as a function of the parameter, the inverse BFI directly yields a lower bound for the expected mean squared error

$$\mathrm{E}_{\theta\sim p_\theta(\theta)}\left[MSE(\hat{\theta})\right] = \mathrm{E}_{\theta\sim p_\theta(\theta)}\left[\sigma^2(\hat{\theta}) + b^2(\hat{\theta})\right] \geq \frac{1}{\int d\theta\, p_\theta(\theta) I(\theta) + I_\theta}\ . \tag{35}$$

Equation (35) is the so called Van Trees Inequality (VTI). Mind that unlike the CRI, the VTI is not locally defined for each parameter value, as the MSE is averaged over θ. An advantage of the Bayesian approach is, however, that the right hand side of Eq. (35) is independent of the estimator's bias, while the CRLB for biased estimators $(\partial_\theta \mu(\hat{\theta}))^2/I(\theta)$ explicitly depends on the bias.

In practice it is often difficult to estimate the prior information I_θ, since it is very sensitive to the shape of the prior density $p_\theta(\theta)$, which itself is hard to model precisely. As an example of a situation where the Van Trees inequality might not be a good predictor of the best possible performance of an estimator, consider a data set that is clearly confined within a region $\theta \in [\theta_1, \theta_2]$. A suitable prior density needs to fall off quickly for θ values outside of this region. If every θ inside the region is approximately equally likely, a uniform distribution $\theta \sim U(\theta_1, \theta_2)$ appears to be the best choice. When inserting the corresponding probability density $p_\theta(\theta)$ into the definition of the prior information, we find that Eq. (33) diverges. This paradox is resolved by assuming that the support of θ is not the entire real axis but coincides with the region $\theta \in [\theta_1, \theta_2]$. As in this situation the prior distribution is a constant along the whole support, $p_\theta(\theta)$ contains no information about the parameter $I_\theta = 0$. Alternatively, one could choose a prior that falls off quickly but in a smooth manner, which gives rise to arbitrary values for the prior information depending on how quickly the prior density approaches zero.

In general, if the data can be well approximated by different priors with prior pieces of information that strongly deviate from each other, the Bayesian approach should only be used with caution.

2.3 Hypothesis Testing and Decisions

The discrete analogue of parameter estimation, where observations are mapped to a finite amount of events instead of a continuous parameter, is referred to as a decision problem [12]. Although the fields of estimation and decision show various similarities, the mathematical foundations such as the FI, the CRI and efficiency of estimators cannot be directly applied to the discrete case. In the most basic situation of decision, one is confronted with an ex-

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

periment where two distinct outcomes are possible and applies some mathematical rule to decide for one of the outcomes based on the measured data. This can be formulated as a problem of hypothesis testing [12]:
Given a data set $D = (\boldsymbol{x}_1, \boldsymbol{x}_2, ..., \boldsymbol{x}_N)$ of size N and the two distributions P_0 and P_1 corresponding to the two distinct hypothesis H_0 and H_1, the aim is to decide whether the data is better described as a sample of P_0 or of P_1. H_0, what we call the null hypothesis, is tested against the alternative hypothesis H_1. The aim of hypothesis testing is to find a rule for deciding which hypothesis shall be regarded as true, i.e., a convention of comparing the probabilities P_0 and P_1 based on the measured data. If one wants to be relatively certain that the assumption H_0 does indeed hold, a threshold γ is introduced such that H_0 is considered to be confirmed when the probability of the data being distributed according to P_0 is above γ. In this work, we are mostly interested in the symmetric situation where we want to decide which distribution is a better match for the data, without making an assumption beforehand. This approach is relevant in machine learning tasks where the aim is to assign one out of two or more attributes to some input data (see Sec. 6).

2.3.1 Likelihood Ratio Test and Bayes Error

Similar to the FI that bounds the best achievable precision of an estimator, we now seek for a quantity that measures how well a hypothesis test could theoretically perform. This is achieved by quantifying the probability of making an error with the hypothesis test [12], i.e., a wrong decision. We speak of an error if either the hypothesis H_0 is true but the test results in H_1 being accepted, or if the reverse situation occurs, i.e., H_1 is true but H_0 is accepted. The total probability of making an error is given by

$$P_{\text{err}} = P(\text{ accept } H_1 | H_0) + P(\text{ accept } H_0 | H_1) \ . \tag{36}$$

The goal is to find a lower bound for the error probability P_{err}. It can be shown [12] that the ideal test for which the probability of making an error takes on a minimal value among any other testing procedure is the so called likelihood ratio test (LRT). Note the similarity between the LRT from hypothesis testing and the efficient estimator from parameter estimation. Both are considered optimal within their field. The LRT is defined in the following way:
The probabilities P_0 and P_1 are given in terms of their likelihoods, which in the continuous case coincide with the PDFs $p_0(\boldsymbol{x})$ and $p_1(\boldsymbol{x})$, the likelihood ratio is defined as

$$r = \frac{p_1(\boldsymbol{x})}{p_0(\boldsymbol{x})} \ . \tag{37}$$

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

The rule for deciding which hypothesis is accepted is

$$LRT \mathrel{\widehat{=}} \begin{cases} \text{pick } H_0, & r \leq 1 \\ \text{pick } H_1, & r > 1 \,. \end{cases} \tag{38}$$

With the LRT we can calculate the minimal value for the error probability P_{err} among all tests. The probability of accepting H_1 when H_0 is true is given by the area under the density $p_0(\boldsymbol{x})$ in the region where $r > 1$ and equivalently, the probability of accepting H_0 when H_1 is true corresponds to the area under $p_1(\boldsymbol{x})$ in the region where $r \leq 1$. Mathematically, this reads

$$B_{\text{err}} = \int_{r>1} d\boldsymbol{x}\, p_{H_0}(\boldsymbol{x}) + \int_{r\leq 1} d\boldsymbol{x}\, p_{H_1}(\boldsymbol{x}) = \int d\boldsymbol{x}\, \min\left\{p_0\left(\boldsymbol{x}\right), p_1\left(\boldsymbol{x}\right)\right\} \,, \tag{39}$$

which we refer to as the Bayes error (BE). A geometric interpretation of the BE is given in Fig. 2

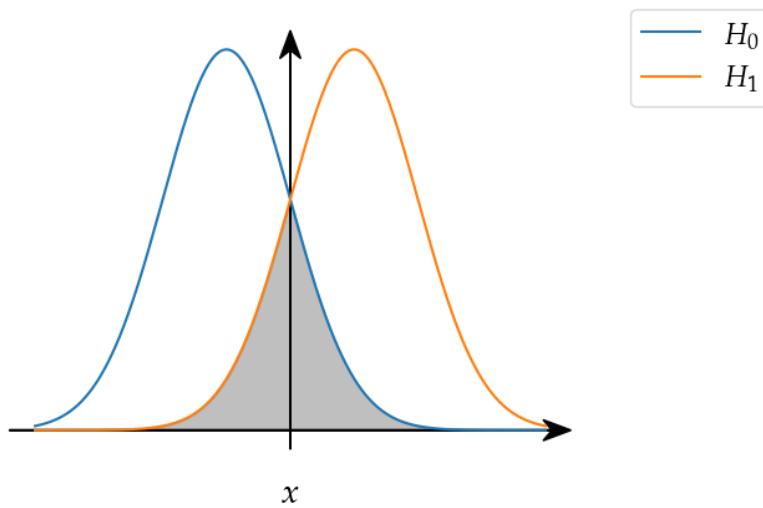

Figure 2: The blue and the orange function are Gaussian PDFs, each corresponding to one of the hypotheses H_0 and H_1. The BE equates to the gray area.

As the LRT is an optimal decision rule for a binary decision problem, the likelihood functions contain all the information that is required to construct such a test. In contrast to the efficient estimators from estimation theory that only exist for exponential families of densities (see Sec. 2.1.6), the LRT test can be constructed whenever the likelihood functions are available. In practice, however, analytic expressions for the likelihoods are often unknown and difficult to estimate. In this work, the LRT is not used directly for hypothesis testing but rather for evaluating a test's performance. In Sec. 6 we use artificial neural networks to construct tests that do not require knowledge about the likelihoods and compare them with the LRT.

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

2.3.2 An Inequality for the Bayes Error

In analogy to the Fisher information inequality from Sec. 2.1.4, it can be shown that the Bayes error obeys a similar rule, that is, it cannot decrease when the data is transformed. Consider a deterministic transformation f that maps the random vector $\boldsymbol{x}$ to a new random vector $\boldsymbol{y}$

$$\boldsymbol{y} = f(\boldsymbol{x}) \ . \tag{40}$$

The initial probability densities describing the data at hand $p_0(\boldsymbol{x})$ and $p_1(\boldsymbol{x})$, respectively, are transformed accordingly, denoted by $p_0(\boldsymbol{y})$ and $p_1(\boldsymbol{y})$. The inequality for the Bayes error B_{err} reads

$$B_{\text{err},\boldsymbol{y}} \geq B_{\text{err},\boldsymbol{x}} \ , \tag{41}$$

where the indices $\boldsymbol{x}$ and $\boldsymbol{y}$ denote that the bound is calculated based on the densities $p_i(\boldsymbol{x})$ and $p_i(\boldsymbol{y})$ $(i = 0, 1)$, respectively. In analogy to the FI, the equality sign in Eq. 41 is achieved when the transformation f is bijective (see Sec. D.1). For the proof of Eq. 41 see Sec. D.2.

3 Calculation of the Fisher Information

In order to give an estimation theoretic description of artificial neural networks (see Sec. 5), we need to be able to approximate the FI of arbitrarily complicated distributions sufficiently well. Therefore, this section concerns with introducing practicable methods of calculating the FI, as evaluating the FI according to Eq. (2) is a non-trivial task. Some reasons for this intricacy are:

- When the dimension of the random variable $\boldsymbol{x}$ is large, the integral in Eq. (2) can be hard to evaluate. Approximating the FI as a Riemann integral on a discrete grid might not be computationally feasible in high dimensions, thus $I(\theta)$ needs to be estimated, e.g., by Monte Carlo integration, which is further described in Sec. 3.1. This means that the expectation in Eq. (2) is replaced by a sum over a finite sample. Due to this approximation, the calculated Fisher information will have noise.

- The PDF $p(\boldsymbol{x}|\theta)$ is often unknown and difficult to estimate. The numerical evaluation of the derivative ∂_θ requires that the density is known with high accuracy. A rough estimate of the density is therefore insufficient.

3.1 Estimation of the Fisher Information with a Monte Carlo Method

We give a brief introduction to Monte Carlo (MC) integration and demonstrate how it is used to estimate the Fisher information when the dimensionality of the data is large. In that case, the conventional Riemann integration, which is often approximated by placing a regular grid along the coordinate axes, exceeds the computational capacities. A regular grid

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

in an N dimensional cartesian coordinate system with M grid points per axis results in a total of M^N grid points. Recall that the FI is calculated as

$$I(\theta) = \int d\boldsymbol{x} \, \frac{(\partial_\theta p(\boldsymbol{x}|\theta))^2}{p(\boldsymbol{x}|\theta)} \, . \tag{42}$$

The function of interest $g(\boldsymbol{x}, \theta) := \frac{(\partial_\theta p(\boldsymbol{x}|\theta))^2}{p(\boldsymbol{x}|\theta)}$ is then evaluated at all of these points.
In Monte Carlo integration, the integral over $g(\boldsymbol{x}, \theta)$ is replaced by an expectation value. This is achieved by rewriting

$$\begin{aligned} I(\theta) &= \int d\boldsymbol{x} \, g(\boldsymbol{x}, \theta) = \int d\boldsymbol{x} \, p(\boldsymbol{x}|\theta) \frac{g(\boldsymbol{x}, \theta)}{p(\boldsymbol{x}|\theta)} \\ &= \mathrm{E}_{\boldsymbol{x} \sim p(\boldsymbol{x}|\theta)} \left[\frac{g(\boldsymbol{x}, \theta)}{p(\boldsymbol{x}|\theta)} \right] = \mathrm{E}_{\boldsymbol{x} \sim p(\boldsymbol{x}|\theta)} \left[\left(\frac{\partial_\theta p(\boldsymbol{x}|\theta)}{p(\boldsymbol{x}|\theta)} \right)^2 \right] = \mathrm{E}_{\boldsymbol{x} \sim p(\boldsymbol{x}|\theta)} \left[S^2(\boldsymbol{x}, \theta) \right] \, , \end{aligned} \tag{43}$$

with the score function $S(\boldsymbol{x}, \theta)$ that was introduced in Eq. (3). It is common practice to estimate expectation values instead of calculating them exactly. An estimator of the expectation value of $S^2(\boldsymbol{x}, \theta)$ is the sample average

$$\mathrm{E}_{\boldsymbol{x} \sim p(\boldsymbol{x}|\theta)} \left[S^2(\boldsymbol{x}, \theta) \right] \approx \frac{1}{N} \sum_{i=1}^{N} S^2(\boldsymbol{x}_i, \theta) \, . \tag{44}$$

For finite N this approximation of the FI, as a function of a random vector, is a random quantity itself, hence there are irreducible random fluctuations. This thesis mostly deals with synthetic data sets where an unlimited amount of samples can be generated, thus these fluctuations can be kept minimal.
Evaluating Eq. (44) requires that the generated samples $\boldsymbol{x}_i$ are distributed according to $p(\boldsymbol{x}|\theta)$. In general, this is a non-trivial task that can be approached, e.g., by a Markov Chain Monte Carlo (MCMC) method, if the density $p(\boldsymbol{x}|\theta)$ is known. In this work we are interested in the reverse situation, where a large sample (the data set) is available but the density is unknown. In this situation, any expectation value of the form

$$\mathrm{E}_{\boldsymbol{x} \sim p(\boldsymbol{x}|\theta)} \left[\Psi(\boldsymbol{x}, \theta) \right] \approx \frac{1}{N} \sum_{i=1}^{N} \Psi(\boldsymbol{x}_i, \theta) \tag{45}$$

can be evaluated if $\Psi(\boldsymbol{x})$ is a function that is known exactly. The score function $S(\boldsymbol{x}, \theta)$, however, depends explicitly on the unknown density $p(\boldsymbol{x}|\theta)$. This problem is tackled in the next section that deals with the estimation of density functions from finite data sets.

3.2 Density Estimation

Various methods have been developed for estimating a PDF from a finite sample of a probability distribution. Given a data set $D = (\boldsymbol{x}_1, \boldsymbol{x}_2, ..., \boldsymbol{x}_N)$ of size N, we assume that the

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

samples are drawn from an unknown probability distribution with the corresponding density $p(\boldsymbol{x})$. This problem is generally difficult and density estimation is an active field of research. Relatively accurate methods of estimating $p(\boldsymbol{x})$ exist in low dimensions, whereas very high dimensional random vectors are often extremely difficult to predict, even if a lot of data is available. One differentiates between parametric and non-parametric density estimation. Methods of the former type require making assumptions about the particular shape of the distribution that is estimated, which could for instance be motivated by physical insights. This distribution depends on a collection of unknown parameters, which need to be determined, which justifies the term "parametric". The following two elementary techniques will be useful for the density estimation tasks within the scope of this work.

3.2.1 The Histogram Method

In situations where the dimension of the random vector $\boldsymbol{x}$ is small (i.e., 3 dimensions or less within the scope of this thesis) and the data set is sufficiently large, the most basic way of estimating the PDF is to make use of the histogram method. Density estimation with histograms is an example of a non-parametric tool for density estimation. The continuous density function is approximated by a discrete histogram as the domain of the variable $\boldsymbol{x}$ is decomposed into distinct hypercubes. The histogram density assigns a value to each hypercube, that is meant to be a good approximation of the true density in that region. This value is calculated by counting the number of sample points that lie within the hypercube and dividing it by the sample size N. The histogram method fails in high dimensionality since the number of hypercubes $N_{\text{hypercubes}}$ scales exponentially with the dimension. When the number of hypercubes exceeds the size of the data set N, most cubes will be empty, which means that a density of zero is assigned in these regions. The condition

$$N \gg N_{\text{hypercubes}} \tag{46}$$

must be fulfilled for the histogram method to yield meaningful results. In addition to Eq. (46) the size of the hypercubes needs to be sufficiently small, such that the density is approximated with a good resolution. This results in a trade-off between resolution and accuracy.

3.2.2 The Maximum Likelihood Method

An example of a parametric density estimation protocol is the so-called Maximum Likelihood Estimation (MLE). This technique turns out to be useful when in addition to the data some attributes of the density function are known. For instance, real world data is often normally distributed, at least to a good approximation. As a (multivariate) normal distribution is determined by its mean and covariance matrix, estimating these parameters yields all the necessary information about the density. In Sec. 3.5 we demonstrate that fitting a normal distribution is particularly convenient for calculating the FI, even if the data is not normally distributed.

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

The goal of MLE is to estimate a PDF $p(\boldsymbol{x}|\Omega)$ with Ω being the collection of all the parameters the distribution depends on. As an assumption of the shape of $p(\boldsymbol{x}|\Omega)$ is made beforehand, the task is to find an estimate $\hat{\Omega}$ of Ω. The likelihood function of the data set is defined as

$$L(D|\Omega) = \prod_{i=1}^{N} p(\boldsymbol{x}_i|\Omega) \ . \tag{47}$$

As the name suggests, MLE is the process of maximizing the likelihood function with respect to the parameters. $\hat{\Omega}$ calculated this way can be interpreted as the collection of parameters for which the data D is the most likely (it has the highest likelihood). Mathematically, this reads

$$\partial_\Omega L(D|\Omega) = 0 \ . \tag{48}$$

Equation (48) yields a system of equations that in general needs to be solved numerically. The solution $\hat{\Omega}$ might not be unique so in principle one needs to take care of local extrema. In many practical situations, however, the optimization problem at hand is convex. It is common practice to replace the likelihood function by its logarithm

$$l(D|\Omega) \coloneqq \log(L(D|\Omega)) \ , \tag{49}$$

since this operation preserves the location of the maximum and it makes the calculations easier since the product in Eq. (47) is converted into a sum.
We present the results of the MLE of the mean and the covariance matrix of a multivariate normal distribution without proof. Equation (48) can be solved analytically in this case. The maximum likelihood estimator for the mean turns out to be

$$\hat{\boldsymbol{\mu}} = \frac{1}{N} \sum_{i=1}^{N} \boldsymbol{x}_i \ . \tag{50}$$

$\hat{\boldsymbol{\mu}}$ coincides with the sample mean. The estimator for the covariance matrix reads

$$\hat{\boldsymbol{\Sigma}} = \frac{1}{N} \sum_{i=1}^{N} (\boldsymbol{x}_i - \hat{\boldsymbol{\mu}})(\boldsymbol{x}_i - \hat{\boldsymbol{\mu}})^T \ , \tag{51}$$

which depends explicitly on the estimated mean $\hat{\boldsymbol{\mu}}$.

3.2.3 Advanced Methods of Density Estimation

The fact that we can estimate PDFs with histograms, at least in low dimensions, with a precision that is good enough to calculate the Fisher information is due to the availability of a large data set. The artificial neural networks that are analysed within the scope of this thesis are trained on synthetic data that is sampled from a well known distribution. For real world problems, where this might not be the case, more elaborate methods of density

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

estimation are available.

In the low dimensional case, a technique called "kernel density estimation" [20] is often employed. Given a data set $D = (\boldsymbol{x}_1, \boldsymbol{x}_2, ..., \boldsymbol{x}_N)$ of i.i.d. observations $\boldsymbol{x}_i$, the probability density $p(\boldsymbol{x})$ is approximated by

$$p(\boldsymbol{x}) \approx p_N(\boldsymbol{x}) = \frac{1}{Nh} \sum_{i=1}^{N} K\left(\frac{\boldsymbol{x} - \boldsymbol{x}_i}{h}\right) . \tag{52}$$

The function $K(\boldsymbol{x})$ is the so-called kernel, for which, e.g., a Gaussian or a rectangular function is chosen. The kernel is non-negative such that $p_N(\boldsymbol{x})$, as an estimate of a PDF, is non-negative as well. The positive number h needs to be chosen suitably. Small values of h lead to a more accurate estimation when lots of data is available, however, larger values of h typically result in a smoother function $p_N(\boldsymbol{x})$. Elaborate methods exist for the choice of this smoothing parameter [21].

State of the art methods of density estimation often deploy artificial neural networks. One example is the technique "normalizing flows" [22], where the network maps the density function $p(\boldsymbol{x})$ to a (multivariate) Gaussian density. This map is constructed such that it has a unique inverse, thus the density $p(\boldsymbol{x})$ can be reconstructed by inverting the map (see Sec. 3.4.1).

3.3 Calculating the Fisher Information from Estimated Densities

The methods of density estimation enable, at least in principle, the estimation of the FI from a data set. We assume that the observation covers a grid of θ values with a sample $D(\theta) = (\boldsymbol{x}_1(\theta), \boldsymbol{x}_2(\theta), ..., \boldsymbol{x}_{N_\theta}(\theta))$ of size N_θ for each value of the parameter. Depending on the structure of the estimated density, which can, e.g., take on the form of a discrete histogram density or a continuous parametric function, the numerical calculation of the FI is carried out with a suitable algorithm.

3.3.1 Calculating the Fisher Information from a Histogram Distribution

When a good approximation of the densities $p(\boldsymbol{x}|\theta + \Delta\theta)$ and $p(\boldsymbol{x}|\theta - \Delta\theta)$ is available, the FI $I(\theta)$ can be directly calculated with Eq. (2) with the finite difference scheme Eq. (114) introduced in B.1. The terms $1/p(\boldsymbol{x}|\theta)$ and $\log(p(\boldsymbol{x}|\theta))$, that occur in the expression of the FI diverge when the estimated density approaches $p(\boldsymbol{x}|\theta) = 0$. In accurate histogram densities with a large number of hypercubes, some of them might not contain any data points, thus the estimated density in this region vanishes. This problem is mitigated by using the definition Eq. (98) of the FI, which involves only the square root of $p(\boldsymbol{x}|\theta)$.

