Department of Financial Mathematics

Compare the quality of forecasting models for value at risk

Author:
Hafees Adebayo Yusuff

Supervisor:
Prof. Ralf Korn

July 1, 2021

This thesis is written in collaboration with SociovestixLabs as a requirement for the completion of my degree of Master of Science at Technical University of Kaiserslautern.

Contents

Chapter 1

Introduction

1.1 Motivation

Basel I (Basel Accord) is the agreement reached in 1988 in Basel (Switzerland) by the Basel Committee on Bank Supervision (BCBS), involving the chairmen of the central banks of some European countries and the United States of America. This accord provides recommendations on banking regulations with regard to credit, market and operational risks. It aims to ensure that financial institutions hold enough capital on account to meet obligations and absorb unexpected losses.

For a financial institution measuring the risk it faces is an essential task. In the specific case of market risk, a possible method of measurement is the evaluation of losses likely to be incurred when the price of the portfolio assets falls. This is what Value at Risk (VaR) does.

Value at Risk(VaR) is the most common way of measuring market risk. It determines the greatest possible loss, assuming an α significance level under a normal market condition at a set time period.

Many VaR estimation methods have been developed in order to reduced uncertainty. It is however of interest to compare these method and determine the prevalence of one VaR estimation approach over others.

1.2 Literature review

The first papers involving the comparison of VaR methodologies, such as those by Beder (1995, 1996), Hendricks (1996), and Pritsker (1997), reported that the Historical Simulation performed at least as well as the methodologies developed in the early years, the Parametric approach and the Monte Carlo simulation. These papers conclude that among earlier methods, no approach appeared to perform better than the others. The evaluation and categorization of models carried out in the work by McAleer, Jimenez-Martin and Perez-Amaral(2009) and Shams and Sina (2014), among others, try to determine the conditions under which certain models predict the best. Researchers compared models in periods of varying volatility-before the crisis and after the crisis (When there was no high volatility and when volatility was high, respectively). However, this confirms that some models have good predictions before the start of the crisis, but their quality reduces with increased volatility. Others are more conservative during periods of low volatility, but in the time of the crisis the number of errors made by these models is relatively low.

Bao et al.(2006), Consigli(2002) and Danielson(2002), among others, show that in stable periods, parametric models provide satisfactory results that become less satisfactory during high volatility periods. Additional studies that find evidence in favour of parametric methods are Sarma et al.(2003), who compare Historical simulation and Parametric methods, and

Danielson and Vries(2000) in a similar comparison that also includes Extreme value theory methods. Chong(2004), who uses parametric methods to estimate VaR under a Normal distribution and under a Student's t-distribution, finds a better performance under Normality. McAleer et al.(2009) showed that RiskMetrics^{TM} was the best fitted model during a crisis, while Shams and Sina(2014) recognized GARCH(1,1) and GJR-GARCH as well forecasting models. In contrast to the results obtained by McAleer et al.(2009), the level of quality of forecasts generated by the RiskMetrics^{TM} model was considered unsatisfactory by them. However, attention needs to be drawn to one difference in the samples, on which the study was conducted, i.e. the first one comes from a developed country (USA, S&P500), and the second one from a developing country (Iran, TSEM). Taylor(2020) evaluate Value at Risk using quantile skill score and the conditional autoregressive model outperformed others.

Attempts have been made to predicts VaR with ANN. VaR estimation on the exchange rate market in the context of ANNs is dealt with in Locarek-Junge and Prinzler (1999), who illustrate how VaR estimates can be obtained by using a USD-portfolio. The empirical outcomes demonstrate an evident superiority of the neural network to other VaR models. Hamid and Iqbal(2004) compared volatility forecasts from neural networks with forecasts of implied volatility from S&P500 index futures options, using the Barone-Adesi and Whaley (BAW) American futures options pricing model. Forecasts from NN outperformed implied volatility forecasts. Similar results are put forth by He et al. (2018), who propose an innovative EMD-DBN type of ANN to estimate VaR on the USD against the AUD, CAD, CHF and the EUR. The authors find positive performance improvement in the risk estimates, and argue that the utilization of an EMD-DBN network can identify more optimal ensemble weights and is less sensitive to noise disruption compared to a FNN. Nevertheless, it is worthwhile to mention that although foreign exchange volatility forecasting through ANNs have gained some attention in the academic field, it still remains a fairly undeveloped area.