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

3.3.2 Calculating the Fisher Information with Maximum Likelihood Estimation

The calculation of the FI for densities that are estimated with MLE is explained based on the example of a multivariate normal distribution. If the assumption that the density has a Gaussian shape can be justified, the FI $I(\theta)$ is determined by the mean $\boldsymbol{\mu}(\theta)$ and the covariance matrix $\boldsymbol{\Sigma}(\theta)$ as functions of θ. When θ is a scalar valued parameter, the FI (see Eq. (6)) is given by

$$I(\theta) = (\partial_\theta \boldsymbol{\mu}(\theta))^T \, \boldsymbol{\Sigma}(\theta)^{-1} \partial_\theta \boldsymbol{\mu}(\theta) + \frac{1}{2} \mathrm{tr} \left(\boldsymbol{\Sigma}^{-1}(\partial_\theta \boldsymbol{\Sigma}) \boldsymbol{\Sigma}^{-1}(\partial_\theta \boldsymbol{\Sigma}) \right) \ . \tag{53}$$

For a reasonable estimate of the FI, the functions $\boldsymbol{\mu}(\theta)$ and $\boldsymbol{\Sigma}(\theta)$, which are typically given on a discrete θ-grid, need to be sufficiently smooth such that the derivatives ∂_θ are not dominated by fluctuations. In case these functions have significant noise, it is necessary to replace them by a smooth approximation, e.g., by a non-linear fit or using a filter. Another source of error is the calculation of the inverse covariance matrix $\boldsymbol{\Sigma}^{-1}(\theta)$ whenever eigenvalues of $\boldsymbol{\Sigma}(\theta)$ become very small. This problem is circumvented by a Singular Value Decomposition (SVD), which is further explained in Sec. B.4.

3.4 Transformation of Probability Densities

Since calculating the FI requires that the PDF is known relatively precisely, density estimation in high dimensions should be avoided whenever possible. As an example for such a situation consider a PDF that is unknown and highly complex but related to a well-known PDF via a transformation of the random variable. In Sec. 4 we elaborate on basic deep learning architectures, which commonly implement complicated functions that transfer input data to some output. Such a function is typically given as a composition of linear and non-linear functions. Assuming that the input data is distributed according to some PDF $p(\boldsymbol{x})$, the deep neural network acts as a series of transformations on this density. We study the behaviour of densities under such transformations of the random variable. It turns out that this is in general a hard problem and cannot be solved in many real-world applications. However, often one is not interested in the entire transformed density function but only in the transformed moments. This problem is addressed in Sec. B.2.

3.4.1 Invertible Transformations

An informal[7] explanation of the behaviour of densities under invertible (bijective) transformations is given in the following. Consider a random variable $\boldsymbol{x}$ of arbitrary (finite) dimension that is distributed according to the density $p_{\boldsymbol{x}}(\boldsymbol{x})$. When $\boldsymbol{x}$ is transformed by a bijective function

$$\boldsymbol{y} = f(\boldsymbol{x}) \ , \tag{54}$$

[7] A rigorous framework for the transformation of densities is offered by measure theory [22].

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

the new random variable $\boldsymbol{y}$ follows a distribution with the new density $p_{\boldsymbol{y}}(\boldsymbol{y})$. For a deterministic and invertible function f, the probability of the random variable $\boldsymbol{y}$ being inside the infinitesimal volume $d\boldsymbol{y}$ must be the same as the probability of $\boldsymbol{x}$ being in $d\boldsymbol{x}$, since $\boldsymbol{y}$ is uniquely determined by $\boldsymbol{x}$. This condition reads

$$p_{\boldsymbol{y}}(\boldsymbol{y})d\boldsymbol{y} = p_{\boldsymbol{x}}(\boldsymbol{x})d\boldsymbol{x} \ . \tag{55}$$

Solving Eq. (55) for $p_{\boldsymbol{y}}(\boldsymbol{y})$ and inserting $y = f(\boldsymbol{x})$ yields

$$p_{\boldsymbol{y}}(\boldsymbol{y}) = p_{\boldsymbol{x}}(\boldsymbol{x}(\boldsymbol{y}))\left|\frac{d\boldsymbol{x}(\boldsymbol{y})}{d\boldsymbol{y}}\right| = p_{\boldsymbol{x}}\left(f^{-1}(\boldsymbol{y})\right)\frac{1}{|f'\left(f^{-1}(\boldsymbol{y})\right)|} \ , \tag{56}$$

in the one dimensional case. For vector valued random variables the derivative $f'(\boldsymbol{x})$, which takes into account the change of volume, needs to be replaced by the Jacobian $\boldsymbol{D}f$ with components $(\boldsymbol{D}f)_{i,j} := \frac{\partial}{\partial_{\boldsymbol{x}j}} f_i$, hence

$$p_{\boldsymbol{y}}(\boldsymbol{y}) = p_{\boldsymbol{x}}\left(f^{-1}(\boldsymbol{y})\right)\frac{1}{|\det\left((\boldsymbol{D}f)\left(f^{-1}(\boldsymbol{y})\right)\right)|} \ . \tag{57}$$

Equation (57) is referred to as the change of variables formula [22]. As long as the inverse function f^{-1} and the Jacobi determinant are tractable, the density $p_{\boldsymbol{y}}(\boldsymbol{y})$ can be calculated analytically. The difficulty arises when f is not bijective, meaning that a unique inverse cannot be found.

3.4.2 Non-Invertible Transformations

The change of variables formula is not directly applicable to non-invertible transformations [22, 23]. This becomes apparent, as there might be more than one vector $\boldsymbol{x}$ that gets mapped onto the same point $\boldsymbol{y}$, i.e., instead of one unique vector, $f^{-1}(\boldsymbol{y})$ is in general a (not necessarily finite) set. We state without proof (for further details see, e.g., the supplementary material of [23]) that the change of variables formula can be restored by integrating over the set $f^{-1}(\boldsymbol{y})$. The transformation law then reads

$$p_{\boldsymbol{y}}(\boldsymbol{y}) = \int_{\boldsymbol{x}\in f^{-1}(\boldsymbol{y})} p_{\boldsymbol{x}}\left(\boldsymbol{x}\right)\frac{1}{|\det\left((\boldsymbol{D}f)\left(\boldsymbol{x}\right)\right)|}d\boldsymbol{x} \ . \tag{58}$$

Equation (58) can be understood as every $\boldsymbol{x}$ that lies within $f^{-1}(\boldsymbol{y})$ must contribute to $p_{\boldsymbol{y}}(\boldsymbol{y})$. Finding a parametrization of the set $f^{-1}(\boldsymbol{y})$ and performing the integration is often very difficult, even numerically. In the next section we restrict the transformations of interest to special kinds of surjective mappings. Although the integral might still be difficult to solve, the set $f^{-1}(\boldsymbol{y})$ can always be parametrized for these functions.

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

3.4.3 Neural Network Transformations

In deep learning, the function acting on the input data is usually a series of matrix multiplications $\boldsymbol{y}' = \boldsymbol{A}\boldsymbol{x}$ [8] followed by non-linear functions ϕ that act on each component of the RV $\boldsymbol{y}_i = \phi(\boldsymbol{y}'_i)$. We refer to these component-wise functions as activation functions (this term is commonly used in the field of machine learning). It can be shown that arbitrarily complicated functions [24] can be approximated this way. For the latter transformation of the density, Eq. (57) can be used if the activation function ϕ is bijective. The matrix $\boldsymbol{A}$, however, is not invertible if it is rectangular (it does not preserve the dimension of $\boldsymbol{x}$). For the sake of simplicity, we only consider matrices that map a higher dimension onto a lower dimension. Moreover, we require that $\boldsymbol{A}$ is a full rank matrix. Such a transformation is non-injective but surjective.

Equation (58) can be brought to a new form such that the integration is performed over one or more coordinate axes instead of a complex set $f^{-1}(\boldsymbol{y})$. The idea is to extend the transformation, such that f becomes invertible. An arbitrary series of transformations

$$\boldsymbol{y} = f(\boldsymbol{x}) = \phi_N \circ \boldsymbol{A}_N \circ \cdots \circ \phi_2 \circ \boldsymbol{A}_2 \circ \phi_1 \circ \boldsymbol{A}_1(\boldsymbol{x}) \ , \tag{59}$$

where the dimension of $\boldsymbol{y}$ is smaller than the one of $\boldsymbol{x}$, is replaced by another transformation

$$\boldsymbol{y}^* = \begin{pmatrix} \boldsymbol{y} \\ \boldsymbol{v} \end{pmatrix} = f^*(\boldsymbol{x}) = \phi_N \circ \boldsymbol{A}^*_N \circ \cdots \circ \phi_2 \circ \boldsymbol{A}^*_2 \circ \phi_1 \circ \boldsymbol{A}^*_1(\boldsymbol{x}) \ , \tag{60}$$

where $\boldsymbol{y}^*$ has the same dimension as $\boldsymbol{x}$. The rectangular matrices $\boldsymbol{A}_i$ are replaced by square matrices $\boldsymbol{A}^*_i$ with $\det(\boldsymbol{A}^*_i) \neq 0$. As a consequence, every part of the composition f^* is invertible and so is the composition itself. Since f^* is uniquely invertible, the change of variables formula can be applied.

$$p_{\boldsymbol{y}^*}(\boldsymbol{y}^*) = p_{\boldsymbol{x}}\left(f^{*-1}(\boldsymbol{y}^*)\right) \frac{1}{|\det\left((\boldsymbol{D}f^*)\left(f^{*-1}(\boldsymbol{y}^*)\right)\right)|} \ . \tag{61}$$

$\boldsymbol{y}^* = (\boldsymbol{y}, \boldsymbol{v})^T$ consists of the variable of interest $\boldsymbol{y}$ and an auxiliary variable $\boldsymbol{v}$. Integration over the auxiliary variable $\boldsymbol{v}$ yields the transformed density function

$$p_{\boldsymbol{y}}(\boldsymbol{y}) = \int d\boldsymbol{v}\, p_{\boldsymbol{x}}\left(f^{*-1}(\boldsymbol{y}^*)\right) \frac{1}{|\det\left((\boldsymbol{D}f^*)\left(f^{*-1}(\boldsymbol{y}^*)\right)\right)|} \ . \tag{62}$$

The choice of the extended matrices and the auxiliary variables is discussed in Sec. B.5.1.

3.5 A Lower Bound of the Fisher Information

Estimating the FI for a data set in a realistic setting might involve an unknown density, that cannot be estimated with sufficient accuracy and relating it to a known density via

[8] $\boldsymbol{y}'$ denotes an intermediate stage of the transformation $\boldsymbol{y} = \phi(\boldsymbol{y}') = \phi(\boldsymbol{A}\boldsymbol{x})$

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

a transformation might also not be feasible. It is therefore necessary to find methods of approximating the FI that work without evaluating the PDF directly. For that sake, we introduce the so-called Linear Fisher Information (LFI)

$$J(\theta) = \left(\partial_\theta \boldsymbol{\mu}(\theta)^T\right) \boldsymbol{\Sigma}(\theta)^{-1} \left(\partial_\theta \boldsymbol{\mu}(\theta)\right) \ . \tag{63}$$

$\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$ are the mean vector and the covariance matrix, respectively. The importance of the LFI comes with the fact that it is a strict lower bound [9] of the exact FI

$$J(\theta) \leq I(\theta) \ . \tag{64}$$

The LFI coincides with the first term of the FI for the multivariate normal distribution for a scalar parameter, see Eq. (6). When the covariance matrix does not depend on the parameter, the second term (the trace term) in Eq. (6) vanishes and the LFI coincides with the exact FI. Therefore, when the unknown distribution is close to Gaussian and the trace term is small, the LFI is expected to be a suitable approximation of the FI. In order to deal with situations where this assumption does not hold, a new method introduced in this thesis can be used, that combines the ideas of LFI and the FII to get a good estimate of the exact FI. This concept is explained in detail in Sec. 3.5.3.
Note that $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$ depend on the PDF, but unlike the FI, they can be estimated on a finite sample, e.g., with Eq. (50) and Eq. (51), even if the PDF is not available. When $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$ are replaced by the sample mean and the sample covariance matrix and the two quantities are inserted into Eq. (63), the resulting quantity is only an estimate of J. As this estimate is of random nature, it could in principle be larger than the exact FI. This problem can occur, e.g., when the data set at hand is too small, such that the estimated LFI fluctuates strongly. This can always be checked by successively removing data points and monitoring the value of the LFI. Within the scope of this thesis, however, we exclusively deal with synthetic data sets, where it is always possible to generate more data until these fluctuations become sufficiently small.

3.5.1 Linear Fisher Information and Efficiency

As described in Sec. 2.1.4 the FI cannot increase when the random variable $\boldsymbol{x}$ is transformed according to $\boldsymbol{y} = f(\boldsymbol{x})$, i.e., $I_{\boldsymbol{y}} \leq I_{\boldsymbol{x}}$. The LFI $J_{\boldsymbol{y}}$, as a lower bound of the FI $I_{\boldsymbol{y}}$ cannot exceed $I_{\boldsymbol{y}}$ and thus $J_{\boldsymbol{y}} \leq I_{\boldsymbol{x}}$. For one dimensional y Eq. (63) reduces to $J = \left(\partial_\theta \mu\right)^2 / \sigma^2$. Inserting into Eq. (64) directly yields $\sigma_y^2 \geq \left(\partial_\theta \mu_y\right)^2 / I_{\boldsymbol{x}}$, which coincides with the CRI when the transformation f is viewed as an estimator for the parameter. If f is an efficient estimator, then $J_{\boldsymbol{y}} = I_{\boldsymbol{x}}$. The efficiency of any general estimator (Eq. (26)) can now be rewritten as

$$\epsilon(\theta) = \frac{J_y(\theta)}{I_{\boldsymbol{x}}(\theta)} \leq 1 \ , \tag{65}$$

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

i.e., the LFI of the transformed data divided by the FI of the original data. Constructing a good estimator with high ϵ is therefore, at least in one dimension, equivalent to finding a transformation of the input data that maximizes the LFI.

3.5.2 Example: The Linear Fisher Information of a Univariate Normal Distribution

The following example allows for a better understanding of the concept of LFI. Consider a univariate normal distribution (see Sec. 2.1.3) with the density

$$p(x) = \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right) . \tag{66}$$

When the parameter of interest is μ, the FI coincides with the LFI

$$I(\mu) = J(\mu) = \frac{1}{\sigma^2} , \tag{67}$$

thus the LFI is maximal to begin with.
In contrast, choosing the parameter to be the variance $v = \sigma^2$, the FI reads

$$I(v) = \frac{1}{2v^2} , \tag{68}$$

however the LFI vanishes

$$J(v) = 0 , \tag{69}$$

since it is proportional to $(\partial_v \mu)^2 = 0$. The transformation that maximizes the LFI can be found using Eq. (25) and it reads

$$\tau(x) = \frac{\partial_v \log(p(x))}{I(v)} + v = (x-\mu)^2 , \tag{70}$$

which is independent of v and thus yields an efficient estimator. The transformation $y = \tau(x)$ results in a transformed density

$$p_y(y) = \frac{e^{-y/(2\sigma^2)}}{\sqrt{2\pi y}\sigma} , \tag{71}$$

The new LFI reads

$$J(v) = \frac{1}{2v^2} = I(v) . \tag{72}$$

3.5.3 The Lower Bound Maximization Algorithm

We seek for an algorithm to find a suitable function $f(\boldsymbol{x})$ such that the transformed LFI $J_{\boldsymbol{y}}$ for a data set becomes large and a good approximation for the initial FI $I_{\boldsymbol{x}}$. This procedure can be viewed as a numerical extension of Eq. (70), with the difference that the PDF is unknown. Such an algorithm is employed for the following reasons:

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

- The result of the algorithm can be a good approximation of the exact FI even if the original density is intractable.

- Data with high LFI is easier to process, since we know from Sec. 3.5.1 that if the LFI is large, it is always possible to construct an estimator with good efficiency. In this case the only remaining task is to compress the data while keeping the LFI constant. This compression can be done by applying a linear transformation. One caveat is that in general, different linear transformations are needed for different parameter values.

In case an efficient estimator does not exist, the condition $J_{\boldsymbol{y}} = I_{\boldsymbol{x}}$ cannot be fulfilled but it might still be possible to bring the LFI close to the exact FI. There are no strict constraints on the function f, thus there is a large degree of freedom in the choice of the algorithm. In the following, a sketch of a basic implementation of such an algorithm is given:

- Apply a non-linear transformation to the data. This function needs to be sufficiently elaborate. This can, e.g., be achieved by a series of invertible matrices and non-linear but bijective functions, similar to Eq. (60). The entries of the matrices are chosen according to a suitable random distribution.

- The LFI $J_{\boldsymbol{y}}$ of the transformed data is calculated with Eq. (63) and compared to the LFI $J_{\boldsymbol{x}}$ before the transformation was applied, i.e., a function $g(J_{\boldsymbol{x}}, J_{\boldsymbol{y}})$ is evaluated. A basic example for this function would be $g(J_{\boldsymbol{x}}, J_{\boldsymbol{y}}) \sim \mathrm{sign}(J_{\boldsymbol{y}} - J_{\boldsymbol{x}})$, however, more sophisticated choices, such as a Boltzmann distribution $g(J_{\boldsymbol{x}}, J_{\boldsymbol{y}}) \sim \exp(A(J_{\boldsymbol{y}} - J_{\boldsymbol{x}}))$ could be advantageous, as they can be more robust to local minima.

- Depending on the value of $g(J_{\boldsymbol{x}}, J_{\boldsymbol{y}})$, it is decided whether the adjustable parameters of the transformation, specifically, the entries of the matrices, are updated or kept constant.

- These steps are repeated until some convergence criterion is met.

In this basic algorithm, the transformation of the data is an invertible function so the exact FI $I_{\boldsymbol{x}}$ stays constant. However, the best possible transformation might not lie within the subspace of bijective functions. Allowing for non-invertible transformations might lead to a decrease in $I_{\boldsymbol{x}}$ but faster convergence and possibly more accurate results. Figure 3 shows the concept of maximising the LFI using a general, non-invertible transformation.

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

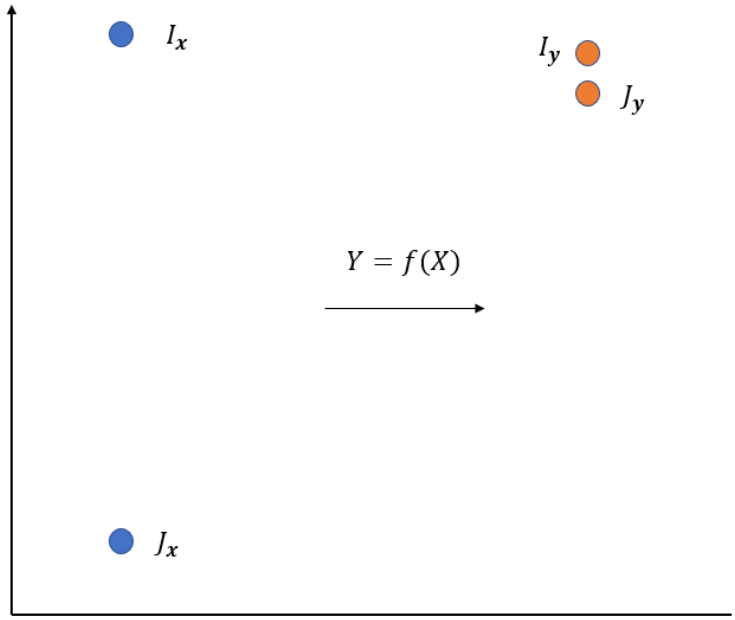

Figure 3: The blue dots show the FI $I_{\boldsymbol{x}}$ and the LFI $J_{\boldsymbol{x}}$ of the initial random variable $\boldsymbol{x}$. After an arbitrary transformation is applied to the data, these are translated into $I_{\boldsymbol{y}}$ and $J_{\boldsymbol{y}}$ of the transformed random variable. Even though the transformed FI might not be exactly equal to the initial FI, the transformed LFI can be a good approximation of the initial FI.

Non-bijectiveness can be achieved by choosing non-bijective activation functions ϕ. Another possibility for extending the transformation to a more general class of functions is allowing for rectangular matrices. As a result, the dimension of the random vector is not preserved and the transformation does not have a unique inverse.
An alternative to the algorithm described above is briefly elaborated, as it is also used in this work. The difference lies in the way the transformation is constructed and in the update of the parameters. Instead of applying a series of a fixed number of random matrices and activation functions and fine tuning all the parameters at once, we take an iterative approach and let the transformation grow step by step, i.e., we start off with a single random matrix followed by an activation function, adjust the corresponding parameters such that the update criterion is satisfied and proceed likewise with the next steps, always freezing the parameters of the previous steps when adjusting the parameters of the new step. One advantage of this approach is that it can yield good results in very few iterations, however, its accuracy might not be satisfactory.
These two algorithms can be combined by constructing a transformation with the latter and using the former for fine tuning all transformation parameters afterwards.

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

4 Artificial Neural Networks and Deep Learning

We briefly elaborate on the concepts of artificial neural networks (ANNs), deep learning (DL) and machine learning (ML). In the vast field of artificial intelligence (AI), software is used to automate intelligent tasks that are normally performed by humans, such as speech or image recognition, medical diagnosis or recognising patterns in data [1, 2]. Some tasks, as differentiating between images of cats and dogs are easily performed by humans, while seeking for structures in large databases cannot be done in a reasonable amount of time. Both of these problems can be approached by AI. Interestingly, AI often excels at recognizing extremely complex patterns but fails at tasks that appear trivial to humans. This discrepancy is not fully understood yet and it is a task of major importance to seek for a better understanding of AI's fundamental principles.

The idea behind ML is that a computer learns to solve a task from a data set. Rather than being programmed, the software learns statistical correlations between input and output data [2]. As humans learn from experience, a ML algorithm learns from the data it receives. A sub-discipline of ML that is frequently deployed is DL. Here the relations between the input and the output data are modelled by a transformation that is a composition of several simple building blocks [1]. These building blocks define the so-called layers of an ANN (see Sec. 4.1 for further details) and the term "deep" refers to the existence of multiple layers. The so-called input layer receives an element of a data set, usually in the form of a vector, which then gets transformed on the way to the output layer. The stages in between are referred to as the hidden layers, since the data at these stages is usually not directly observed as it is virtually impossible for a human to extract any features from these representations of the data.

4.1 Deep Neural Networks

Mathematically, a DL model is a function composed of multiple functions. An ANN must be able to describe a relation between input and output data that is possibly very complex, thus this composite function should in general be non-linear and highly complex as well. The basic idea is then to make an ansatz for a function that is parametrized by multiple parameters and to incrementally adapt these parameters while feeding data to the network, such that the input-output relation improves with each iteration.

4.2 The Structure of Deep Neural Networks

There are many ways of defining a parametric function that can approximate any arbitrarily complex function. In principle, a Taylor series or a Fourier series could be used, with the parameters being the coefficients. This approach is, however, not feasible for high dimensional data. For example, in order to make a third order approximation of a function that receives an N dimensional input vector, the number of adjustable parameters is at the order of N^3. When the function that is modelled is of a much more complex nature, it becomes apparent

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

that a polynomial ansatz is pointless, as the number of adjustable parameters grows exponentially with the polynomial's order.
The ansatz [1] that is commonly deployed in DL is similar to the transformation Eq. (59), i.e., each layer of the network consists of a linear transformation followed by a non-linear activation function acting on every component of the vector it receives in the same way. If the size of these matrices does not greatly exceed the dimension of the input vector N, the number of adjustable parameters is of the order $N \times M$, with M being the number of layers. For large N and $M \ll N$ (very common in DL tasks), the ANN approach has already less parameters than a qubic polynomial although it is of much higher complexity.
Specifically, the data in the layer l of the deep network is calculated as

$$a_i^l = \phi^l \left(b_i^l + \sum_{j=1}^{N_l - 1} w_{ij}^l a_j^{l-1} \right) . \tag{73}$$

The entries a_i of the transformed vectors are called activations of the respective neurons, in analogy to the human brain. Visualizing the function as a collection of artificial neurons with activations that interact with other neurons via some mathematical rule, motivates the term ANN. A common way of sketching an ANN is shown in Fig. 4.

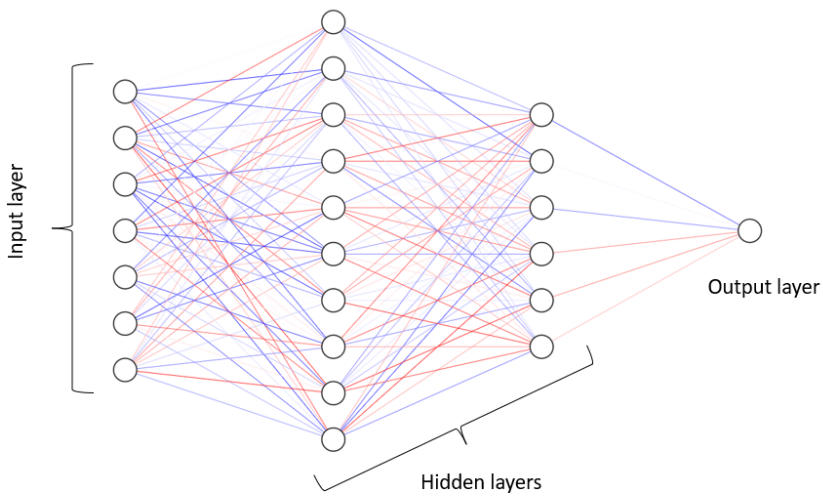

Figure 4: A schematic representation of an ANN. The white dots depict the artificial neurons. The lines connecting the neurons of subsequent layers represent the weights and the different colors of these lines denote their magnitudes.

A DNN can then be regarded as a series of functions that transforms the activations of the input layer $a_i^{l=0}$, i.e., the input data, to the activations in the output layer $a_i^{l=N}$, the output data. The linear transformations are determined by w_{ij}^l and b_i^l, the components of the so-called weights and biases, which are parameters that are adjusted during the learning process. The corresponding weight matrices and bias vectors are denoted by w^l and b^l. An ANN based on the structure Eq. (73) is referred to as a fully connected feedforward neural

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

network. The name "feedforward" comes from the fact that the activations a_i^{l+1} in the layer $l+1$ are directly calculated from the previous activations a_i^l. Figure 4 shows an example of such a fully connected feedforward ANN. The term "fully connected" expresses that each activation a_i^l contributes to each activation a_j^{l+1}. In the remainder of this thesis we exclusively consider fully connected, feedforward ANNs.

Some of the activation functions ϕ^l, commonly those of the hidden layers, are chosen to be non-linear, to ensure that the ANN is able to learn functions with sufficient complexity. In case that every activation function is linear, Eq. (73) reduces to a linear transformation. The activation of the output layer needs to be chosen such that the output activations can express the output data. For instance, if the network should learn a continuous number between -1 and 1, the range of the output activation should match that region (this is achieved, e.g., by $\phi^N = \tanh$). When the number of layers, the number of nodes in each layer and the activation functions are chosen appropriately such that the desired connection between input and output data can be modelled, one remaining task is to implement an algorithm to adjust the weights and biases in Eq. (73).

4.3 Training of an ANN

The process of finding a suitable set of weights and biases, such that the ANN models the input-output relation accurately, is referred to as the training of an ANN. The basic concepts of training are shown based on fully connected feedforward neural networks.

4.3.1 Cost Function and Loss Function

Training an ANN requires a measure to compare different parameter configurations among each other. This is achieved by a loss function (see Sec. 2). The explicit choice of the loss function depends on the problem at hand. Many ML tasks fall either into the category regression or classification. The former is closely related to parameter estimation (see Sec. 2.1), whereas the latter is closely connected to decision theory and hypothesis testing (see Sec. 2.3). As described in Sec. 2, loss functions commonly deployed in these situations are the MSE and the cross-entropy, respectively, although many others exist. For a concise notation of the input-output relationship we introduce $\hat{y} = f_w(\boldsymbol{x})$, which is defined as the activation in the output layer given the input activation $\boldsymbol{x}$. The index w denotes the set of trainable parameters, i.e., the weights and biases. In a ML setting, the cost function $C(y, f_w(\boldsymbol{x})) = C(y, \hat{y})$ measures how well the predicted value $\hat{y}$ approximates the true output value y. The loss function $L(w)$ is then defined as the cost function averaged over the data, i.e., over the distribution $(\boldsymbol{x}, y) \sim P$ of the input-output pairs $(\boldsymbol{x}, y)$

$$L(w) := \mathrm{E}_{(\boldsymbol{x},y)\sim P}\left[C(y, f_w(\boldsymbol{x}))\right] \ . \tag{74}$$

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

$L(w)$ is in general unknown as the distribution P is unknown, thus in practice, the expectation is replaced by a sample average over small subsets B of the data set, the so-called batches. The objective of DL is then to tune the parameters w in order to minimize $L(w)$.

4.3.2 The Backpropagation Algorithm

With the notion of loss functions, training an ANN can be formulated as an optimization problem. We seek for an algorithm that searches for an optimal set of parameters w such that the loss function $L(w)$ becomes minimal. In feedforward ANNs this is commonly achieved by the so-called backpropagation algorithm. The parameters are updated based on the gradient descent

$$w_{n+1} = w_n - \lambda \nabla_w L(w_n) \ , \tag{75}$$

with the learning rate λ. Replacing the loss function by an average over a batch allows to calculate the gradient analytically. Since the "forward propagation" $f_w(\boldsymbol{x})$ is a composition of functions, the chain rule can be applied to arrive at an explicit expression for the components of the gradient. Repeating this process of updating the parameters with multiple batches until the entire training data set is shown to the ANN is referred to as an epoch. Usually, the data set is then rearranged (shuffled) and fed back to the ANN to perform more epochs of training. In the following we address the question of how long an ANN should be trained.

4.4 Underfitting and Overfitting

An ANN that is not trained at all or only shown a few batches of training data will not be able to fit a complex connection between input and output data. If the network did not yet learn to express most of the important relations, one speaks of underfitting. To mitigate this problem, one could simply extend the training process to a very large number of epochs. However, under some conditions, the finite size of the data set at hand results in a complementary effect, which is called overfitting. This problem can be intuitively understood in the limit of infinitely many parameters and infinite training duration (infinite number of epochs):

Since the data can be regarded as drawn from a probability distribution, it is inherently noisy and it is impossible to find an exact relation between input and output. However, a fitting model with sufficient complexity could in principle fit this noisy relation exactly when the sample size is finite. In that case, the ANN would handle the training data perfectly, but it would fail at generalizing to new data that comes with a different noise realization.

This phenomenon does not occur for an ANN with very few parameters, as it cannot approximate a sufficiently complex function. Similarly, for larger data sets it is more difficult to model the noise when the number of parameters is kept the same. Another way to avoid overfitting is to stop the training before overfitting takes place. This technique is referred to as early stopping (see Sec. 4.4.1).

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

4.4.1 Training, Validation and Testing

In order to avoid overfitting, some portion of the data set is not used for training but is reserved for the purpose of quantifying the performance of the ANN. Typically, the largest part of the data is used for training while the remainder is split into a validation and a test data set.

To make sure the ANN is not strongly overfitting, the loss function Eq. (74) is not only estimated based on the training data, but also on an independent validation data set. The cost function averaged over the training / validation data set is what we call the training / validation loss, respectively. The ANN did not yet "see" the validation data set, hence a network that is overfitting will perform worse on the validation data set than on the training data set, i.e., the validation loss will be significantly larger than the training loss.

Typically, both the training and the validation loss at first decrease as a function of the training duration but the latter eventually reaches a minimum and starts to increase. This behaviour is demonstrated in Fig. 5.

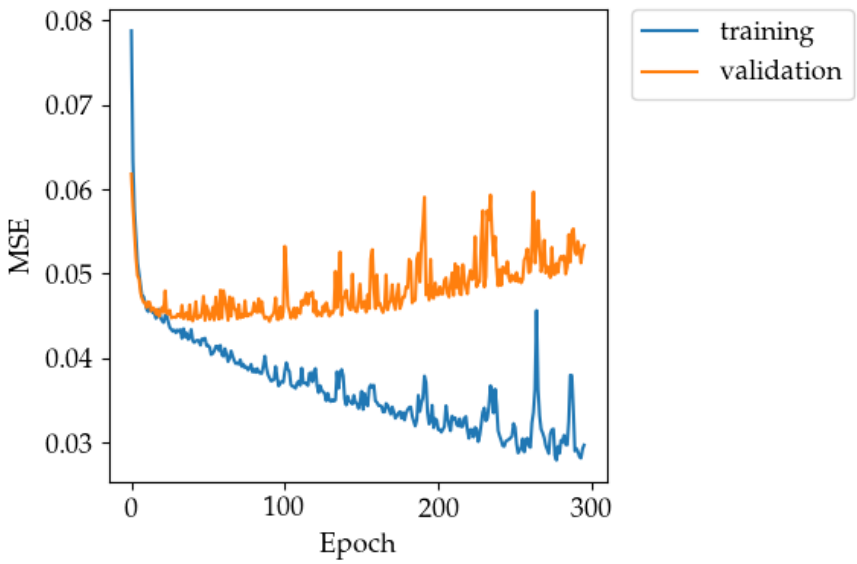

Figure 5: The blue curve shows the training loss and the orange curve the validation loss as a function of the training time (measured in epochs). The loss function is the MSE. The training loss decreases arbitrarily but the validation loss is clearly bounded from below.

Stopping the training when the minimum of the validation loss is reached is referred to as early stopping. Besides early stopping, the validation data set can be used to tune all kinds of the network's external parameters (hyperparameters) such as the number of layers, the number of nodes, etc. As knowledge about this data set is included into the network's training process, the empirical evaluation of the loss function on the validation data set will be biased. Therefore, the third independent portion of the data, the so-called test data set is reserved to evaluate the loss function after the training. No hyperparameters are changed based on the test data.

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

In general, deciding how much data is reserved for validation and testing is a difficult task. The more training data is available, the better the ANN will be able to learn features. When the validation data set is too small, however, the validation loss might not yield an accurate estimation of the true loss function. In Sec. 5.2 we propose a new technique based on Fisher information that allows to dispense with validation data.

5 Fisher Information in Neural Network Regression Models

The concepts of estimation theory introduced in Sec. 2 can be used to quantify the performance of Neural Networks that are trained for regression tasks. When a regression model is interpreted as a statistical estimator (see Sec. 2.2.1), the Cramér-Rao Inequality yields an upper bound for its efficiency. In this thesis, for the first time, we peek into the black box of the ANN, by tracking the FI through its layers. Based on the layer-wise transformations of a deep ANN, a FII can be formulated for neural networks.

5.1 Approximating a Cramér-Rao Efficient Estimator

Consider an estimation problem with data that is distributed according to $P(\theta)$ with the density $p(\boldsymbol{x}|\theta)$. The goal is to train an ANN such that it returns an estimate of the parameter θ when it receives input data. Training an ANN to predict θ from a random vector $\boldsymbol{x}$ is then equivalent to constructing an estimator $f_w(\boldsymbol{x}) = \hat{\theta}(\boldsymbol{x})$. The estimator $f_w(\boldsymbol{x})$ depends on the network's parameters, such as the activation functions, the weights and the biases, but is independent of the parameter θ to estimate. We refer to it as a Neural Network Estimator (NNE).
Recall from Sec. 2.1.5 that a lower bound for the variance of the estimator is given by the CRI

$$\sigma(\hat{\theta})^2 \geq \frac{(\partial_\theta \mu(\hat{\theta}))^2}{I(\theta)} \ , \tag{76}$$

where $\mu(\hat{\theta}) = \mathrm{E}[f_w(\boldsymbol{x})]$ is the expectation of the estimator, $\sigma^2(\hat{\theta}) = \mathrm{E}[(f_w(\boldsymbol{x}) - \mu(\hat{\theta}))^2]$ is its variance and $I(\theta) = \mathrm{E}[(\partial_\theta \log(p(\boldsymbol{x}|\theta)))^2]$ is the Fisher information. The expectation E is done with respect to the density $p(\boldsymbol{x}|\theta)$. The CRI can also be rewritten in terms of the efficiency $\epsilon(\hat{\theta})$ and the LFI $J(\hat{\theta}) = (\partial_\theta \mu(\hat{\theta}))^2/\sigma(\hat{\theta})^2$

$$\epsilon(\hat{\theta}) = \frac{J(\hat{\theta})}{I(\theta)} \leq 1 \ . \tag{77}$$

Equation (77) states that the efficiency of the NNE cannot exceed the value 1, and this result is independent of how large the network is and how long it is trained. Note that an efficient estimator $\epsilon(\hat{\theta}) = 1$ only exists if the data follows a distribution from the exponential family. For non-exponential family distributions, however, it is often possible to find an

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

approximately efficient estimator with $\epsilon(\hat{\theta}) \approx 1$.
Several ANNs are trained to predict parameters from simple distributions, such as the mean from a Gaussian distribution, in order to analyse their efficiency. Synthetic training data of arbitrary size can then be generated, which allows for an accurate prediction of the estimator's mean and variance. Moreover, if the random variable $\boldsymbol{x}$ stems from a simple distribution, where an analytic expression for the density $p(\boldsymbol{x}|\theta)$ is available, the Fisher information is known either analytically, or it can be precisely estimated with a Monte Carlo simulation.

For estimation problems the MSE loss

$$L(\theta, f_w(\boldsymbol{x})) = MSE(\theta, f_w(\boldsymbol{x})) = \mathrm{E}[(f_w(\boldsymbol{x}) - \theta)^2] \ , \tag{78}$$

turns out to be a convenient choice, although most of the loss functions that are suitable for regression problems could be used as well. Recall from Sec. 2.1.1 that this equation can be expressed in terms of the estimator's bias $b(\hat{\theta})$ and its variance

$$MSE(\theta, f_w(\boldsymbol{x})) = \sigma(\hat{\theta})^2 + b(\hat{\theta})^2 = \sigma(\hat{\theta})^2 + (\mu(\hat{\theta}) - \theta)^2 \ . \tag{79}$$

As for unbiased estimators, the MSE is equal to the variance, hence the Cramér-Rao inequality yields a lower bound for the loss function

$$L(\theta, f_w(\boldsymbol{x})) \geq \frac{1}{I(\theta)} \ . \tag{80}$$

If we expect the Neural Network model to solve the regression problem at hand by fitting an (approximately) unbiased estimator, Eq. (80) yields a point of reference for when to stop training, that is, when the equality in Eq. (80) is (approximately) achieved and thus the ANN acts as an efficient estimator. Further minimization of the loss function is only possible when bias is added, which seems counter-intuitive, as the bias contribution of the MSE is always positive. However, the CRI Eq. (76) states that the variance of the estimator can be smaller than the inverse FI when the expectation value of the estimator has a flat shape, i.e., $\partial_\theta \mu(\hat{\theta}) < 1$.

5.1.1 Calculating the Efficiencies of Neural Network Estimators

In order to test whether the efficiencies in ANNs reach the CRLB, we train ANNs to perform estimation tasks. In order to not repeatedly give the same explanation, the preparation of the data sets for training, validation and testing of NNEs is briefly explained in Sec. E.1. All the information about the ANNs used in this chapter is given in Sec. E.2.

The first network that is studied receives 50 dimensional, normal and i.i.d. data, sampled from the density $p(\boldsymbol{x}|\theta) = \prod_{i=1}^{50} \frac{1}{\sqrt{2\pi\sigma^2}} e^{-(x_i-\theta)^2/2\sigma^2}$, with $\sigma = 1$. Although an efficient

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek WIEN Your knowledge hub

estimator for this problem exists, training the ANN is non-trivial, as the network only receives a finite sample and no additional information about the underlying distribution is incorporated beforehand. Further details concerning the network are listed in Table 1. We find that the resulting efficiency, evaluated at the parameter $\theta = 0$ is[9]

$$\epsilon(\theta = 0) \approx 0.95 \ . \tag{81}$$

Real world data is often to a good approximation normally distributed. Moreover, an even more realistic scenario can be created by adding non-Gaussian [10] noise to each component of the random vector following the distribution above $x_i \to x_i + \delta_i$. The noise contribution is drawn from a Laplace distribution [25], with a scale factor of 0.18. Note that this choice is completely generic and any other distribution would work as well, as long as the fluctuations of δ_i are small compared to the fluctuations of x_i. For the ANN determined by Table. 2, the efficiency is

$$\epsilon(\theta = 0) \approx 0.95 \ . \tag{82}$$

Interestingly, the ANN trained on the perturbed data set performs equally well as the one trained on the Gaussian data. This is however not a problem, since the FI of the perturbed data is smaller.
To give an example of a distribution that is not similar to a Gaussian, we train a network to estimate the mean of a bimodal distribution[11]. The PDF and the network's specifics are listed in Table. 3. The resulting efficiency is

$$\epsilon(\theta = 0) \approx 0.97 \ . \tag{83}$$

The results above suggest that an ANNs trained to perform parameter estimation tend to approximate efficient estimators, even if the data does not follow a simple Gaussian distribution. Recall that the efficiency $\epsilon(\hat{\theta}) = J(\hat{\theta})/I(\theta)$ is given by the ratio of the linear Fisher information of the estimator and the Fisher information of the input data. This allows us to interpret the action of an NNE on the data in the following way:

- The dimension of the data distribution is compressed to the number of estimated parameters. For example, estimating one parameter from an N dimensional data array corresponds to a compression from an N dimensional to a 1 dimensional distribution.
- The LFI is maximized. In the optimal case of an efficient estimator, the Neural Network acts as a transformation that increases the LFI, such that it coincides with the FI of the input data.

[9] This choice is arbitrary and any other value of θ could have been analysed as well. However, repeating the calculation for a tight grid of θ-values is very time consuming. We thus focus on a single value for demonstration purposes.

[10] Adding Gaussian noise to a Gaussian distribution results in another Gaussian distribution and hence does not yield any additional insights.

[11] i.e., a distribution with two peaks

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

5.1.2 Transformation of the Training Data: Increasing the Linear Fisher Information

Before training a Network, it is common practice to transform the data, such that it is more convenient to process and a less complex model can be employed. This reduces the time required for training and, in addition, allows for better generalizability. A common way of preprocessing the data is standardization, where it is rescaled such that its mean and covariance match those of a standard normal distribution. The results from Sec. 5.1.1 motivate another method of preprocessing based on maximising the linear Fisher information of the input data. We observe the influence of LFI maximization of the training data on the training speed (see Fig. 6).

Considering the input data $\boldsymbol{x}$ whose distribution is determined by the density $p(\boldsymbol{x}|\theta)$, we aim at finding a transformation $\boldsymbol{y} = \Psi(\boldsymbol{x})$, such that the transformed LFI $J_{\boldsymbol{y}}$ is larger than the LFI before the transformation $J_{\boldsymbol{y}}(\theta) > J_{\boldsymbol{x}}(\theta)$. Optimization procedures for finding a suitable transformation are described in Sec. 3.5.3. The transformation Ψ is represented by one or more consecutive Neural Network layers. One caveat of this method is that in general, the theoretically optimal transformation that maximises the LFI depends on the parameter itself. When a suitable transformation is found for one value of θ, the same transformation will not necessarily maximise the LFI for different parameter values. In practice, finding a transformation for one single parameter value θ can still be a viable choice as long as it increases the LFI for different parameters in a similar manner. We give an example, where the optimal transformation is known analytically, and does not depend on the estimated parameter. Such a transformation can always be found if an efficient estimator exists. An ANN is then trained on the original and the transformed data set and their performance is compared by measuring the FI in the output layer as a function of the training duration.