All in all, there is no full approval in the evaluation of which models should be used during periods of calm (low volatility), and which ones during crisis (High volatility).

1.3 Thesis Structure

The next chapter of discusses the properties and basic methods to estimate VaR. Subsequent chapters discuss use of Neural Network in Estimating Value at Risk and numerical comparison of the methods with examples. Findings are summarized in the last chapter.

Chapter 2

Value-at-Risk: Concept, properties and methods

2.1 Concept

Higher volatility in exchange markets, credit defaults, even endangering countries, and the call for more regulation drastically changed the circumstances in which banks operate. These situations of uncertainty are called risks and managing them is of great importance to financial institutions (e.g Banks) in order to keep them afloat. A possible method of measurement is the evaluation of losses likely to be incurred when the price of the portfolio falls. Value at Risk (VaR) does this.

According to Jorion (2001), "VaR measure is defined as the worst expected loss over a given horizon under normal market conditions at a given level of confidence. For instance, a bank might say that the daily VaR of its trading portfolio is \$2 million at the 99% confidence level. In other words, under normal market conditions, only 1% of the time, the daily loss will exceed \$2 million (99% of the time, their loss will not be more than \$2 million)". As represented in the mathematical representation below, it can also be stated as the least expected return of a portfolio at time t and at a certain level of significance, α.

Mathematically,

Let $r_1, r_2, ..., r_n$ be independently and identically distributed(iid) random variables representing financial log returns. Use $F(r)$ to denote the cumulative distribution function, $F(r) = Pr(r_t < r|\Omega_{t-1})$ conditional on the information set Ω_{t-1} available at time t-1. Assume that $\{r_t\}$ follows the stochastic process;

$$\begin{aligned} r_t &= \mu_t + \varepsilon_t \\ \varepsilon_t &= \sigma_t z_t \qquad z_i \sim N(0,1) \end{aligned} \tag{2.1}$$

where $\sigma_t^2 = E[z_t^2|\Omega_{t-1}]$ and z_t has a conditional distribution function $G(z)$, $G(z) = Pr(z_t < z|\Omega_{t-1})$. The VaR with a given probability α $\epsilon(0,1)$, denoted by VaR(α), is defined as the α quantile of the probability distribution of financial returns:
$F(\text{VaR}(\alpha)) = Pr(r_t < \text{VaR}(\alpha)) = \alpha$ or $\text{VaR}(\alpha) = \inf\{v|P(r_t \leq v) = \alpha\}$

One can estimate this quantile in two different ways: (1) inverting the distribution function of financial returns, F(r), and (2) inverting the distribution function of innovations, with regard to $G(z)$ the latter, it is also necessary to estimate σ_t^2.

$$\text{VaR}(\alpha) = F^{-1}(\alpha) = \mu + \sigma_t G^{-1}(\alpha) \tag{2.2}$$

Hence, a VaR model involves the specification of $F(r)$ or $G(r)$. There are several method for these estimations. Having explained the concept of Value at Risk, it is however necessary to state some of its properties or attributes.

2.2 Properties

Fix α $\epsilon(0,1)$, then the Value at Risk of a portfolio where the net payoff is modelled by X at a level α is given as:
$\text{VaR}_\alpha(X) = \inf \{x\epsilon\mathbb{R}|P(X \leq x) = \alpha\}$ has the following properties

- Monotonicity
  if $X \leq Y$ then $\text{VaR}_\alpha(X) \leq \text{VaR}_\alpha(Y)$
- Translation invariance
  $\text{VaR}_\alpha(X + c) = \text{VaR}_\alpha(X) + c$
- Positive Homogeneity
  $\text{VaR}_\alpha(cX) = c\text{VaR}_\alpha(X)$ if $c > 0$.
- VaR is not subadditive: The sum of the VaRs of individual portfolio (VaR(X)+VaR(Y)) can be lesser than the VaR of the combined portfolio (VaR(X+Y)). This is however not a desirable property of value at risk

2.3 Popular methods for estimating VaR

The estimation of these functions ($F(r)$ or $G(r)$) can be carried out using the following methods:

2.3.1 Historical simulation

The historical simulation involves using past data to predict future. First of all, we have to identify the market variables that will affect the portfolio. Then, the data will be collected on the movements in these market variables over a certain time period. This provides us the alternative scenarios for what can happen between today and tomorrow. For each scenario, we calculate the changes in the dollar value of portfolio between today and tomorrow. This defines a probability distribution for changes in the value of portfolio. For instance, VaR for a portfolio using 1-day time horizon with 99% confidence level for 500 days data is nothing but an estimation of the loss when we are at the fifth-worst daily change.