As shown in Sec. 3.5.2, estimating the standard deviation σ from normally distributed data with zero mean, is a problem where the LFI of the raw data vanishes. The data is optimally preprocessed by the activation function $\phi(x) = x^2$. This preprocessing can be directly performed by the neural network when adding a first hidden layer with a unit weight matrix and $\phi(x) = x^2$ as the activation.

Detailed information about the experimental setup can be found in Sec. E.3. In Fig. 6, the normalized[12] FI in the output layer of the ANN, as a function of the training epochs is shown for the original and the preprocessed training data. When trained on the preprocessed data set, the ANN allows more FI to pass through. The training duration can thus be significantly decreased for preprocessed data. This also means that, even when the algorithm to find a suitable transformation for the data at hand is very expensive, it might still be worthwhile, considering the accelerated training. The slight decrease of the FI in the last step is not a general tendency but can happen at any stage of training coincidentally.

[12]The FI in any layer is divided by the FI of the initial data set. The normalized FI in the input layer is then always equal to one and the normalized FIs in the subsequent layers are smaller than one.

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

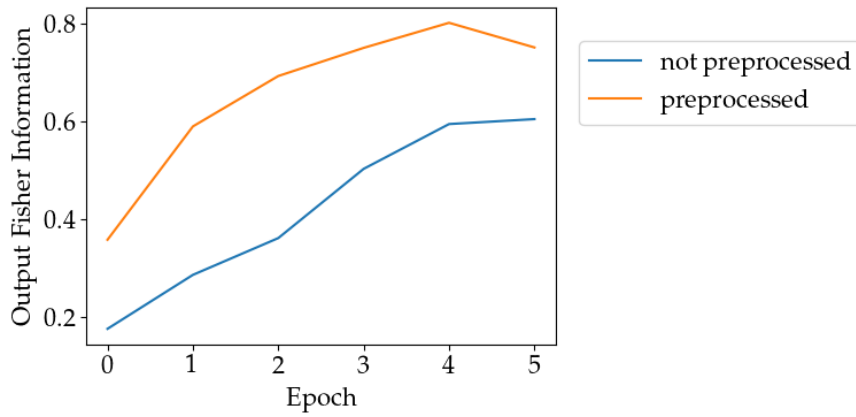

Figure 6: The blue curve depicts the FI (normalized with respect to the FI in the input layer) in the output layer of the ANN, trained on a data set with vanishing LFI. The orange curve shows the output FI of an ANN with the same architecture but trained on preprocessed data, with maximal LFI.

Note that if the goal was to estimate the mean $\boldsymbol{\mu}$ from the data set used above, no preprocessing is required at all, since the LFI with respect to the mean $J(\boldsymbol{\mu}) = I(\boldsymbol{\mu})$ already coincides with the exact FI.

As in many real-world applications the data is close to normally distributed, it might often be possible to guess a suitable transformation Ψ instead of calculating it numerically. For more complex data, that follows, e.g., a multimodal distribution, it can be convenient to seek for a transformation into a lower dimensional subspace. This can be done by adding a first hidden layer with a number of nodes that is less than the number of input nodes. By guessing a suitable activation function and optimizing the weights and bias in the first hidden layer, the LFI can often be increased substantially by a simple algorithm. One needs to be cautious, however, that the dimensionality reduction does not lead to a large loss of the Fisher information contained within the data (see Sec. 5.3).

5.2 Early Stopping based on Fisher Information

In ML terminology, the loss function's lower bound Eq. (80) gives a criterion for early stopping. When the ANN is overfitting, the loss function evaluated on the training data can become arbitrarily small (see Sec. 4.4.1). A loss of $L = MSE = 1/I$ is a necessary condition for an estimator to be unbiased and efficient. From Sec. 5.1.1 we know that ANNs can indeed approximate efficient estimators. Furthermore, we can expect these estimators to not have a strong bias when the MSE is chosen as the loss function, as a large bias implies that the MSE is also large. We track the training loss of an ANN and compare it to the validation loss, to figure out whether or not $L = 1/I$ is a valid condition for stopping the training.
To analyse the behaviour of the loss function of an NNE, a model is trained to predict a parameter θ of normally distributed data. The training data set is again drawn from a multi-

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

variate normal distribution with added noise. As we want to observe the effect of overfitting, we choose a large number of parameters. Further details are listed in Table 5. Figure 7 shows the resulting training and validation loss curves.

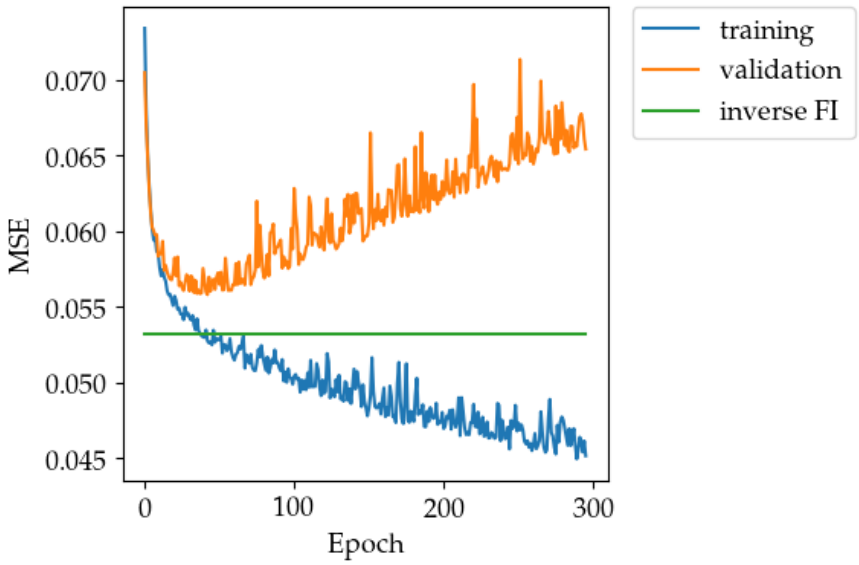

Figure 7: The loss function (MSE) evaluated on the training data set (blue curve) and the validation (orange curve) data set. The green line denotes the inverse FI of the input data.

We can clearly see from the figure that at the time where the training loss becomes equal to the inverse FI, the validation loss is approximately at its minimum. Therefore, when the inverse FI can be precisely estimated from the training data set, the validation data set becomes obsolete. Apparently, the minimum of the training loss does not exactly coincide with the inverse FI. This is due to the fact that the ANN does not allow 100% of the input FI to pass through, but a small part of the information gets lost at some stage of the transformation. The behaviour of FI inside ANNs is studied in the following section.

5.3 The Propagation of Fisher Information through Deep Neural Networks

To this point, we treated entire NNEs as non-linear functions that approximate efficient estimators. The Fisher information of the training data yields an upper bound for the efficiency, i.e., a limit that even a perfect network cannot exceed. ANNs however, approximate functions, so there is in general some loss of information. By viewing deep neural networks as a series of transformations acting on the input data, we formulate a Fisher information inequality and calculate the layer-wise Fisher information for various problems. Measuring a significant loss of information at some layer, hints at an error in the network's parameters.

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

5.3.1 Fisher Information Inequality in Deep Neural Networks

There are multiple ways of defining probability distributions and density functions in ANNs, thus in order to avoid misunderstandings we briefly describe how the layer-wise densities and FIs in an ANN are defined in this thesis. As shown above, the probability density $p(\boldsymbol{x}|\theta)$ of the input data $\boldsymbol{x}$ is transformed by the NNE $\hat{\theta}(\boldsymbol{x}) = f_w(\boldsymbol{x})$. Interpreting the estimator as a transformation $\boldsymbol{y} = f_w(\boldsymbol{x})$ of the random vector $\boldsymbol{x}$ allows to define a probability density $p_{\text{out}}(\boldsymbol{y}|\theta)$ for the output data $\boldsymbol{y}$. The FI of the output data I_{out} cannot be larger than the initial FI I, i.e., $I \geq I_{\text{out}}$. As the function $f_w(\boldsymbol{x})$ in a deep feedforward ANN is a composition of affine transformations (a weight matrix is applied and a bias vector is added) and activation functions

$$\boldsymbol{y} = f_w(\boldsymbol{x}) = \phi_N \circ \boldsymbol{b}_N + \boldsymbol{W}_N \circ \cdots \circ \phi_2 \circ \boldsymbol{b}_2 + \boldsymbol{W}_2 \circ \phi_1 \circ \boldsymbol{b}_1 + \boldsymbol{W}_1 \boldsymbol{x} \ , \tag{84}$$

the data $\boldsymbol{x}^{(i)}$ can be extracted at any layer i when the transformation

$$\boldsymbol{x}^{(i)} = f_w^{(i)}(\boldsymbol{x}) = \phi_i \circ \boldsymbol{b}_i + \boldsymbol{W}_i \circ \cdots \circ \phi_2 \circ \boldsymbol{b}_2 + \boldsymbol{W}_2 \circ \phi_1 \circ \boldsymbol{b}_1 + \boldsymbol{W}_1 \boldsymbol{x} \ , \tag{85}$$

is applied, instead of the full transformation Eq. (84). Mind that the function $f_w^{(i)}(\boldsymbol{x})$ depends on the collection of parameters w, however, the parameters of the layers $l > i$ do not contribute. The FI I_i in the layer i can be calculated from the density $p_i(\boldsymbol{x}^{(i)}|\theta)$ of the random variable $\boldsymbol{x}^{(i)}$. The FII reads

$$I \geq I_1 \geq ... \geq I_{N-1} \geq I_N = I_{\text{out}} \ . \tag{86}$$

Mind that the FI in the output layer yields a lower bound for all the previos FIs. When I_{out} is already very close to I, the same must necessary hold for the FIs in the previous layers.

5.3.2 The Fisher Information in Shallow Neural Networks

Tracking the FI through a deep ANN with multiple hidden layers and arbitrary input densities is a hard problem, since the layer-wise transformations are by construction non-linear and highly complex and thus the probability density is intractable. At the output layer, or more general, in every layer with a very small number of nodes, the FI can be estimated with the histogram method, as it is shown in Sec. 3.3.1. With the result from Sec. C.1 it is now possible to track the propagation of the FI through ANNs with one hidden layer (a shallow neural network). Moreover, as invertible layers do not change the FI, an arbitrary amount of invertible layers can be added to the first hidden layer and the output layer, respectively. We train a network to learn the mean value from a 50-dimensional Gaussian distribution. Further details about the network and the data set are shown in Table 6. Figure 8 shows the normalized FIs in the ANNs layers, each as a function of the parameter. The FI in the input and the first two hidden layers can be calculated analytically. Apparently, at these stages of the transformation, the FI is perfectly conserved. When compressing the data from the

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

25-dimensional second hidden layer to the 1D output layer, some of the FI gets lost. One could achieve a NNE that allows more FI to pass through by adding more parameters, e.g., via a third hidden layer or by adding more nodes to the layers.

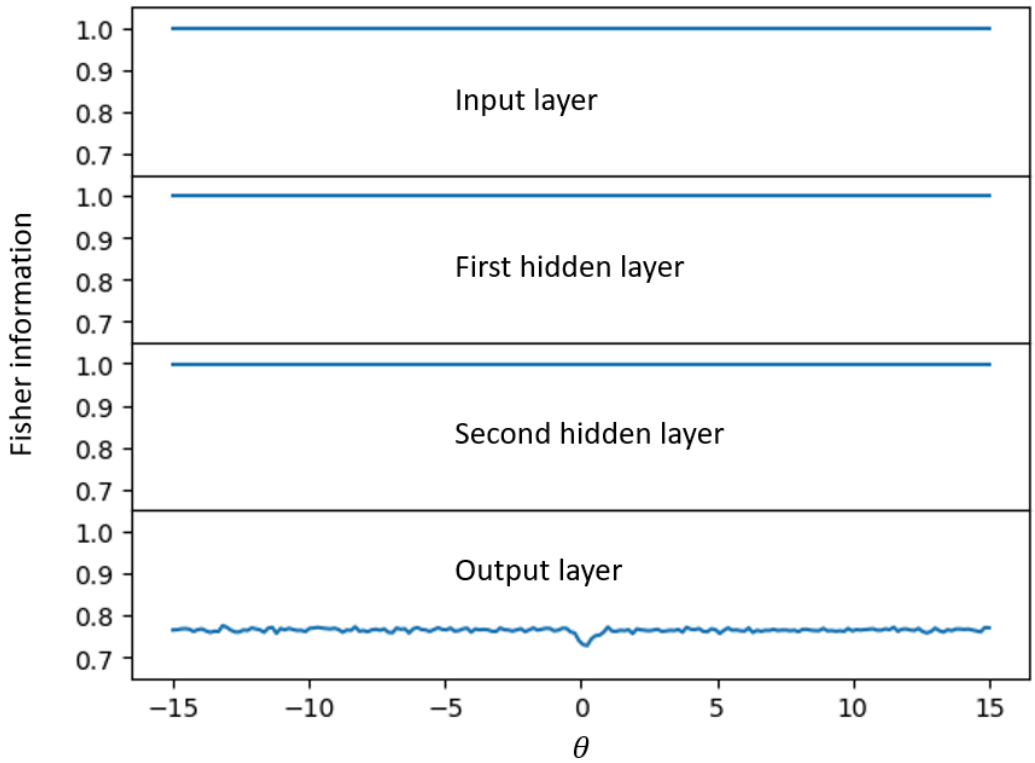

Figure 8: The blue curves show the FI (normalized with respect to the input FI) as functions of the parameter θ in each layer. Ideally, the FI in any layer is $I(\theta) = 1$, although some finite loss of FI, i.e., $I(\theta) < 1$, is typically inevitable. In this experiment, most of the FI loss happens when the data is transferred from the second hidden layer to the output layer.

Although the training data consists of parameters in the range of $-10 \leq \theta \leq 10$ the NNE allows a large part of the FI to pass through, even for parameters outside of this region. This indicates that the network generalizes well. From the small dip of the FI in the output layer at $\theta = 0$ we conclude that the NNE performs slightly worse around the middle of the parameter range that was shown to the network during the training. We expect that this is due to the fact that the training data is centered around $\theta = 0$ and the network learns some bias towards this value.

5.3.3 The Linear Fisher Information Approach

In most applications, neither analytical approaches nor the MCMC method (see Sec. C.2) have the power to track the FI through the entire network. Hence, at some point, approximations have to be made. A powerful approach of calculating the FI is to use the LFI J as an approximation, and, where appropriate, perform LFI maximization such that J gets close

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

to the exact FI. For ANNs with normally distributed input data with the aim of learning the mean, however, the LFI appears to be an approximation that is already sufficient[13]. At first we track the FI propagation through an ANN trained to predict the mean from normally distributed input with multiple hidden layers using the LFI directly. In a next step we analyse the FI of an ANN trained to predict the mean from data drawn from a bimodal distribution. In that case, the LFI is not necessarily a good approximation for the exact FI, thus we apply the LFI maximization algorithm to find a better estimate. For the sake of simplicity, we do not plot the FI as a function of the parameter but only evaluate it at a single point in the parameter space. This is due to the fact that the LFI maximization algorithm must be applied locally for each parameter, which turns out to be a very tedious task.

In the first case we use again 50-dimensional, i.i.d. and normally distributed input data. The difference to the previous analysis is the network architecture, as several hidden layers are added, to increase its complexity. For further details see Table 7. Our goal is to evaluate the FI at a generic parameter value $I(\theta = 1)$ in each layer. The FI in the input and first hidden layer is known analytically and the one in the output layer is calculated using histograms. The FI in the remaining hidden layers is approximated by its LFI. Figure 9 depicts the FI flow through the layers of the ANN. As a result of the FII, the curve is monotonically decreasing. Almost 99% of the FI of the training data reaches the output layer.

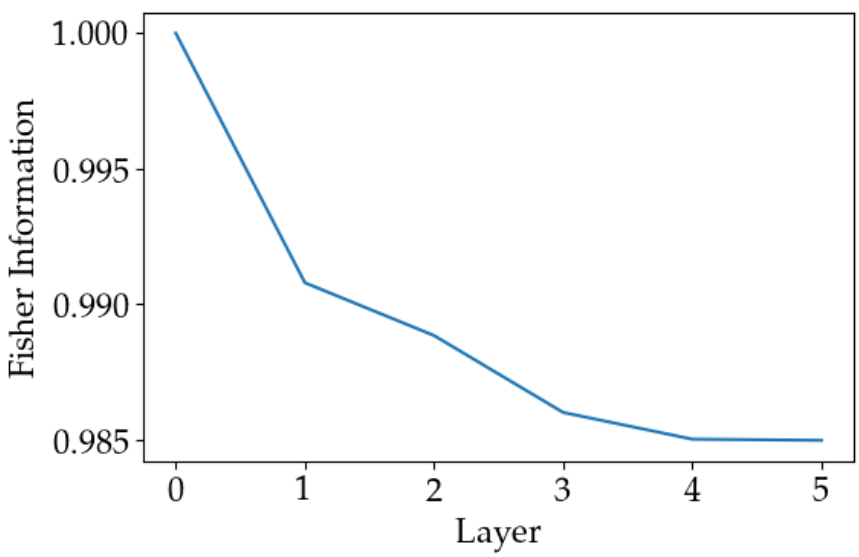

Figure 9: The FI (normalized with respect to the input FI) in each layer of the ANN, trained on Gaussian input data.

For the second experiment we use input data sampled from a 20-dimensional, bimodal distribution (see Table 8). The FI is evaluated at $I(\sigma = 1)$. We find that the LFI maximisa-

[13]Whether the LFI is already a good approximation of the exact FI can be examined by running a LFI maximisation algorithm only for a few iterations. In case the LFI never increases, one can expect it to be close to maximal.

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

tion algorithm must only be applied at the first hidden layer, as the FI is analytically known in the input layer and the LFI is already close to being optimal[14] in the second hidden layer and in the following layers. Figure 10 shows the LFI in each layer, together with the exact FI.

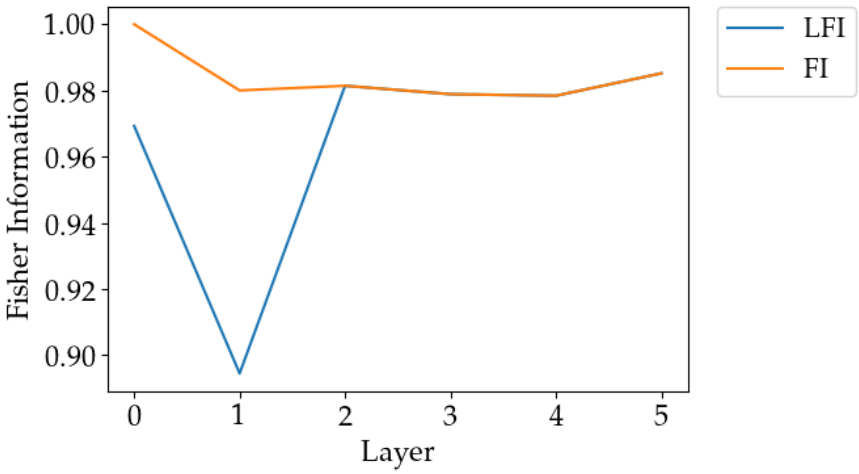

Figure 10: The orange curve depicts the FI (equivalent to the optimized LFI) in each layer of the ANN, trained on bimodal input data. As a comparison, the blue curve shows the LFI (without optimization) in each layer.

One clearly sees that in the first hidden layer, the LFI is not sufficient and the LFI maximisation algorithm can be used to achieve better results. The fact that the LFI is approximately equal to the exact FI in the second hidden layer and consecutive layers suggests that deep neural networks tend to increase the LFI. It is, however, not yet clear whether this behaviour is caused by the consecutive reduction of the dimensionality (the number of nodes decreases decreases or stays constant) or an inherent property of ANNs trained to perform regression tasks.
The FI does not seem to be perfectly monotonous, which violates the FII. This is, however, due to numerical errors, as only approximations of the exact FI in the hidden layers and in the output layer are available.

5.3.4 Tracking the Fisher Information During the Training

It is expected that before any training took place, an ANN is not able to conserve most of the FI of the input data, but a large portion gets lost in some layer. The FI should build up as a function of the training time. Tracking the FI layer by layer during the training allows to observe a 'time-resolved' FI flow.

[14]Note that the optimization becomes increasingly difficult, the closer the LFI is to the exact FI. When the LFI is already very close to the exact FI, performing several optimization steps might not improve the result at all.

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

The ANN is again trained to predict the mean of normally distributed input data. Training data and detailed information about the ANN are given in Table 9. The FI in the input and in the first hidden layer are calculated analytically, while for any other layer the LFI is used as an estimate of the exact FI. For this data set, we dispense with using the histogram method for calculating the FI in the output layer, as the LFI yields the same result (within some numerical uncertainty) but requires less data and is therefore faster. Figure 11 depicts the FI flow through the ANN after each epoch of training. By "epoch 0" we denote the stage before any training happened, thus the weights and biases are random and the network did not yet learn any features from the data. After 5 epochs of training $\sim 95\%$ of the initial FI get transferred to the output layer.

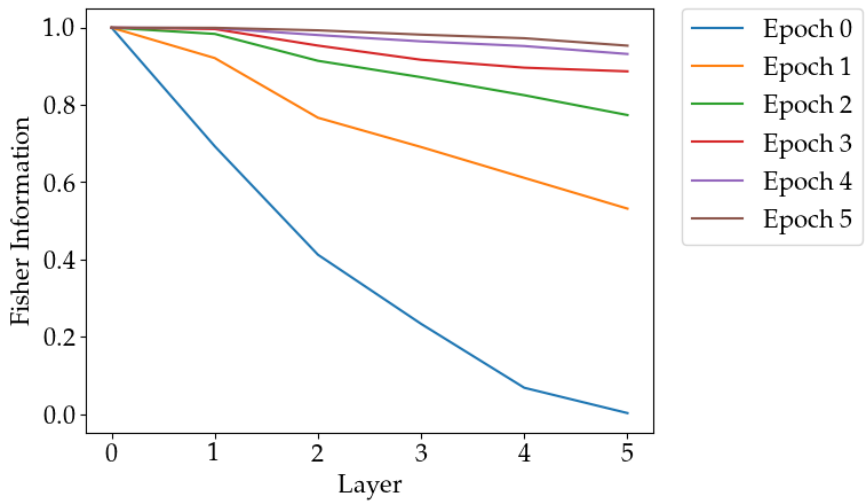

Figure 11: The (normalized) FI in each layer of the ANN. The different curves denote different stages (epochs) of training.

By using a more elaborate architecture and training for a longer time, this FI in the output layer gets even closer to 100% of the FI of the training data, however, the simulation would become very time-consuming, without adding any new insights.

5.3.5 Error Detection with Fisher Information

As shown above, when the weights and biases of an ANN are chosen randomly, most of the initial FI cannot pass through the network. This suggests that a deficient layer, where the parameters are not chosen appropriately, acts as a sink for the FI. This assumption is tested by tracking the FI in a well-trained ANN and afterwards adding a random perturbation to one of its weight matrices. We expect that from the FI flow we can locate the layer where the perturbation is added.

As for the well-trained ANN, we use again the data and the architecture from Table 10. We introduce an artificial error by adding a random matrix to the weight matrix that connects the first and the second hidden layer. Each entry of the random matrix is independently

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

drawn from a standard normal distribution, although this choice is completely generic and any other perturbation would work as well. A scheme of the network with a deficient layer is provided by Fig. 12.

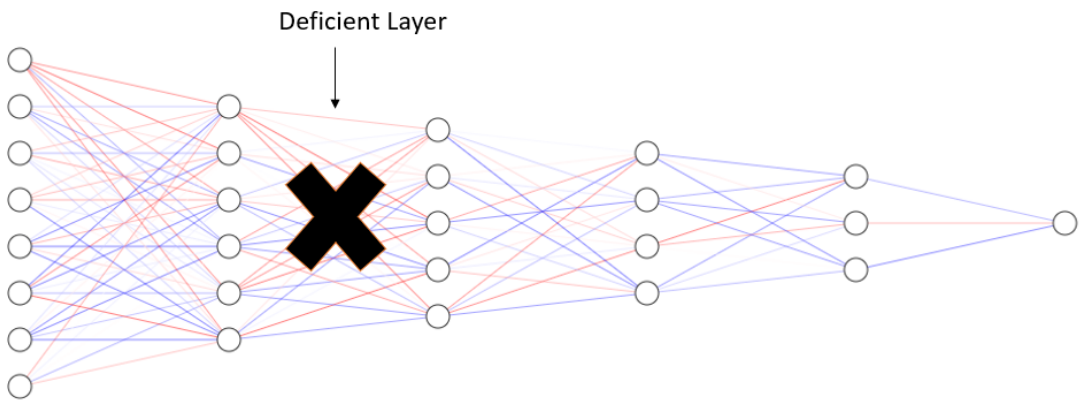

Figure 12: A schematic plot of the ANN with a random perturbation of the weights between the first and the second hidden layer. For better visualization, the number of artificial neurons in this image does not match with that of the ANN we used for the experiment.

The layer-wise FIs from the well-trained and the perturbed ANN are compared in Fig. 13.

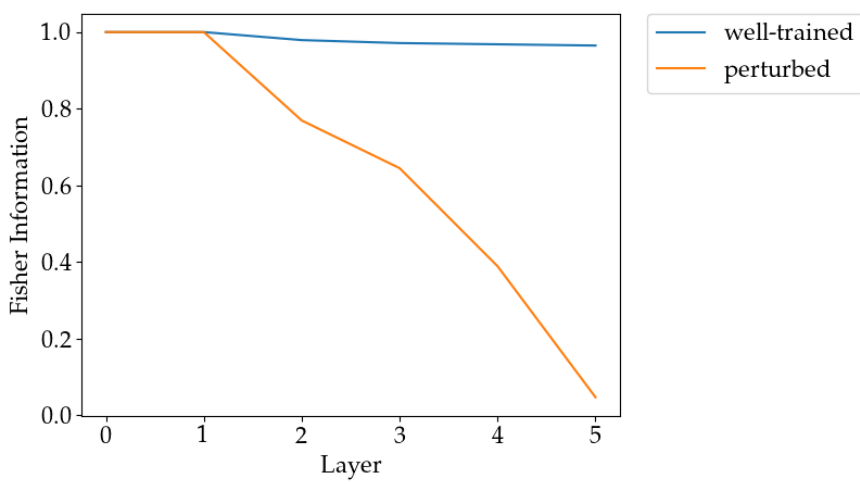

Figure 13: The (normalized) FI in each layer of the ANN. The blue curve represents the FI flow through the well-trained ANN and the orange curve that of the ANN with a perturbation of the weights after the first hidden layer.

One can clearly locate the deficient layer visually, as the one where the FI starts to decrease. One caveat with this approach is that there are situations where it is not possible to detect the exact position of the perturbation. Consider a network with two layers that are connected by an invertible map, i.e., the weight matrix and the activation function are

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

invertible. An example for such a network is given in Sec. 5.3.2. The FI is then necessarily the same in both of these layers. If the error is located between these layers, the FI will decrease only after the next non-invertible transformation is applied.

6 Quantifying the Performance of Neural Network Classification Models

The results from Sec. 2.3 are now applied to ANNs that are trained to perform classification tasks. For simplicity, we focus on a basic problem of binary classification. In analogy to Sec. 5.3.1, the inequality from Sec. 2.3.2 is applied to the layers of the ANN. These methods enable us to analyse the performance of individual layers of ANNs trained for classification.

In our basic implementation, the ANN receives data sampled from two normal distributions $p_0(\boldsymbol{x})$ and $p_1(\boldsymbol{x})$ with the same covariance, only differing in their mean values. The network is then trained to assign an observation to the distribution from which it was drawn, or equivalently, to map the observation to its distributions mean. On the basis of Sec. 2.3 we formulate this classification task as a hypothesis test. For that sake we denote the hypothesis of an observation being generated from the distribution $p_{0/1}$ by $H_{0/1}$.

6.1 Approximately Achieving the Bayes Error

By estimating the density in the output of the ANN trained for the decision task, the Bayes error B^{out} can be estimated and compared to the one of the input data B^{in} ($B^{\text{out}} \geq B^{\text{in}}$, see Sec. 2.3.2). The estimation is performed with MLE or histogram fits, depending on the width of the respective layer. If errors (approximately) match, the ANN performs at (close to) the optimum. The data set consists of N_{data} observations from N_{nodes}-dimensional normal distributions p_0 and p_1, respectively, where each component of the random vector is i.i.d. $N_{\text{data}}/2$ observations are drawn from p_0 and the other half from p_1.
The activations in the output layer should not return a binary value that decides between H_0 and H_1 but rather give information about the probability of the hypothesis being true. This is achieved, e.g., by a continuous number between 0 and 1, where 0 corresponds to $H_{0/1}$ being certainly true, or with a so-called 'one-hot encoding' scheme (see [2] for further details), where instead of one number a vector with two components is used to differentiate between the hypotheses. $(1,0)^T$ would correspond to H_0 and $(0,1)^T$ to H_1 being certainly true, while any other configuration, e.g., $(1/2,1/2)^T$ corresponds to some mixture, where the entries (probabilities) add up to one[15]. Hence in this case, the ANN architecture requires N_{nodes} nodes in the input layer and two nodes in the output layer. As the dimension is sufficiently low and samples of arbitrary size N_{data} can be generated, the density of the

[15]The activation function in the output layer is a so-called softmax, which ensures that the property that all the vector's components add up to one is fulfilled.

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

transformed data in the output layer is estimated with histograms (see Sec. 3.2.1). $P_{\text{err}}^{\text{in}}$ and $P_{\text{err}}^{\text{out}}$ are then calculated via Eq. (39).

6.2 The Propagation of the Bayes Error in Deep Neural Networks

In analogy to the FII in ANNs trained for regression, the Bayes error flow through the layers of a network obeys an inequality (see Sec. 2.3.2)

$$B_{N_l} \geq B_{N_l-1} \geq \cdots \geq B_1 \geq B_0 \ , \tag{87}$$

where the index denotes the layer of the ANN and B_{0/N_l} corresponds to the Bayes error in the input/output layer $B_{\text{in/out}}$. The difficulty lies in estimating the densities in the hidden layers when their number of nodes is not close to one. We make the assumption of the densities in the hidden layers being approximately normally distributed and estimate them with a MLE method (see Sec. 3.2.2). Mind that in the analysis of the FI propagation, we did not rely on MLE, as is in general not sufficiently accurate for this purpose. The reason why it is deployed in the discrete case is that the Bayes error, unlike the Fisher information, is not very sensitive to small noise of the density as it does not involve derivatives that need to be approximated with a finite difference scheme. The densities $p_0(\boldsymbol{x})$ and $p_1(\boldsymbol{x})$ can often be clearly differentiated, unlike $p(\boldsymbol{x}|\theta - \Delta\theta)$ and $p(\boldsymbol{x}|\theta + \Delta\theta)$ in a regression task. Also, the concept of LFI does not generalize to the BE and thus we cannot find a lower bound that is easy to calculate. To visualize the propagation of the BE through an ANN, we train a network to perform a binary classification task. The generation of the training data set and details about the ANN are listed in Table 13. Fig. 14 shows the propagation of the BE through the layers of the ANN

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

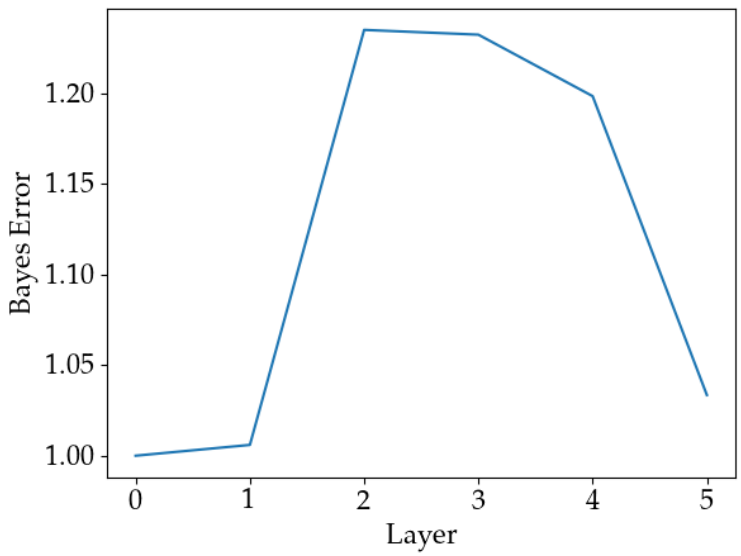

Figure 14: The BE (normalized with respect to the BE in the input layer) in each layer of the ANN trained to perform a classification task. Ideally, the BE would be the same in each layer, while in practice, it increases slightly.

From Eq. (87) we would expect the curve to be monotonically increasing. However, due to the inaccuracies of the MLE, the accuracy we can achieve is restricted. Therefore, by analysing the BE of an ANN, we cannot differentiate between networks that are very well-trained. In the early stages of training, however, the increase of the BE in the layers is strong and can thus be detected with sufficient accuracy. This issue is addressed in the section below.

6.2.1 Tracking the Bayes Error During the Training

We train again an ANN to perform a simple, binary classification task. Further details concerning the experimental setup are provided by Table. 14. After each epoch of training, the BE flow through the layers is calculated. The result is shown in Fig. 15

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

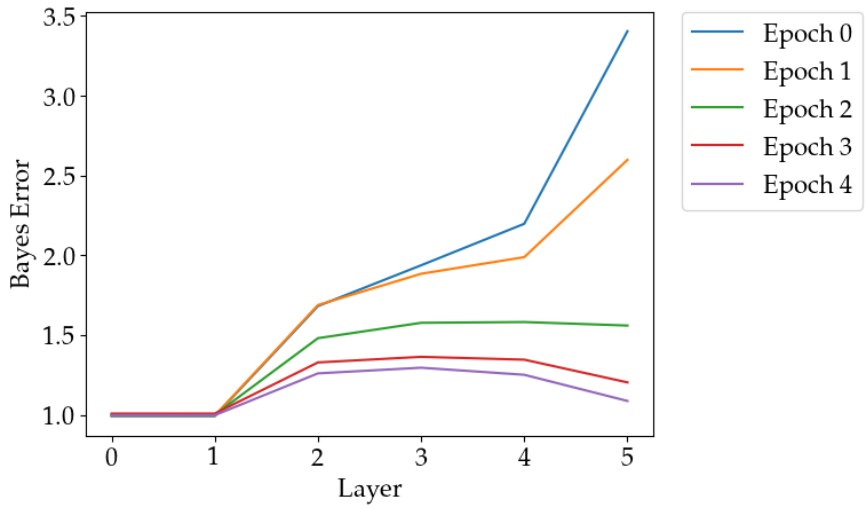

Figure 15: The BE (normalized with respect to the BE in the input layer) in each layer. The different curves correspond to different stages of training, with 'Epoch 0' denoting the ANN before any training happened.

Despite the difficulties in calculating the BE, we were able to clearly visualize the different stages of training.

6.2.2 Error Detection with the Bayes Error

It was shown in Sec. 6.2 that the Bayes error stays almost constant when it propagates through a well-trained ANN. We expect that when the parameters in one of the layers in this network are perturbed by adding a random matrix to the weight matrix of this layer, the Bayes error is increased after this perturbation. The deficient layer is thus expected to act as a source for Bayes error. The analysis in this section is closely related to that of Sec. 5.3.5. Figure 12 offers a visualization of the network.

The specifications of the well-trained ANN are listed in Table 15. In Fig. 16 we plot the BE propagation through the ANN, for both the well-trained and the perturbed case.

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

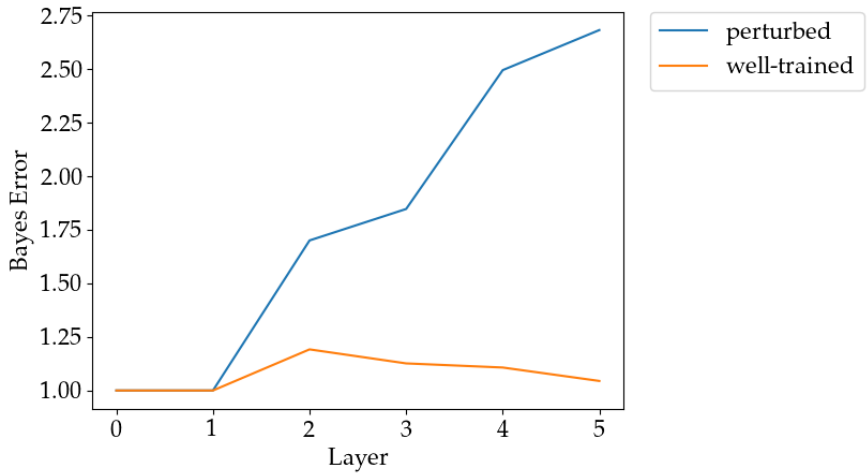

Figure 16: The BE (normalized with respect to the BE in the input layer) in each layer. The orange curve corresponds to the well-trained ANN and the blue curve to the ANN with a random perturbation added to the weights between the first and the second hidden layer. One can clearly see that after the perturbation, the BE in the faulty network starts to increase significantly, while that in the well-trained network remains approximately constant.

It might not always be possible to detect the exact location of the deficient layer with this method. We know from Sec. D.1 that the BE stays constant under invertible transformations. Thus, when several layers of the ANN are connected by invertible transformations, we can only narrow down the location of the deficiency to one of these layers. Another problem is that when the perturbation is weak, the BE might not show a strong increase immediately, but will rather start to grow steadily from layer to layer. The strong numerical fluctuations in the calculation of the BE might be of the same order as the increase due to imperfections.

7 Summary and Outlook

We studied Deep Neural Networks from an estimation theoretic perspective, using the quantity of Fisher Information. The non-trivial task of estimating the FI from arbitrarily distributed, high-dimensional data sets was met by introducing the LFI maximization algorithm. Not only could we measure the performance of the ANN by comparing the variance in its output layer to the Cramér-Rao Lower Bound, but extracting the data in each of the network's layers allowed us to formulate a Fisher Information Inequality for any DNN regression model. This approach enabled us to interpret the FI as a quantity that flows through the layers of an ANN. We used this result to visualize the training of a network by tracking the FI flow at several stages of the training process. Faulty weights and biases manifested as sinks for the FI flow. We managed to localise the position of a perturbed weight matrix directly from the FI. Moreover, we could compute a lower bound of the MSE loss function using FI, which allowed us to completely dispense with the validation data set, as a crossing of the training loss with this lower bound yields a suitable condition for early stopping.

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

One caveat of the techniques mentioned above is that they work exclusively for regression models. Thus, we sought for a practicable extension to classification tasks. Using the Bayes Error instead of the Fisher Information allowed for an analysis in close analogy to that of regression problems. However, the Bayes Error turned out to be more difficult to compute, leading to less accurate results. Moreover, identifying a lower bound for loss functions suitable for classification, such as the Cross Entropy Error, is a task that has yet to be done.

The following ideas could be implemented in future work:
In addition to the MSE, one could predict similar bounds for different loss functions, both for regression and classification problems. These bounds might not directly relate to FI but to different quantities that yet need to be identified.
One important step in opening the black box of deep neural networks would be to introduce a unified theory that allows to deal with regression and classification models based on one generalised framework. This task could, e.g., be met by identifying a generalization of FI that applies to classification models in the same way as it does to regression models.
One also could explicitly incorporate the FI into a loss function, such that the ANN is directly trained to maximize the FI. Although the MSE works sufficiently well for most regression models, we expect there to be various situations where this new loss function gives rise to better performance of the network, possibly in a smaller training duration. Instead of the exact FI, one could use the LFI as an approximation, since this quantity can be estimated much faster and more accurately.

8 Acknowledgements

First I would like to thank my supervisors Stefan Rotter and Lukas Rachbauer for giving me the opportunity to explore the vast fields of Deep Learning and Mathematical Statistics and the freedom to bring my own interests into this thesis. Without exception, I could count on their help and received valuable inputs, whenever I faced a problem.
I want to thank my colleagues for always being supportive and sharing their experiences with me. Special thanks to Günther Hackl, Dorian Bouchet and Linara Adylova for all the fruitful discussions.
My utmost gratitude to my family, my friends and my girlfriend for their constant support and encouragement.

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

Appendices

A Estimation Theory

A.1 Fisher Information under Reparametrization

As demonstrated in Sec. 2.1.3, the FI depends on the parametrization [26]. We present a derivation of the transformation behaviour of the FI under an arbitrary reparametrization $\theta = \theta(\xi)$. Starting off with the definition of the FI

$$I_\theta(\theta) = \int d\boldsymbol{x}\, p(\boldsymbol{x}|\theta)\, (\partial_\theta \log(p(\boldsymbol{x}|\theta)))^2 \ , \tag{88}$$

we derive an expression for the FI $I_\xi(\xi)$ under the reparametrization

$$\theta = \theta(\xi) \ . \tag{89}$$

With the chain rule, the derivative ∂_ξ is rewritten as

$$\partial_\xi = \frac{\partial\theta(\xi)}{\partial\xi}\partial_\theta \ , \tag{90}$$

which yields

$$I_\xi(\xi) := \int d\boldsymbol{x}\, p(\boldsymbol{x}|\theta)\, (\partial_\xi \log(p(\boldsymbol{x}|\theta(\xi))))^2 = \int d\boldsymbol{x}\, p(\boldsymbol{x}|\theta) \left(\frac{\partial\theta(\xi)}{\partial\xi}\partial_\theta \log(p(\boldsymbol{x}|\theta))\right)^2 \tag{91}$$

and thus

$$I_\xi(\xi) = I_\theta(\theta(\xi)) \left(\frac{\partial\theta(\xi)}{\partial\xi}\right)^2 \ . \tag{92}$$

A.2 Generating Distributions with Arbitrarily Shaped Fisher Information

The result from A.1 is particularly useful, as it allows to taylor any desired shape of the FI as a function of the parameter. The goal is to find a transformation $\theta = \theta(\xi)$, such that the reparametrized FI takes on some desired form $I_\xi(\xi) = Q(\xi)$. Inserting this definition into Eq. (92) yields the first order differential equation

$$\partial_\xi\theta(\xi) = \pm\sqrt{\frac{Q(\xi)}{I_\theta(\theta(\xi))}} \ . \tag{93}$$

For simplicity, we choose the initial distribution to be a univariate normal distribution $N(\mu(\theta) = \theta, \sigma^2)$ with FI $I(\theta) = 1/\sigma^2$, which is a constant function of the parameter θ.

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

Equation (93) then reduces to

$$\partial_\xi \theta(\xi) = \pm\sigma\sqrt{Q(\xi)}\ , \tag{94}$$

which is solved by integration

$$\theta(\xi) = \pm\sigma \int d\xi \sqrt{Q(\xi)}\ . \tag{95}$$

As long as this integral can be calculated, distributions with arbitrarily shaped Fisher informations can be found.