Basically, historical simulation is extremely different from other type of simulation in that estimation of a covariance matrix is avoided. Therefore, this approach has simplified the computations especially for the cases of complicated portfolio.

The core of this approach is the time series of the aggregate portfolio return. More importantly, this approach can account for fat tails and is not prone to the accuracy of the model due to being independent of model risk. As this method is very powerful and intuitive, it is then become the most widely used methods to compute VaR. However, Historical simulation requires data on all risk factors to be available over a reasonably long historical period in order to give a good representation of what might happen in the future. As it depends on history, if we run a Historical Simulations VaR in a bull market, VaR may be underestimated. Similarly, if we run a Historical Simulations VaR just after a crash, the falling returns which the portfolio has experienced recently may distort VaR.

2.3.2 GARCH Model

The Generalized Autoregressive Conditional Heteroskedasticity(GARCH) model, proposed by Bollerslev (1986) is a generalization of the ARCH process created by Engle (1982), in which the conditional variance is not only the function of lagged random errors, but also of lagged conditional variances. The standard GARCH model (p, q) can be written as:

$$\begin{aligned} r_t &= \mu_t + \varepsilon_t \\ \varepsilon_t &= \sigma_t \xi_t \end{aligned} \tag{2.3}$$

where r_t = rate of return of the asset in the period t,
μ_t =conditional mean

ε_t = random error in the period t, which equals to the product of conditional standard deviation σ_t and the standardized random error ξ_t in the period t ($\xi_t \sim$ iid(0,1))

In turn, the equation of conditional variance, in the GARCH(p,q) model can be written as:

$$\sigma_t^2 = \omega + \sum_{i=1}^{q} \alpha_i \varepsilon_{t-i}^2 + \sum_{i=1}^{p} \beta_i \sigma_{t-i}^2 \tag{2.4}$$

where σ =conditional variance in the period t,
ω = constant (ω¿0)
α_i = weight of the random squared error in the period $t-1$,
β_i = weight of the conditional variance in the period $t-1$,
ε_{t-i}^2= squared random error in the period $t-1$,
σ_{t-i}^2 =variance in the period $t-1$,
q = number of random error squares periods used in the functional form of conditional variance,
p = number of lagged conditional variances used in the functional form of conditional variance.

. When we use high frequency data in conjunction with GARCH models, these need to be modified to incorporate the financial market micro structure. For example, we need to incorporate heterogeneous characteristics that appear when there are many traders working in a financial market trading with different time horizons. The HARCH(n) model was introduced by Müller et al. (1997) to try to solve this problem.

2.3.3 HAR Method

High frequency data are those measured in small time intervals. This kind of data is important to study the micro structure of financial markets and also because their use is becoming feasible due to the increase of computational power and data storage. The HARCH(n) model was introduced by Müller et al. (1997) to estimate the VaR for this kind of data. In fact, this model incorporates heterogeneous characteristics of high frequency financial time series and it is given by

$$r_t = \sigma_t \varepsilon_t$$

$$\sigma^2 = c_0 + \sum_{j=1}^{n} c_j \left(\sum_{i=1}^{j} r_{t-i} \right)^2 \tag{2.5}$$

where $c_0 > 0, c_n > 0, c_j \geq 0 \; \forall j = 1, ..., n-1$ and ε_t are identically and independent distributed (i.i.d.) random variables with zero expectation and unit variance and the c_j are parameters estimated using least squares.

Intraday data have been found to be useful in estimating features of the distribution of daily returns. For example, the realized volatility has been used widely as a basis for forecasting the daily volatility. The heterogeneous autoregressive (HAR) model of the realized volatility is a simple and pragmatic approach, where a volatility forecast is constructed from the realized volatility over different time horizons (Corsi, 2009). However, intraday data can be expensive, and resources are required for pre-processing. Given the ready availability of the daily high and low prices, an alternative way of capturing the intraday volatility is to use the intraday range. Where Range_t is the difference between the highest and lowest log prices on day t, to predict tomorrow's range from past daily, weekly, monthly averages of Range_t, we set up the linear regression model;

$$\text{Range}_t = \beta_1 + \beta_2\text{Range}_{t-1} + \beta_3\text{Range}^w_{t-1} + \beta_4\text{Range}^m_{t-1} + \varepsilon_t$$

$$\text{Range}^w_{t-1} = \frac{1}{5}\sum_{i=1}^{5}\text{Range}_{t-i} \tag{2.6}$$

$$\text{Range}^m_{t-1} = \frac{1}{22}\sum_{i=1}^{22}\text{Range}_{t-i}$$

where Range_t is the difference between the highest and lowest log prices on day t; Range^w_{t-1} and Range^m_{t-1} are averages of Range_t over a week and month, respectively; ε_t is an i.i.d. error term with zero mean; and the β_i are parameters that are estimated using least squares. The conditional variance is then expressed as a linear function of the square of Range_t , where the intercept and the coefficient are estimated using maximum likelihoods based on a Student t distribution.

2.3.4 CaViaR Method

Engle and Manganelli (2004) propose a conditional autoregressive quantile specification (CAViaR) quantile estimation. Instead of modeling the whole distribution, the quantile is modelled directly. The empirical fact that volatilities of stock market returns cluster over time may be translated in statistical words by saying that their distribution is autocorrelated. Consequently, the VaR, which is a quantile, must behave in similar way. A natural way to formalize this characteristic is to use some type of autoregressive specification

let $\{y_t\}_{t=1}^T$ and θ be a vector of portfolio returns and the probability associated with VaR respectively. Let x_t be a vector of time t observable variables(return or any other observable variables), and β_θ be a p-vector of unknown parameters. Finally, let $f_t(\beta) \equiv f_t(x_{t-1}, \beta_\theta)$ denote the time t θ-quantile of the distribution of the portfolio returns formed at $t-1$, The θ subscript is however suppressed from β_θ for convenience in notation: The general specification of CaViar would be:

$$f_t(\beta) = \beta_0 + \sum_{i=1}^{q}\beta_i f_{t-i}(\beta) + \sum_{j=1}^{r}\beta_j l(x_{t-j}) \tag{2.7}$$

where $p = q + r + 1$ is the dimension of β and l is a function of a finite number of lagged values of observables. The autoregressive terms $\beta_i f_{t-i}(\beta)$, $i = 1, ..., q$, ensure that the quantile changes smoothly over time. The parameters of CaViaR are estimated by quantile regression. The role of $l(x_{t-j})$ is to connect $f_t(\beta)$ to observable variables that belong to the information set.

The asymmetric slope is a variant of CaViar model, which allows the response to positive and negative returns to be different. It is modelled as:

$f_t(\beta) = \beta_1 + \beta_2 f_{t-1}(\beta) + \beta_3(y_{t-1})^+ + \beta_4(y_{t-1})^-$. where, $(y_{t-1})^+ = \max(y_{t-1}, 0)$ and $(y_{t-1})^- = \min(y_{t-1}, 0)$

Chapter 3

Estimating VaR using Neural Networks

Neural networks, also known as artificial neural networks (ANNs) are computing systems vaguely inspired by the biological neural networks that constitute animal brains. Their name and structure are inspired by the human brain, mimicking the way that biological neurons signal to one another.

Artificial neural networks (ANNs) are comprised of a node layers, containing an input layer, one or more hidden layers, and an output layer. Each node, or artificial neuron, connects to another and has an associated weight and threshold. If the output of any individual node is above the specified threshold value, that node is activated, sending data to the next layer of the network. Otherwise, no data is passed along to the next layer of the network.