A.3 General Properties of the Fisher Information

Depending on the shape of the density function and the way it is available (e.g., analytically, in terms of histograms, etc.), utilizing different expressions for the FI might be convenient for computational reasons, but also for developing an intuitive understanding for this quantity. When dealing with integrals involving score functions, a helpful identity is given by

$$\mathrm{E}[S(\theta)] = \int d\boldsymbol{x}\, p(\boldsymbol{x}|\theta) \left(\frac{\partial_\theta p(\boldsymbol{x}|\theta)}{p(\boldsymbol{x}|\theta)}\right) = \int d\boldsymbol{x}\, \partial_\theta p(\boldsymbol{x}|\theta) = \partial_\theta \int d\boldsymbol{x}\, p(\boldsymbol{x}|\theta) = \partial_\theta 1 = 0\ , \tag{96}$$

i.e., the expectation of the score function vanishes. For the third equality sign in Eq. (96) we assumed that the regularity conditions from Sec. 2.1.2 hold. Exchanging the integral with the derivative allows to derive an alternative expression of the FI by partial integration

$$\begin{aligned}
-\mathrm{E}[\partial_\theta^2 \log(p(\boldsymbol{x}|\theta))] &= -\int d\boldsymbol{x}\, p(\boldsymbol{x}|\theta)\partial_\theta^2 \log(p(\boldsymbol{x}|\theta)) = \\
&\quad -\int d\boldsymbol{x}\, p(\boldsymbol{x}|\theta) \left(\frac{\partial_\theta^2 p(\boldsymbol{x}|\theta)}{p(\boldsymbol{x}|\theta)} - \frac{(\partial_\theta p(\boldsymbol{x}|\theta))^2}{p(\boldsymbol{x}|\theta)^2}\right) = \\
&\quad -\int d\boldsymbol{x}\, \partial_\theta^2 p(\boldsymbol{x}|\theta)\ + \int d\boldsymbol{x} \frac{(\partial_\theta p(\boldsymbol{x}|\theta))^2}{p(\boldsymbol{x}|\theta)} = -\partial_\theta^2 \int d\boldsymbol{x}\, p(\boldsymbol{x}|\theta)\ + I(\theta) \\
&= I(\theta)\ .
\end{aligned} \tag{97}$$

From this expression we conclude that the FI is a measure for the expected curvature of the log likelihood function with respect to the parameter θ. When the log likelihood landscape is flat, the FI is small and estimating the parameter is difficult. For large curvatures, which occur for example when the density has a pronounced peak at some location parameter $x = \theta$, the FI is large and the parameter can be estimated more precisely. In the extreme case, where $p(x|\theta) \sim \delta(x-\theta)$ is a delta distribution peaked at $x = \theta$, the FI diverges, since knowledge of the distribution determines θ with perfect precision.

For numerical calculations another expression of the FI

$$\left(\partial_\theta \sqrt{p(\boldsymbol{x}|\theta)}\right)^2 = \frac{1}{4}\frac{(\partial_\theta p(\boldsymbol{x}|\theta))^2}{p(\boldsymbol{x}|\theta)} \implies I(\theta) = 4\int d\boldsymbol{x}\, \left(\partial_\theta \sqrt{p(\boldsymbol{x}|\theta)}\right)^2\ , \tag{98}$$

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

is particularly useful when the density is given in terms of histograms (see Sec. 3.3.1).
For random variables

$$\boldsymbol{x} = \begin{pmatrix} \boldsymbol{x}_1 \\ \boldsymbol{x}_2 \end{pmatrix} \tag{99}$$

with multiple components it can be shown that the FI obeys a 'chain rule', which is derived in Sec. A.3.1. At first we introduce the notion of joint, marginal and conditional Fisher informations. The joint probability density of $\boldsymbol{x}_1$ and $\boldsymbol{x}_2$ is denoted as $p(\boldsymbol{x}_1, \boldsymbol{x}_2|\theta)$ and the FI is calculated according to

$$I_{\boldsymbol{x}_i,\boldsymbol{x}_j}(\theta) = \int d\boldsymbol{x}_1 d\boldsymbol{x}_2\, p(\boldsymbol{x}_1, \boldsymbol{x}_2|\theta) \left(\frac{\partial_\theta p(\boldsymbol{x}_1, \boldsymbol{x}_2|\theta)}{p(\boldsymbol{x}_1, \boldsymbol{x}_2|\theta)} \right)^2 . \tag{100}$$

A FI can also be defined for marginal PDFs

$$p(\boldsymbol{x}_i|\theta) := \int d\boldsymbol{x}_j p(\boldsymbol{x}_1, \boldsymbol{x}_2|\theta)$$

$$I_{x_i}(\theta) := \int d\boldsymbol{x}_i\, p(\boldsymbol{x}_i|\theta) \left(\frac{\partial_\theta p(\boldsymbol{x}_i|\theta)}{p(\boldsymbol{x}_i|\theta)} \right)^2 \tag{101}$$

and conditional PDFs

$$p(\boldsymbol{x}_i|\boldsymbol{x}_j|\theta) = \frac{p(\boldsymbol{x}_i, \boldsymbol{x}_j|\theta)}{p(\boldsymbol{x}_j|\theta)} \; ; i, j = (1, 2)$$

$$I_{\boldsymbol{x}_i|\boldsymbol{x}_j}(\theta) := \int d\boldsymbol{x}_i\, p(\boldsymbol{x}_i|\boldsymbol{x}_j|\theta) \left(\frac{\partial_\theta p(\boldsymbol{x}_i|\boldsymbol{x}_j|\theta)}{p(\boldsymbol{x}_i|\boldsymbol{x}_j|\theta)} \right)^2 . \tag{102}$$

These marginal and conditional FIs are related to the joint FI by the chain rule [9]

$$I_{\boldsymbol{x}_i,\boldsymbol{x}_j}(\theta) = I_{\boldsymbol{x}_i|\boldsymbol{x}_j}(\theta) + I_{\boldsymbol{x}_j}(\theta) . \tag{103}$$

When $\boldsymbol{x}_1$ and $\boldsymbol{x}_2$ of the random vector $\boldsymbol{x}$ are independent random variables, the joint density factorizes and equivalently, the conditional distribution $p(\boldsymbol{x}_i|\boldsymbol{x}_j|\theta)$ does not depend on the variable $\boldsymbol{x}_j$ and is equal to the marginal $p(\boldsymbol{x}_i|\theta)$. In this case the joint FI coincides with the sum of the marginal FIs $I_{\boldsymbol{x}_i,\boldsymbol{x}_j}(\theta) = I_{\boldsymbol{x}_i}(\theta) + I_{\boldsymbol{x}_j}(\theta)$. Generalizing this result to an M dimensional random vector $\boldsymbol{x}$ with independent entries $\boldsymbol{x}_i$ yields

$$I_{\boldsymbol{x}_1,\boldsymbol{x}_2,\ldots,\boldsymbol{x}_M}(\theta) = \sum_{i=1}^{M} I_{\boldsymbol{x}_i}(\theta) . \tag{104}$$

A.3.1 Chain Rule for the Fisher Information

For the FI, a so-called chain rule [9] can be derived. This result yields an understanding of the information content of marginal, conditional and joint PDFs, which is important when the

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

data is multi-dimensional. The relation between joint and conditional probability densities

$$p(\boldsymbol{x}_i|\boldsymbol{x}_j|\theta) = \frac{p(\boldsymbol{x}_i, \boldsymbol{x}_j|\theta)}{p(\boldsymbol{x}_j|\theta)} \ , \tag{105}$$

is inserted into the definition of the FI

$$I(\theta) = \int d\boldsymbol{x}\, p(\boldsymbol{x}|\theta)\, (\partial_\theta \log(p(\boldsymbol{x}|\theta)))^2 \ , \tag{106}$$

which yields

$$\begin{aligned} I_{\boldsymbol{x}_i,\boldsymbol{x}_j}(\theta) &= \int d\boldsymbol{x}\, p(\boldsymbol{x}_i, \boldsymbol{x}_j|\theta)\, (\partial_\theta \log(p(\boldsymbol{x}_i, \boldsymbol{x}_j|\theta)))^2 = \int d\boldsymbol{x}\, p(\boldsymbol{x}_i, \boldsymbol{x}_j|\theta)\, (\partial_\theta \log(p(\boldsymbol{x}_i|\boldsymbol{x}_j|\theta)p(\boldsymbol{x}_j|\theta)))^2 \\ &= \int d\boldsymbol{x}\, p(\boldsymbol{x}_i, \boldsymbol{x}_j|\theta)\, (\partial_\theta \log(p(\boldsymbol{x}_i|\boldsymbol{x}_j|\theta) + \log(p(\boldsymbol{x}_j|\theta)))^2 \\ &= I_{\boldsymbol{x}_i|\boldsymbol{x}_j}(\theta) + I_{\boldsymbol{x}_j}(\theta) + 2\int d\boldsymbol{x}\, p(\boldsymbol{x}_i, \boldsymbol{x}_j|\theta)\, (\partial_\theta \log(p(\boldsymbol{x}_i|\boldsymbol{x}_j|\theta))\, (\partial_\theta \log(p(\boldsymbol{x}_j|\theta))) \ . \end{aligned} \tag{107}$$

With $d\boldsymbol{x} := d\boldsymbol{x}_i d\boldsymbol{x}_j$ the cross term is rewritten as

$$\int d\boldsymbol{x}_j\, p(\boldsymbol{x}_j|\theta)\, (\partial_\theta \log(p(\boldsymbol{x}_j|\theta))) \int d\boldsymbol{x}_i p(\boldsymbol{x}_i|\boldsymbol{x}_j|\theta)\, (\partial_\theta \log(p(\boldsymbol{x}_i|\boldsymbol{x}_j|\theta))) \ . \tag{108}$$

The integral

$$\int d\boldsymbol{x}_j\, p(\boldsymbol{x}_j|\theta)\, (\partial_\theta \log(p(\boldsymbol{x}_j|\theta))) \ , \tag{109}$$

vanishes (see Eq. (96)). We arrive at the chain rule for FI

$$I_{\boldsymbol{x}_i,\boldsymbol{x}_j}(\theta) = I_{\boldsymbol{x}_i|\boldsymbol{x}_j}(\theta) + I_{\boldsymbol{x}_j}(\theta) \ . \tag{110}$$

A.4 Invertible Transformations are Sufficient Statistics

When $\boldsymbol{y} = f(\boldsymbol{x})$ is a bijective map between two random variables $\boldsymbol{x}$ and $\boldsymbol{y}$, there exists a unique inverse function $\boldsymbol{x} = f^{-1}(\boldsymbol{y})$. Now consider the chain of transformations

$$\boldsymbol{x} \to f(\boldsymbol{x}) \to f^{-1}(f(\boldsymbol{x})) = \boldsymbol{x} \ . \tag{111}$$

The FII for each transformed RV must fulfil

$$I_{\boldsymbol{x}} \geq I_{f(\boldsymbol{x})} \geq I_{f^{-1}(f(\boldsymbol{x}))} = I_{\boldsymbol{x}} \ , \tag{112}$$

what can only be true if both equality signs hold. Thus it can be concluded that the FI is invariant under (uniquely) invertible transformations.

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

B Mathematical Properties of Probability Densities

B.1 Finite Difference Method

When the data at hand is of the form $D(\theta)$ on a discrete grid $(\theta_1, \theta_2, ..., \theta_M)$, the derivative $\partial_\theta p(\boldsymbol{x}|\theta)$ that occurs in the definition of the FI (Eq. (2)) is approximated by a symmetric finite difference

$$\partial_\theta p(\boldsymbol{x}|\theta)\big|_{\theta=\theta_i} \approx \frac{p(\boldsymbol{x}|\theta_{i+1}) - p(\boldsymbol{x}|\theta_{i-1})}{\theta_{i+1} - \theta_{i-1}} \ . \tag{113}$$

If one is interested in approximating derivatives at the boundaries of the theta-interval, i.e., at θ_1 and θ_M, a single-sided finite difference can be used. When the parameter values are equally spaced, the grid can be denoted by $(\theta_{\min}, \theta_{\min} + \Delta\theta, ..., \theta_{\max} - \Delta\theta, \theta_{\max})$ and the numerical derivative becomes

$$\partial_\theta p(\boldsymbol{x}|\theta)\big|_{\theta=\theta_i} \approx \frac{p(\boldsymbol{x}|\theta_i + \Delta\theta) - p(\boldsymbol{x}|\theta_i - \Delta\theta)}{2\Delta\theta} \ . \tag{114}$$

To arrive at a precise estimation of the FI, the estimated density functions for neighbouring θ values need to be clearly distinguishable from each other. Thus the method of density estimation must be highly precise, which typically requires that a considerable amount of data $D(\theta)$ is available for each parameter value. On the one hand, the grid spacing $\Delta\theta$ must not be too small, in order to end up with approximated density functions whose differences are not dominated by noise. On the other hand, $\Delta\theta$ also must not be too large, in order to yield a good numerical approximation of the derivative in Eq. (114). So a trade-off has to be made between these two effects. In this thesis we analyse the FI of synthetic data sets, which means that we can generate an arbitrary amount of data for any parameter value θ. This allows us to fine-tune $\Delta\theta$ such that the derivatives converge well. In the general case where the data is limited, it might be necessary to artificially increase the amount of data by means of data augmentation.

B.2 Transformation of Expectations

When a random variable $\boldsymbol{x}$ with density $p_{\boldsymbol{x}}(\boldsymbol{x})$ is transformed according to $\boldsymbol{y} = f(\boldsymbol{x})$, the transformed density $p_{\boldsymbol{y}}(\boldsymbol{y})$ is often intractable if the transformation is not bijective. When evaluating expressions such as the LFI of the transformed data, it is not necessary to estimate $p_{\boldsymbol{y}}(\boldsymbol{y})$ since expectations of any function of $\boldsymbol{y}$

$$\mathrm{E}_{\boldsymbol{y}\sim p_{\boldsymbol{y}}(\boldsymbol{y})}\left[\Psi(\boldsymbol{y})\right] = \int d\boldsymbol{y}\, p_{\boldsymbol{y}}(\boldsymbol{y})\Psi(\boldsymbol{y}) \ , \tag{115}$$

can be directly estimated as a sample average. This is due to the fact that when a sample distributed according to $p_{\boldsymbol{x}}(\boldsymbol{x})$ is available, a sample for $p_{\boldsymbol{y}}(\boldsymbol{y})$ is generated by applying the transformation Ψ to each random variate $\boldsymbol{x}$.
While this appears to be trivial, it is an important insight, since it enables to evaluate all

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

kinds of integrals of the form Eq. (115). For instance, one can compute Fourier coefficients of the transformed density function, although the density itself is unknown.

B.3 The Moments under Affine Transformations

When an affine transformation $\boldsymbol{y} = A(\boldsymbol{x})$ acts upon an N-dimensional random variable $\boldsymbol{x}$ with density $p_{\boldsymbol{x}}(\boldsymbol{x})$, the transformed mean and the transformed covariance matrix of the M dimensional random variable $\boldsymbol{y}$ can be found directly, without evaluating Eq. (115). An affine transformation can be written as a linear transformation together with a translation

$$A(\boldsymbol{x}) = \boldsymbol{W}\boldsymbol{x} + \boldsymbol{b} \,, \tag{116}$$

where $\boldsymbol{W}$ is an $M \times N$ matrix and $\boldsymbol{b}$ and M dimensional vector. Note that this function coincides with a neural network layer with linear activation function. It turns out that the transformed mean $\boldsymbol{\mu}' = \mathrm{E}_{\boldsymbol{y}\sim p_{\boldsymbol{y}}(\boldsymbol{y})}\left[A(\boldsymbol{y})\right]$ simplifies to

$$\boldsymbol{\mu}' = \boldsymbol{W}\boldsymbol{\mu} + \boldsymbol{b} \tag{117}$$

and the covariance $\boldsymbol{\Sigma}' = \mathrm{E}_{\boldsymbol{y}\sim p_{\boldsymbol{y}}(\boldsymbol{y})}\left[(A(\boldsymbol{y}) - \boldsymbol{\mu}')^2\right]$ becomes

$$\boldsymbol{\Sigma}' = \boldsymbol{W}\boldsymbol{\Sigma}\boldsymbol{W}^T \,. \tag{118}$$

The identities Eq. (117) and Eq. (118) follow directly from the linearity of the expectation.

B.4 Singular Value Decomposition

We demonstrate the benefits of SVD based on the calculation of the LFI (see Sec. 3.5)

$$J(\theta) = \left(\partial_\theta \boldsymbol{\mu}(\theta)\right)^T \boldsymbol{\Sigma}(\theta)^{-1} \partial_\theta \boldsymbol{\mu}(\theta) \,. \tag{119}$$

In case one or more eigenvalues of $\boldsymbol{\Sigma}$ are very close to zero, even small numerical fluctuations can have a large effect on the inverse $\boldsymbol{\Sigma}^{-1}$. The basic idea to resolve this problem is to diagonalize the $N \times N$ covariance matrix $\boldsymbol{\Sigma}$. In the rotated frame where $\boldsymbol{\Sigma}$ is diagonal, the coordinate axes corresponding to small eigenvalues are said to be of little significance, as the data shows almost no spread in these directions. When evaluating the quadratic form Eq. (119) we insert orthogonal transformations

$$\boldsymbol{\Sigma} = \boldsymbol{T}^T\boldsymbol{T}\boldsymbol{\Sigma}\boldsymbol{T}^T\boldsymbol{T} = \boldsymbol{T}^T\boldsymbol{\Sigma}'\boldsymbol{T} \,, \tag{120}$$

where $\boldsymbol{\Sigma}' := \boldsymbol{T}\boldsymbol{\Sigma}\boldsymbol{T}^T$ is diagonal. The inverse becomes

$$\boldsymbol{\Sigma}^{-1} = \boldsymbol{T}^T \left(\boldsymbol{\Sigma}'\right)^{-1} \boldsymbol{T} \tag{121}$$

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

and thus

$$J(\theta) = (\partial_\theta \boldsymbol{\mu}(\theta))^T \boldsymbol{T}^T \boldsymbol{\Sigma}'(\theta)^{-1} \boldsymbol{T} \partial_\theta \boldsymbol{\mu}(\theta)$$
$$= (\partial_\theta \boldsymbol{T}\boldsymbol{\mu}(\theta))^T \boldsymbol{\Sigma}'(\theta)^{-1} \partial_\theta \boldsymbol{T}\boldsymbol{\mu}(\theta) = (\partial_\theta \boldsymbol{\mu}'(\theta))^T \boldsymbol{\Sigma}'(\theta)^{-1} \partial_\theta \boldsymbol{\mu}'(\theta) \ , \quad (122)$$

with $\boldsymbol{\mu}' := \boldsymbol{T}\boldsymbol{\mu}$. In a next step we replace every component of $\boldsymbol{\Sigma}'^{-1}$ that is the inverse of an eigenvalue below a certain threshold $\lambda_i < \lambda_{\min}$ by $\lambda_i = 0$. This means that the term $(\partial_\theta \boldsymbol{\mu}')_i^T \left(\boldsymbol{\Sigma}'(\theta)^{-1}\right)_{i,i} (\partial_\theta \boldsymbol{\mu}')_i$ does not contribute to the quadratic form $J(\theta)$. This approach is closely related to concept of Principal Component Analysis (PCA) [27] where the dimensionality is reduced by confining the coordinate system to the subspace spanned by the directions with eigenvalues that are significantly larger than zero. Note that the covariance matrix is symmetric and positive semidefinite, so the eigenvalues are real and non-negative.

B.5 Transformation of Probability Densities

B.5.1 Neural Network Transformations

For a random variable $\boldsymbol{x}$ with density $p_{\boldsymbol{x}}(\boldsymbol{x})$ we seek for the transformed density $p_{\boldsymbol{y}}(\boldsymbol{y})$ that is induced by a transformation of the form

$$\boldsymbol{y} = f(\boldsymbol{x}) = \phi_N \circ \boldsymbol{A}_N \circ \cdots \circ \phi_2 \circ \boldsymbol{A}_2 \circ \phi_1 \circ \boldsymbol{A}_1(\boldsymbol{x}) \ , \quad (123)$$

with matrices $\boldsymbol{A}_i$ that are not necessarily quadratic. The aim is now to replace the matrices by invertible matrices $\boldsymbol{A}_i^*$

$$\boldsymbol{y}^* = f^*(\boldsymbol{x}) = \begin{pmatrix} \boldsymbol{y} \\ \boldsymbol{v} \end{pmatrix} = \phi_N \circ \boldsymbol{A}_N^* \circ \cdots \circ \phi_2 \circ \boldsymbol{A}_2^* \circ \phi_1 \circ \boldsymbol{A}_1^*(\boldsymbol{x}) \ . \quad (124)$$

Since the first M components $\boldsymbol{y} = (\boldsymbol{y}_1^*, ..., \boldsymbol{y}_M^*)^T$ of the transformation $\boldsymbol{y}^*$ coincide with the original transformation $\boldsymbol{y}$, these components of the transformed random vector $\boldsymbol{y}^*$ are distributed according to $p_{\boldsymbol{y}}(\boldsymbol{y})$. When the lower components $\boldsymbol{v} = (\boldsymbol{y}_{M+1}^*, ..., \boldsymbol{y}_N^*)^T$, which act as auxiliary variables, are integrated out, the remaining marginal density

$$p_{\boldsymbol{y}}(\boldsymbol{y}) = \int d\boldsymbol{v}\, p_{\boldsymbol{x}}\left(f^{*-1}(\boldsymbol{y}^*)\right) \frac{1}{|\det\left((\boldsymbol{D}f^*)\left(f^{*-1}(\boldsymbol{y}^*)\right)\right)|} \quad (125)$$

is exactly the density of interest $p_{\boldsymbol{y}}(\boldsymbol{y})$. The remaining task is to choose the matrices $\boldsymbol{A}_i^*$ such that $f^*(\boldsymbol{x}) = (\boldsymbol{y}, ...)^T$ and to ensure that $f^{*-1}(\boldsymbol{x})$ exists. If every ϕ_i is invertible, one finds that the $P \times Q$ matrix

$$\boldsymbol{A}_i^* = \begin{pmatrix} \boldsymbol{A}_i & \boldsymbol{0}_{Q\times N-P} \\ \boldsymbol{0}_{N-Q\times Q} & \boldsymbol{1}_{N-Q\times N-Q} \end{pmatrix} \ , \quad (126)$$

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

with $P, Q < N$ is a suitable choice. Here $\mathbf{0}$ denotes a matrix with every component equal to zero and $\mathbf{1}$ is a matrix with ones in its main diagonal and zeros elsewhere. Many other choices for the matrices $\boldsymbol{A}_i^*$ are viable as well.
Note that DNNs usually act as transformations like Eq. (123) with the single difference that, in addition to the matrices $\boldsymbol{A}_i$, the random vector is translated as a bias is added. Since translations are always invertible transformations, incorporating them in the analysis is straight forward.