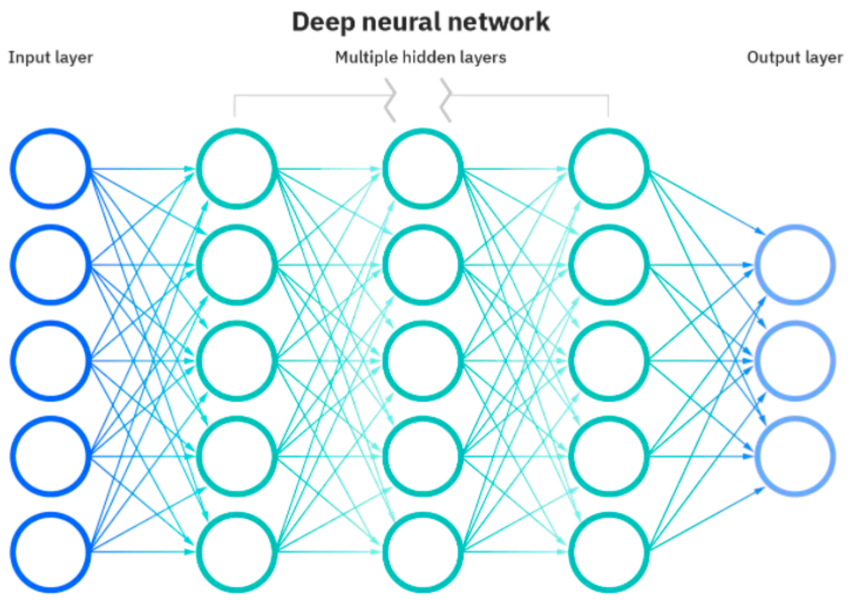

Figure 3.1: A figure showing the layers of a Neural Network

Neural networks rely on training data to learn and improve their accuracy over time. However, once these learning algorithms are fine-tuned for accuracy, they are powerful tools in computer science and artificial intelligence, allowing us to classify and cluster data at a high velocity. Tasks in speech recognition or image recognition can take minutes versus hours when compared to the manual identification by human experts.

3.1 Mathematics of Neural Network

3.1.1 A single Neuron

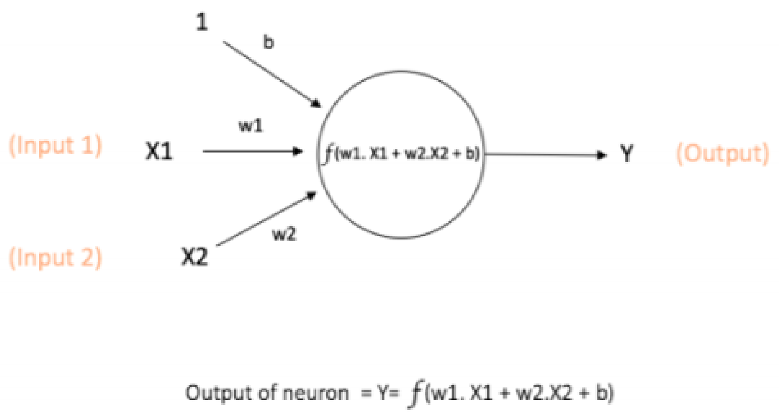

Figure 3.2: A single neuron of neural networks

Figure 3.2 shows a network with one layer containing a single neuron. This neuron receives input from the prior input layer, performs computations, and gives output. x_1 and x_2 are inputs with weights w_1 and w_2 respectively. The neuron applies a function f to the dot-product of these inputs, which is $w_1 x1 + w_2 x_2 + b$. Besides the two numerical input values, there is one input value 1 with weight b, called the Bias. The main function of bias is to stand for unknown parameters or unforeseen factors. The output Y is computed by taking the dot-product of all input values and their associated weights and putting it into the function f. This function is called the Activation Function.

Activation functions are needed because many problems take multiple influencing factors into account and yield classifications. For example, if one encounters a binary classification problem, the results would be either yes or no, activation functions are needed to map the results inside this range. If one encounters a problem involving probability, then one would wish to see the predictions from the neural network being in the range of [0, 1]. This is what activation functions can do.

There are two types of activation functions: linear activation functions and non-linear ones. The biggest limitation of linear ones is that they cannot learn complex function mappings because they are just polynomials of one degree. Therefore, we always need non-linear activation functions to produce results in desired ranges and to send them as inputs to the next layer. The following subsection will introduce few generally used non-linear activation functions.