B.6 Multivariate Gaussian Densities under Affine Transformations

We study the behaviour of a multivariate normal distribution under an affine transformation $\boldsymbol{y} = \boldsymbol{W}\boldsymbol{x} + \boldsymbol{b}$. The density of the N dimensional random vector $\boldsymbol{x}$ is a multivariate Gaussian

$$p_{\boldsymbol{x}}(\boldsymbol{x}) = \frac{1}{\sqrt{(2\pi)^N \ \det(\boldsymbol{\Sigma})}} \exp\left(-\frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu})^T \boldsymbol{\Sigma}^{-1} (\boldsymbol{x}-\boldsymbol{\mu})\right) . \tag{127}$$

The dimension of the transformed random vector $\boldsymbol{y}$ is denoted by M. From Sec. B.3 we know that the mean and the covariance matrix transform according to $\boldsymbol{\mu}' = \boldsymbol{W}\boldsymbol{\mu} + \boldsymbol{b}$ and $\boldsymbol{\Sigma}' = \boldsymbol{W}\boldsymbol{\Sigma}\boldsymbol{W}^T$. One can show that an affine transformation of a Gaussian density results in another Gaussian density. As the normal distribution is uniquely determined by its mean and its covariance matrix, we immediately arrive at an expression for the transformed density

$$p_{\boldsymbol{y}}(\boldsymbol{y}) = \frac{1}{\sqrt{(2\pi)^M \ \det(\boldsymbol{W}\boldsymbol{\Sigma}\boldsymbol{W}^T)}} \exp\left(-\frac{1}{2}(\boldsymbol{y}-\boldsymbol{W}\boldsymbol{\mu}-\boldsymbol{b})^T \boldsymbol{W}^{-T}\boldsymbol{\Sigma}^{-1}\boldsymbol{W} (\boldsymbol{y}-\boldsymbol{W}\boldsymbol{\mu}-\boldsymbol{b})\right) . \tag{128}$$

C Fisher Information in Neural Network Regression Models

C.1 Analytic Fisher Information in the First Hidden Layer

For simple input distributions and invertible activation functions the Fisher information in the first hidden layer can often be found analytically.
Consider an ANN trained to predict the parameter θ from input data with the density

$$p(\boldsymbol{x}|\theta) = \frac{1}{\sqrt{\det(2\pi\boldsymbol{\Sigma})}} \exp\left(-\frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu}(\theta))^T\boldsymbol{\Sigma}^{-1}(\boldsymbol{x}-\boldsymbol{\mu}(\theta))\right) . \tag{129}$$

Transferring the data to the first hidden layer means that an affine transformation $\boldsymbol{x}' = \boldsymbol{W}\boldsymbol{x} + \boldsymbol{b}$ acts upon the random vector $\boldsymbol{x}$, followed by an activation function ϕ. If ϕ is invertible, the FI before and after its application are equal. It is therefore sufficient to calculate the FI for the distribution $p'(\boldsymbol{x}'|\theta)$ of $\boldsymbol{x}'$. From Sec. B.6 we find that the mean of the transformed data reads $\boldsymbol{\mu}'(\theta) = \boldsymbol{W}\boldsymbol{\mu}(\theta) + \boldsymbol{b}$, with $\boldsymbol{\mu}(\theta) = \theta \times (1, 1, ..., 1)^T$ and the

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

covariance matrix becomes $\boldsymbol{\Sigma}' = \boldsymbol{W}\boldsymbol{\Sigma}\boldsymbol{W}^T$. The FI in the first hidden layer I_{FHL} is calculated with the help of Eq. (6)

$$I_{\text{FHL}} = \left(\partial_\theta \boldsymbol{\mu}'(\theta)\right)^T \left(\boldsymbol{\Sigma}'\right)^{-1} \partial_\theta \boldsymbol{\mu}'(\theta) \ . \tag{130}$$

C.2 A Markov Chain Monte Carlo Algorithm

When the input density is known analytically, the transformed densities in deep hidden layers are evaluated by solving the integral in Eq. (62). We compute the integral by a Markov Chain Monte Carlo method [28]. These methods were outperformed by the numerical approximations used in Sec. 5.3.3, however, they might yield additional insights into the behaviour of FI in ANNs in the future. Additionally, further developing the Monte Carlo approach could yield an efficient tool that is more accurate than LFI maximization algorithms.
Recall that the PDF under neural network type transformations introduced in Sec. 3.4.3 transforms according to

$$p_{\boldsymbol{y}}(\boldsymbol{y}|\theta) = \int d\boldsymbol{v}\, p_{\boldsymbol{x}}\left(f^{*-1}(\boldsymbol{y},\boldsymbol{v})|\theta\right) \frac{1}{|\det\left(\left(\boldsymbol{D}f^*\right)\left(f^{*-1}(\boldsymbol{y},\boldsymbol{v})\right)\right)|} \ . \tag{131}$$

In ANNs that are used to process high dimensional, real-world data, the amount of variables that are integrated out can become arbitrarily large, thus Riemann integration on a regular grid is often not feasible. Monte Carlo integration is often used in such a situation. Based on Sec. 3.1, we aim rewriting integrals as expectation values. For simplicity, the notations $g_{\boldsymbol{y}}(\boldsymbol{v},\theta) \coloneqq p_{\boldsymbol{x}}\left(f^{*-1}(\boldsymbol{y},\boldsymbol{v})|\theta\right)$ and $q_{\boldsymbol{y}}(\boldsymbol{v}) \coloneqq |\det\left(\left(\boldsymbol{D}f^*\right)\left(f^{*-1}(\boldsymbol{y},\boldsymbol{v})\right)\right)|^{-1}$ are introduced. The transformed density becomes

$$p_{\boldsymbol{y}}(\boldsymbol{y}|\theta) = \int d\boldsymbol{v}\, g_{\boldsymbol{y}}(\boldsymbol{v},\theta)\, q_{\boldsymbol{y}}(\boldsymbol{v}) \ . \tag{132}$$

As a density function, $p_{\boldsymbol{x}}$ is non-negative and so is $g_{\boldsymbol{y}}(\boldsymbol{v},\theta)$. The normalization $\tilde{g}_{\boldsymbol{y}}(\boldsymbol{v},\theta) \coloneqq g_{\boldsymbol{y}}(\boldsymbol{v},\theta)/A$ with $A = A(\boldsymbol{y}) \coloneqq \int d\boldsymbol{v} g_{\boldsymbol{y}}(\boldsymbol{v},\theta)$ allows to interpret $\tilde{g}_{\boldsymbol{y}}(\boldsymbol{v},\theta)$ as a density function as well. Thus $p_{\boldsymbol{y}}(\boldsymbol{y}|\theta)$ can be brought to the form of an expectation value

$$p_{\boldsymbol{y}}(\boldsymbol{y}|\theta) = A \int d\boldsymbol{v}\, \tilde{g}_{\boldsymbol{y}}(\boldsymbol{v},\theta)\, q_{\boldsymbol{y}}(\boldsymbol{v}) \approx \frac{A}{N} \sum_{\boldsymbol{v}_i \sim \tilde{g}_{\boldsymbol{y}}(\boldsymbol{v},\theta)}^{N} q_{\boldsymbol{y}}(\boldsymbol{v}) \ , \tag{133}$$

which is then approximated by the average over a sample of size N. Similarly, the derivative with respect to θ can be rewritten as

$$\begin{aligned} \partial_\theta p_{\boldsymbol{y}}(\boldsymbol{y}|\theta) &= \int d\boldsymbol{v}\, \left(\partial_\theta g_{\boldsymbol{y}}(\boldsymbol{v},\theta)\right) q_{\boldsymbol{y}}(\boldsymbol{v}) = \\ &= \int d\boldsymbol{v}\, g_{\boldsymbol{y}}(\boldsymbol{v},\theta)\, q_{\boldsymbol{y}}(\boldsymbol{v}) \frac{\partial_\theta g_{\boldsymbol{y}}(\boldsymbol{v},\theta)}{g_{\boldsymbol{y}}(\boldsymbol{v},\theta)} = \int d\boldsymbol{v}\, g_{\boldsymbol{y}}(\boldsymbol{v},\theta)\, q_z(\boldsymbol{v})\, d(\boldsymbol{v},\theta) \ , \end{aligned} \tag{134}$$

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

with $d(\boldsymbol{v},\theta) := \frac{\partial_\theta g_{\boldsymbol{y}}(\boldsymbol{v},\theta)}{g_{\boldsymbol{y}}(\boldsymbol{v},\theta)}$. With normalization the derivative reads

$$\partial_\theta p_{\boldsymbol{y}}(\boldsymbol{y}|\theta) = \frac{1}{A}\int d\boldsymbol{v}\; \tilde{g}(\boldsymbol{v},\theta)\, q_z(\boldsymbol{v})\, d(\boldsymbol{v},\theta) \approx \frac{1}{NA}\sum_{\boldsymbol{v}_i\sim\tilde{g}_{\boldsymbol{y}}(\boldsymbol{v},\theta)}^{N} q_{\boldsymbol{y}}(\boldsymbol{v})\, d(\boldsymbol{v},\theta)\ . \tag{135}$$

The normalization factor A is often intractable, however, the score function

$$S(\boldsymbol{y},\theta) = \left(\frac{\partial_\theta p_{\boldsymbol{y}}(\boldsymbol{y}|\theta)}{p_{\boldsymbol{y}}(\boldsymbol{y}|\theta)}\right)^2 = \left(\frac{\sum_{\boldsymbol{v}_i\sim\tilde{g}_{\boldsymbol{y}}(\boldsymbol{v},\theta)}^{N} q_{\boldsymbol{y}}(\boldsymbol{v})\, d(\boldsymbol{v},\theta)}{\sum_{\boldsymbol{v}_i\sim\tilde{g}_{\boldsymbol{y}}(\boldsymbol{v},\theta)}^{N} q_{\boldsymbol{y}}(\boldsymbol{v})}\right)^2 \tag{136}$$

is independent of A. The FI can then be evaluated with the method introduced in Sec. 3.1. The remaining non-trivial task is to generate a sample according to the density $\tilde{g}_{\boldsymbol{y}}(\boldsymbol{v},\theta)$. The common approach is to construct a so-called Markov chain. A detailed explanation of the Markov chain Monte Carlo method is given in [28]. We demonstrate that this approach allows to track the FI through a small ANN, trained to predict the mean of a normal distribution. Further details are given in Table 12. Figure 17 shows the layer-wise FI evaluated at a single parameter value in this network. As the input data is normally distributed, the FI in the input layer and the first hidden layer is known analytically. For the consecutive layers, the Markov chain Monte Carlo method was employed. As expected, the figure clearly shows that the FII is fulfilled and that the ANN allows most of the initial FI to pass through.

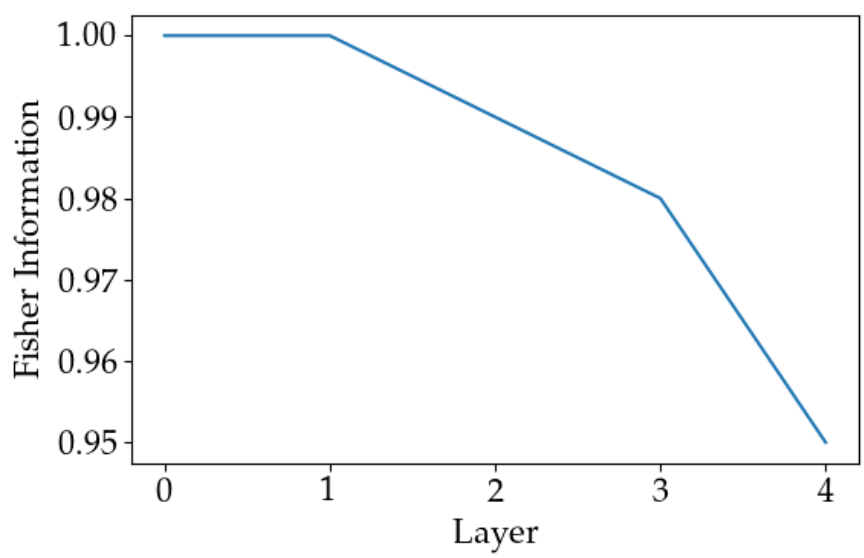

Figure 17: The blue curve depicts the normalized FI (evaluated at the parameter $\theta = 0$) in each layer of a small ANN. The FI was estimated using the Monte Carlo method.

One major caveat of this algorithm is that it is time consuming and converges slowly. This is due to the fact that the density $\tilde{g}_{\boldsymbol{y}}(\boldsymbol{v},\theta)$ is a highly complicated function, as it involves the non-linear neural network function in its argument. Furthermore, an analytical expression of the PDF of the input data must be known beforehand. Finding ways to make

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

the computation more efficient would be highly rewarding, since the curse of dimensionality is (mostly) avoided by the Monte Carlo integration. The FI could then be evaluated even for very large neural networks, if the input data set allows for a good guess of its PDF.

C.3 The Calculation of $\partial_\theta \mu$ is a Source of Error

Estimating the FI for complex, unknown distributions, is an error-prone task. We demonstrate a source of error that can have a strong effect on the outcome but may be difficult to detect. When the LFI is evaluated, either as a direct approximation of the exact FI or during a LFI maximization algorithm, the derivative $\partial_\theta \mu(\theta)$ for some estimator needs to be evaluated, as described in Sec. 3.5. This term also appears in the CRLB for biased estimators. We assume that $\mu(\theta)$, as a function of the parameter, can be decomposed into a smooth function capturing the important behaviour, together with some undesired error contribution. This error originates, e.g., from statistical fluctuations of the estimate of $\mu(\theta)$ when the sample is small (since the expectation is replaced by the sample average) or due to imperfections of the NNE (the ANN introduces bias although an unbiased estimate would yield better results). These contributions often occur on different scales, measured in units of the parameter. When an unbiased estimator is approximated, $\mu(\theta)$ will on a large scale resemble a linear function $\mu(\theta) \approx \theta$, on a smaller scale show distortions due to (mostly undesired) bias, and on a very small scale there will be fluctuations due to the finite data set. These different scales are demonstrated in Fig. 18. The plot is based on a NNE that is trained on a sparse parameter grid and therefore strongly biased. Further details of the network are given in Table 11.

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

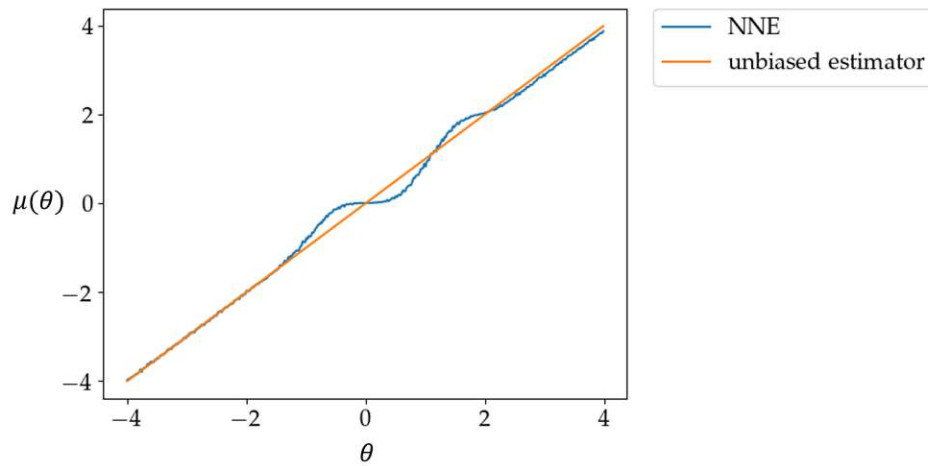

Figure 18: The blue curve depicts the expectation value $\mu(\theta)$ of a NNE, estimated as an average over a finite sample of 200 data points per parameter value. The parameter grid consists of 400 equally spaced points. The orange curve shows the optimal case, where the estimator is unbiased and the sample average exactly coincides with the expectation value. We observe that on a large scale, the blue curve approximates $\mu(\theta) = \theta$, but on a smaller scale, the bias of the estimator manifests around $\theta = 0$. On an even smaller scale, there are fluctuations due to the finite data set.

As the derivative ∂_θ is approximated with a finite difference (see Sec. B.1), the scale of the finite step $\Delta\theta$ must be compared with the scale of the error source. As a rule of thumb $\Delta\theta$ should be chosen large enough such that the random fluctuations are averaged out but not too large such that the smooth shape bias curve of the NNE is resolved. For example, if we want to evaluate the LFI in the output layer $\sigma^2(\theta) \geq \partial_\theta\mu(\theta)/I(\theta)$ around $\theta = 0$ for the example above, $\Delta\theta$ must be small enough such that the flat shape of $\mu(\theta)$ is resolved. As $\partial_\theta\mu(\theta)$ is small in that region, the variance $\sigma^2(\theta)$ can be much smaller than the unbiased CRLB $1/I(\theta)$. A large $\Delta\theta$ in that scenario results in an approximation of $\partial_\theta\mu(\theta)$ that is too large and thus the estimator seems to perform much better than it is allowed by the CRI.

D Bayes Error in Neural Network Classification Models

D.1 The Bayes Error under Invertible Transformations

Like the FI, the BE is invariant under any transformation of the random variable $\boldsymbol{y} = f(\boldsymbol{x})$ that is invertible, i.e., $f^{-1}(\boldsymbol{y}) = \boldsymbol{x}$ is uniquely defined. The transformed PDF is given by the change of variables formula introduced in Sec. 3.4.1

$$p_{\boldsymbol{y}}(\boldsymbol{y}) = p_{\boldsymbol{x}}\left(f^{-1}(\boldsymbol{y})\right) \frac{1}{\left|\det\left(\left(\boldsymbol{D}f\right)\left(f^{-1}(\boldsymbol{y})\right)\right)\right|} \,. \tag{137}$$

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

Consider two hypotheses H_0 and H_1 with corrsponding PDFs $p_{0,\boldsymbol{x}}$ and $p_{1,\boldsymbol{x}}$. The BE reads

$$B_{\boldsymbol{x}}^{\text{err}} = \int d\boldsymbol{x} \, \min \left\{ p_{0,\boldsymbol{x}} \left(\boldsymbol{x} \right), p_{1,\boldsymbol{x}} \left(\boldsymbol{x} \right) \right\} \tag{138}$$

and the BE of the transformed variable is given by

$$\begin{aligned} B_{\boldsymbol{y}}^{\text{err}} &= \int d\boldsymbol{y} \, \min \left\{ p_{0,\boldsymbol{y}} \left(\boldsymbol{y} \right), p_{1,\boldsymbol{y}} \left(\boldsymbol{y} \right) \right\} \\ &= \int \frac{d\boldsymbol{y}}{\left| \det \left(\left(\boldsymbol{D} f \right) \left(f^{-1}(\boldsymbol{y}) \right) \right) \right|} \min \left\{ p_{0,\boldsymbol{x}} \left(f^{-1}(\boldsymbol{y}) \right), p_{1,\boldsymbol{x}} \left(f^{-1}(\boldsymbol{y}) \right) \right\} \ . \end{aligned} \tag{139}$$

This expression, however, can be rewritten in terms of the original coordinate $\boldsymbol{x}$ as the volume element $\frac{d\boldsymbol{y}}{|\det((\boldsymbol{D}f)(f^{-1}(\boldsymbol{y})))|}$ is equal to the original volume element $d\boldsymbol{x}$, thus

$$\begin{aligned} &\int \frac{d\boldsymbol{y}}{\left| \det \left(\left(\boldsymbol{D} f \right) \left(f^{-1}(\boldsymbol{y}) \right) \right) \right|} \min \left\{ p_{0,\boldsymbol{x}} \left(f^{-1}(\boldsymbol{y}) \right), p_{1,\boldsymbol{x}} \left(f^{-1}(\boldsymbol{y}) \right) \right\} \\ &\qquad = \int d\boldsymbol{x} \, \min \left\{ p_{0,\boldsymbol{x}} \left(\boldsymbol{x} \right), p_{1,\boldsymbol{x}} \left(\boldsymbol{x} \right) \right\} = B_{\boldsymbol{x}}^{\text{err}} \ . \end{aligned} \tag{140}$$

Hence we have $B_{\boldsymbol{x}}^{\text{err}} = B_{\boldsymbol{y}}^{\text{err}}$.

D.2 Bayes Error Inequality

We show that the BE cannot decrease under the transformations introduced in Sec. 3.4.3. Any ANN used within the scope of this thesis can be expressed as a composition of functions

$$\boldsymbol{y} = f(\boldsymbol{x}) = \phi_N \circ A_N \circ \cdots \circ \phi_2 \circ A_2 \circ \phi_1 \circ A_1(\boldsymbol{x}) \ , \tag{141}$$

where ϕ_i are arbitrary non-linear activation functions and A_i are affine transformations. Recall from Eq. (62) that the transformed PDF reads

$$p_{\boldsymbol{y}}(\boldsymbol{y}) = \int d\boldsymbol{v} \, p_{\boldsymbol{x}} \left(f^{*-1}(\boldsymbol{y}^*) \right) \frac{1}{\left| \det \left(\left(\boldsymbol{D} f^* \right) \left(f^{*-1}(\boldsymbol{y}^*) \right) \right) \right|} \ . \tag{142}$$

The goal is to proof that the BE

$$B_{\boldsymbol{x}}^{\text{err}} = \int d\boldsymbol{x} \, \min \left\{ p_{0,\boldsymbol{x}} \left(\boldsymbol{x} \right), p_{1,\boldsymbol{x}} \left(\boldsymbol{x} \right) \right\} \tag{143}$$

is always smaller or equal to the BE of the transformed variable

$$B_{\boldsymbol{y}}^{\text{err}} = \int d\boldsymbol{y} \, \min \left\{ p_{0,\boldsymbol{y}} \left(\boldsymbol{y} \right), p_{1,\boldsymbol{y}} \left(\boldsymbol{y} \right) \right\} \ , \tag{144}$$

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

where the indices 0 and 1 denote that the PDFs are assigned to two distinct hypotheses. Inserting Eq. (142) into Eq. (144) yields

$$B_{\boldsymbol{y}}^{\text{err}} = \int d\boldsymbol{y} \min\left\{ \int d\boldsymbol{v} \frac{p_{0,\boldsymbol{y}}\left(f^{*-1}(\boldsymbol{y}^*)\right.}{|\det\left((\boldsymbol{D}f^*)\left(f^{*-1}(\boldsymbol{y}^*)\right)\right)|}, \int d\boldsymbol{v} \frac{p_{1,\boldsymbol{y}}\left(f^{*-1}(\boldsymbol{y}^*)\right.}{|\det\left((\boldsymbol{D}f^*)\left(f^{*-1}(\boldsymbol{y}^*)\right)\right)|} \right\}$$
$$\geq \int d\boldsymbol{y} d\boldsymbol{v} \min\left\{ \frac{p_{0,\boldsymbol{y}}\left(f^{*-1}(\boldsymbol{y}^*)\right.}{|\det\left((\boldsymbol{D}f^*)\left(f^{*-1}(\boldsymbol{y}^*)\right)\right)|}, \frac{p_{1,\boldsymbol{y}}\left(f^{*-1}(\boldsymbol{y}^*)\right.}{|\det\left((\boldsymbol{D}f^*)\left(f^{*-1}(\boldsymbol{y}^*)\right)\right)|} \right\} . \quad (145)$$

The greater or equal to sign in the equation above can be understood in the following way: When the summation (the integral) is performed first and the minimum is taken afterwards, the result is necessarily greater or equal to the sum of the minimum values.

Since invertible transformations do not change the BE, we can write

$$B_{\boldsymbol{x}}^{\text{err}} = \int d\boldsymbol{y}^* \min\left\{ \frac{p_{0,\boldsymbol{y}}\left(f^{*-1}(\boldsymbol{y}^*)\right.}{|\det\left((\boldsymbol{D}f^*)\left(f^{*-1}(\boldsymbol{y}^*)\right)\right)|}, \frac{p_{1,\boldsymbol{y}}\left(f^{*-1}(\boldsymbol{y}^*)\right.}{|\det\left((\boldsymbol{D}f^*)\left(f^{*-1}(\boldsymbol{y}^*)\right)\right)|} \right\} , \quad (146)$$

with $d\boldsymbol{y}^* = d\boldsymbol{y}d\boldsymbol{v}$. Comparing these expressions immediately yields $B_{\boldsymbol{y}}^{\text{err}} \geq B_{\boldsymbol{x}}^{\text{err}}$.

E Specifications of the Neural Networks

The hyperparameters and further specifications of the ANNs that are used within the scope of this thesis are listed in this section. Giving a detailed explanation everytime would make the main text difficult to read.

E.1 Generation of the Datasets used for Training, Validation and Testing

Whenever in this thesis, an ANN is trained to perform an estimation task, the training data set is generated in the following way:
The task of the ANN is to estimate the parameter θ from data drawn from a distribution $P(\theta)$. A data set is generated that consists of multiple output values θ_k, $k \in \{1, \ldots, K\}$ and a sample $\boldsymbol{x}_i \sim P(\theta_k)$, $i \in \{1, \ldots, N\}$. For simplicity, the grid of θ-values is equally spaced, i.e., $\theta_{k+1} - \theta_k = (\theta_K - \theta_1)/(K-1)$. The training data set then consists of N random vectors for each of the K parameter values, thus $N \times K$ random vectors in total.
Validation data sets are generated in the same way.
In principle, testing data sets are also generated analogously, although the grid spacing is chosen differently. We use the test data set to calculate the FI, thus at every point θ_k where the FI is evaluated, θ_{k+1} and θ_{k-1} must be sufficiently close together, such that the derivative w.r.t. θ can be evaluated accurately. This can be achieved by either an unequally spaced grid, where each θ_k is near two other grid points, or with an equally spaced grid that is very fine, i.e., K is very large and thus $(\theta_K - \theta_1)/(K-1) \ll (\theta_K - \theta_1)$.

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

For the binary classification tasks used within the scope of this thesis, the structure of the data sets is much simpler, as the number of grid points reduces to $K = 2$.

E.2 Neural Network Efficiency

Training Dataset	
$p(\boldsymbol{x}\|\theta)$	$\prod_{i=1}^{50} \frac{1}{\sqrt{2\pi\sigma^2}} e^{-(x_i-\theta)^2/2\sigma^2}$, $\sigma = 1$
θ_1	-10
θ_K	10
K	200
N	1000
Network	
Nodes	$50 \to 20 \to 10 \to 5 \to 1$
Hidden layer activation function	ReLU
Output layer activation function	Linear
Loss function	MSE
Batch size	30
Epochs	10

Table 1: Training data set and the network's specifics for the ANN with Gaussian input, used to evaluate the efficiency.

Training Dataset	
$p(\boldsymbol{x}\|\theta)$ (unperturbed)	$\prod_{i=1}^{50} \frac{1}{\sqrt{2\pi\sigma^2}} e^{-(x_i-\theta)^2/2\sigma^2}$, $\sigma = 1$
$p(\delta)$ (noise)	$\prod_{i=1}^{50} \frac{1}{2\alpha} e^{-\|\delta_i\|/\alpha}$, $\alpha = 0.18$
θ_1	-10
θ_K	10
K	200
N	1000
Network	
Nodes	$50 \to 20 \to 10 \to 5 \to 1$
Hidden layer activation function	ReLU
Output layer activation function	Linear
Loss function	MSE
Batch size	30
Epochs	10

Table 2: Training data set and the network's specifics for the ANN with Gaussian input and a Laplacian perturbation, used to evaluate the efficiency.

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

Training Dataset	
$p(\boldsymbol{x}\|\theta)$	$\left(e^{-(x_1-(\theta-\delta))^2/(2\sigma^2)} + e^{-(x_1-(\theta+\delta))^2/(2\sigma^2)}\right)/(\sqrt{8\pi}\sigma)$ $\times \prod_{i=2}^{50} \frac{1}{\sqrt{2\pi\sigma^2}} e^{-(x_i-\theta)^2/2\sigma^2}$, $\sigma = 1$, $\delta = 5$
θ_1	-10
θ_K	10
K	200
N	1000
Network	
Nodes	$50 \to 20 \to 10 \to 5 \to 1$
Hidden layer activation function	ReLU
Output layer activation function	Linear
Loss function	MSE
Batch size	30
Epochs	10

Table 3: Training data set and the network's specifics for the ANN. The input distribution has two peaks, one at each of the positions $x_1 \pm \delta, x_{i>1} = 0$.

E.3 Maximising the LFI of the Training Data

Training Dataset	
$p(\boldsymbol{x}\|\theta)$	$\prod_{i=1}^{20} \frac{1}{\sqrt{2\pi\sigma^2}} e^{-(x_i-\mu)^2/2\theta_i^2}$, $\mu = 0$
θ_1	0.2
θ_K	1.4
K	200
N	1000
Network	
Nodes	$20 \to 20 \to 10 \to 5 \to 2 \to 1$
Hidden layer activation function	ReLU
Output layer activation function	Linear
Loss function	MSE
Batch size	100
Epochs	5

Table 4: Training data set and the network's specifics for the ANN with Gaussian input. The parameter of interest is now the standard deviation, instead of the mean. Note that the LFI of the data set with respect to the parameter vanishes.

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

E.4 Early Stopping

Training Dataset	
$p(\boldsymbol{x}\|\theta)$ (unperturbed)	$\prod_{i=1}^{50} \frac{1}{\sqrt{2\pi\sigma^2}} e^{-(x_i-\theta)^2/2\sigma^2}$, $\sigma = 1$
$p(\delta)$ (noise)	$\prod_{i=1}^{50} \frac{1}{2\alpha} e^{-\|\delta_i\|/\alpha}$, $\alpha = 0.18$
θ_1	-10
θ_K	10
K	100
N	1000
Network	
Nodes	$20 \to 50 \to 100 \to 200 \to 100 \to 50 \to 1$
Hidden layer activation function	ReLU
Output layer activation function	Linear
Loss function	MSE
Batch size	5000
Epochs	300

Table 5: Training data set and the network's specifics for the evaluation of the loss functions's lower bound. The number of nodes is large to ensure that the network is overfitting when it is trained for a sufficiently long time.

E.5 Shallow Neural Networks

Training Dataset	
$p(\boldsymbol{x}\|\theta)$	$\prod_{i=1}^{20} \frac{1}{\sqrt{2\pi\sigma^2}} e^{-(x_i-\theta)^2/2\sigma^2}$, $\sigma = 1$
θ_1	-10
θ_K	10
K	100
N	1000
Network	
Nodes	$50 \to 25 \to 25 \to 1$
Hidden layer activation function	LeakyReLU
Output layer activation function	Linear
Loss function	MSE
Batch size	30
Epochs	20

Table 6: Training data set and the network's specifics for a shallow ANN with Gaussian input.

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

E.6 Linear Fisher Information Approach

Training Dataset	
$p(\boldsymbol{x}\|\theta)$	$\prod_{i=1}^{50} \frac{1}{\sqrt{2\pi\sigma^2}} e^{-(x_i-\theta)^2/2\sigma^2}$, $\sigma = 1$
θ_1	-10
θ_K	10
K	200
N	1000
Network	
Nodes	$50 \to 30 \to 20 \to 10 \to 5 \to 1$
Hidden layer activation function	ReLU
Output layer activation function	Linear
Loss function	MSE
Batch size	30
Epochs	15

Table 7: Training data set and the network's specifics for the ANN with Gaussian input, used to evaluate the the FI in each layer.

Training Dataset	
$p(\boldsymbol{x}\|\theta)$	$\left(e^{-(x_1-(\theta-\delta))^2/(2\sigma^2)} + e^{-(x_1-(\theta+\delta))^2/(2\sigma^2)}\right)/(\sqrt{8\pi}\sigma)$ $\times \prod_{i=2}^{50} \frac{1}{\sqrt{2\pi\sigma^2}} e^{-(x_i-\theta)^2/2\sigma^2}$, $\sigma = 1$, $\delta = 5$
θ_1	-10
θ_K	10
K	200
N	1000
Network	
Nodes	$20 \to 15 \to 10 \to 5 \to 1$
Hidden layer activation function	ReLU
Output layer activation function	Linear
Loss function	MSE
Batch size	30
Epochs	20

Table 8: Training data set and the network's specifics for the ANN with bimodal input, used to evaluate the the FI in each layer.

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

E.7 Fisher Information during Training

Training Dataset	
$p(\boldsymbol{x}\|\theta)$	$\prod_{i=1}^{100} \frac{1}{\sqrt{2\pi\sigma^2}} e^{-(x_i-\theta)^2/2\sigma^2}$, $\sigma = 0.2$
θ_1	-10
θ_K	10
K	200
N	1000
Network	
Nodes	$100 \to 70 \to 50 \to 30 \to 15 \to 1$
Hidden layer activation function	LeakyReLU
Output layer activation function	Linear
Loss function	MSE
Batch size	30
Epochs	5

Table 9: Training data set and the network's specifics for the ANN with Gaussian input, used to evaluate the the FI in each layer during each epoch of training.

E.8 Error Detection with Fisher Information

Training Dataset	
$p(\boldsymbol{x}\|\theta)$	$\prod_{i=1}^{30} \frac{1}{\sqrt{2\pi\sigma^2}} e^{-(x_i-\theta)^2/2\sigma^2}$, $\sigma = 0.2$
θ_1	-10
θ_K	10
K	200
N	1000
Network	
Nodes	$30 \to 20 \to 10 \to 5 \to 3 \to 1$
Hidden layer activation function	LeakyReLU
Output layer activation function	Linear
Loss function	MSE
Batch size	30
Epochs	5

Table 10: Training data set and the network's specifics for the ANN with Gaussian input, used for error detection.

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

E.9 Source of Error

Training Dataset	
$p(\boldsymbol{x}\|\theta)$	$\prod_{i=1}^{30} \frac{1}{\sqrt{2\pi\sigma^2}} e^{-(x_i-\theta)^2/2\sigma^2}$, $\sigma = 1$
θ_1	-10
θ_K	10
K	10
N	1000
Network	
Nodes	$20 \to 15 \to 10 \to 5 \to 1$
Hidden layer activation function	ReLU
Output layer activation function	Linear
Loss function	MSE
Batch size	10
Epochs	30

Table 11: Training data set and the network's specifics for the ANN with Gaussian input, used for demonstrating the sources of error when evaluating the FI with a finite difference method.

E.10 Monte Carlo

Training Dataset	
$p(\boldsymbol{x}\|\theta)$	$\prod_{i=1}^{10} \frac{1}{\sqrt{2\pi\sigma^2}} e^{-(x_i-\theta)^2/2\sigma^2}$, $\sigma = 0.1$
θ_1	-10
θ_K	10
K	200
N	10000
Network	
Nodes	$10 \to 7 \to 5 \to 3 \to 1$
Hidden layer activation function	LeakyReLU
Output layer activation function	Linear
Loss function	MSE
Batch size	30
Epochs	10

Table 12: Training data set and the network's specifics for the ANN with Gaussian input, used for demonstrating the Markov chain Monte Carlo method.

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

E.11 Bayes Error Propagation

Training Dataset	
$p_0(\boldsymbol{x})$	$\prod_{i=1}^{30} \frac{1}{\sqrt{2\pi\sigma^2}} e^{-(x_i-1)^2/2\sigma^2}$, $\sigma = 4$
$p_1(\boldsymbol{x})$	$\prod_{i=1}^{30} \frac{1}{\sqrt{2\pi\sigma^2}} e^{-(x_i+1)^2/2\sigma^2}$, $\sigma = 4$
N	10000
Network	
Nodes	$30 \to 20 \to 15 \to 10 \to 5 \to 2 \to 1$
Hidden layer activation function	LeakyReLU
Output layer activation function	Softmax
Loss function	Binary Cross Entropy
Batch size	30
Epochs	10

Table 13: Training data set and the network's specifics for an ANN trained to perform a binary classification task with Gaussian input distributions. The goal is to compute the BE in each layer. Note that instead of the MSE, the binary cross entropy loss function is used.

E.12 Bayes Error During the Training

Training Dataset	
$p_0(\boldsymbol{x})$	$\prod_{i=1}^{30} \frac{1}{\sqrt{2\pi\sigma^2}} e^{-(x_i-1)^2/2\sigma^2}$, $\sigma = 4$
$p_1(\boldsymbol{x})$	$\prod_{i=1}^{30} \frac{1}{\sqrt{2\pi\sigma^2}} e^{-(x_i+1)^2/2\sigma^2}$, $\sigma = 4$
N	10000
Network	
Nodes	$30 \to 20 \to 15 \to 10 \to 5 \to 2 \to 1$
Hidden layer activation function	LeakyReLU
Output layer activation function	Softmax
Loss function	Binary Cross Entropy (binary cross entropy)
Batch size	500
Epochs	4

Table 14: Training data set and the network's specifics for an ANN trained to perform a binary classification task with Gaussian input distributions, with the aim of tracking the BE during the training.

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

E.13 Error Detection with Bayes Error

Training Dataset	
$p_0(\boldsymbol{x})$	$\prod_{i=1}^{30} \frac{1}{\sqrt{2\pi\sigma^2}} e^{-(x_i-1)^2/2\sigma^2}$, $\sigma = 4$
$p_1(\boldsymbol{x})$	$\prod_{i=1}^{30} \frac{1}{\sqrt{2\pi\sigma^2}} e^{-(x_i+1)^2/2\sigma^2}$, $\sigma = 4$
N	10000
Network	
Nodes	$30 \to 20 \to 15 \to 10 \to 5 \to 2 \to 1$
Hidden layer activation function	LeakyReLU
Output layer activation function	Softmax
Loss function	Binary cross entropy
Batch size	30
Epochs	10

Table 15: Training data set and the network's specifics an ANN trained to perform a binary classification task with Gaussian input distributions, with the aim of tracking the BE during the training.

References

[1] Y. Bengio, I. Goodfellow, and A. Courville, *Deep learning*, vol. 1. MIT press Cambridge, MA, USA, 2017.

[2] F. Chollet, *Deep learning with Python*. Simon and Schuster, 2021.

[3] Y. Li, Y. Xue, and L. Tian, "Deep speckle correlation: a deep learning approach toward scalable imaging through scattering media," *Optica*, vol. 5, no. 10, pp. 1181–1190, 2018.

[4] N. Tishby and N. Zaslavsky, "Deep learning and the information bottleneck principle," in *2015 ieee information theory workshop (itw)*, pp. 1–5, IEEE, 2015.

[5] R. Shwartz-Ziv and N. Tishby, "Opening the black box of deep neural networks via information," *arXiv preprint arXiv:1703.00810*, 2017.

[6] A. Kolchinsky, B. D. Tracey, and S. Van Kuyk, "Caveats for information bottleneck in deterministic scenarios," *arXiv preprint arXiv:1808.07593*, 2018.

[7] B. C. Geiger, "On information plane analyses of neural network classifiers–a review," *IEEE Transactions on Neural Networks and Learning Systems*, 2021.

[8] A. Van den Bos, *Parameter estimation for scientists and engineers*. John Wiley & Sons, 2007.

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

[9] P. Zegers, "Fisher information properties," *Entropy*, vol. 17, no. 7, pp. 4918–4939, 2015.

[10] R. A. Fisher, "On the mathematical foundations of theoretical statistics," *Philosophical transactions of the Royal Society of London. Series A, containing papers of a mathematical or physical character*, vol. 222, no. 594-604, pp. 309–368, 1922.

[11] R. Zamir, "A proof of the fisher information inequality via a data processing argument," *IEEE Transactions on Information Theory*, vol. 44, no. 3, pp. 1246–1250, 1998.

[12] H. L. Van Trees, *Detection, estimation, and modulation theory, part I: detection, estimation, and linear modulation theory*. John Wiley & Sons, 2004.

[13] D. Wackerly, W. Mendenhall, and R. L. Scheaffer, *Mathematical statistics with applications*. Cengage Learning, 2014.

[14] L. Malagò and G. Pistone, "Information geometry of the gaussian distribution in view of stochastic optimization," in *Proceedings of the 2015 ACM Conference on Foundations of Genetic Algorithms XIII*, pp. 150–162, 2015.

[15] D. Pollard, "Insuffciency and the preservation of fisher information," in *A plenary lecture given at the New England Statistics symposium*, 2012.

[16] M. H. DeGroot and M. J. Schervish, *Probability and statistics*. Pearson Education, 2012.

[17] S. J. Prince, *Computer vision: models, learning, and inference*. Cambridge University Press, 2012.

[18] D. A. Freedman, *Statistical models: theory and practice*. cambridge university press, 2009.

[19] J. L. Devore, *Probability and Statistics for Engineering and the Sciences*. Cengage learning, 2011.

[20] E. Parzen, "On estimation of a probability density function and mode," *The annals of mathematical statistics*, vol. 33, no. 3, pp. 1065–1076, 1962.

[21] B. Park, B. Turlach, *et al.*, "Practical performance of several data driven bandwidth selectors," tech. rep., Université catholique de Louvain, Center for Operations Research and . . . , 1992.

[22] I. Kobyzev, S. J. Prince, and M. A. Brubaker, "Normalizing flows: An introduction and review of current methods," *IEEE transactions on pattern analysis and machine intelligence*, vol. 43, no. 11, pp. 3964–3979, 2020.

[23] M. Cvitkovic and G. Koliander, "Minimal achievable sufficient statistic learning," in *International Conference on Machine Learning*, pp. 1465–1474, PMLR, 2019.

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

[24] G. Cybenko, "Approximation by superpositions of a sigmoidal function," *Mathematics of control, signals and systems*, vol. 2, no. 4, pp. 303–314, 1989.

[25] T. Eltoft, T. Kim, and T.-W. Lee, "On the multivariate laplace distribution," *IEEE Signal Processing Letters*, vol. 13, no. 5, pp. 300–303, 2006.

[26] E. L. Lehmann and G. Casella, *Theory of point estimation*. Springer Science & Business Media, 2006.

[27] R. Bro and A. K. Smilde, "Principal component analysis," *Analytical methods*, vol. 6, no. 9, pp. 2812–2831, 2014.

[28] C. Andrieu, N. De Freitas, A. Doucet, and M. I. Jordan, "An introduction to mcmc for machine learning," *Machine learning*, vol. 50, no. 1, pp. 5–43, 2003.

Acronyms

ANN Artificial Neural Network

BE Bayes Error

CRI Cramér-Rao Inequality

CRLB Cramér-Rao Lower Bound

DL Deep Learning

DNN Deep Neural Network

FI Fisher Information

FII Fisher Information Inequality

FIM Fisher Information Matrix

LFI Linear Fisher Information

ML Machine Learning

MLE Maximum Likelihood Estimation

MSE Mean Squared Error

MVUE Minimum Variance Unbiased Estimator

NNE Neural Network Estimator

PCA Principal Component Analysis

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

PDF Probability Density Function

RV Random Variable

SVD Singular Value Decomposition

VTI Van Trees Inequality

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU WIEN Bibliothek Your knowledge hub