3.1.2 Activation Functions

An activation function takes the dot-product mentioned before as a input and performs a certain computation on it. We put a certain activation function inside of neurons of hidden layers based on the range of the result we expect to see. A notable property of activation functions is that they should be differentiable, because later we need this property to train the neural network using backpropagation optimization.

Here are few commonly used activation functions:

Sigmoid: This takes a real-valued input and returns a output in the range [0,1]:

$\delta = \frac{1}{1+e^{-x}}$

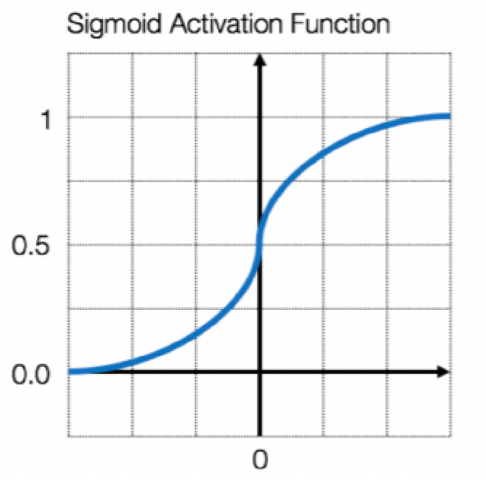

Figure 3.3: Sigmoid() Activation Function

Figure 3.3 shows an S-shaped curve and the values going through the Sigmoid function will be squeezed in the range of [0, 1]. Since the probability of anything exists only between the range of 0 and 1, Sigmoid is a compatible transfer function for probability. Although the Sigmoid function is easy to understand and ready to use, it is not frequently used because it has vanishing gradient problem. This problem is that, in some cases, the gradient gets so close to zero that it does not effectively apply change to the weight. In the worst case, this may completely stop the neural network from further training. Second, the output of this function is not zero-centered, which makes the gradient updates go far in different directions. Besides, the fact that output is in the narrow range [0, 1] makes optimization harder. In order to compensate the shortcomings, tanh() is an alternative option because it is a stretched version of the Sigmoid function, in which its output is zero-centered.

tanh: This takes real-valued input and produces the results in the range [-1, 1]:

tanh(x) $= \frac{sinh(x)}{cosh(x)} = \frac{e^x - e^{-x}}{e^x + e^{-x}}$

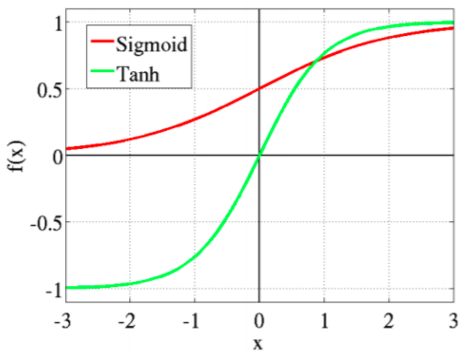

Figure 3.4: tanh() Activation Function

The advantage is that the negative input values will be mapped strongly negative and the zeros will be mapped near zero through this function. Therefore, this function is useful in the performance of a classification between two distinct classes. Though, this function is preferred over the Sigmoid function in practice (output in wider range), but the gradient vanishing problem still exists. The following ReLU function rectifies this problem using a relatively simple formula.

ReLU (Rectified Linear Unit): It takes a real-valued input and replaces the negative values with zero:

$R(x) = \max(0, x)$

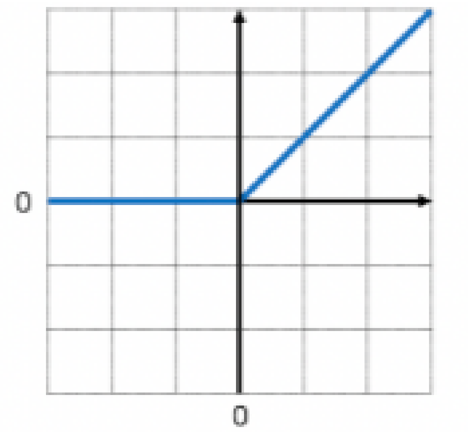

Figure 3.5: ReLU() Activation Function

It is used in almost all the convolutional neural networks or deep learning because it is a relatively simple and efficient function which avoids and rectifies the gradient vanishing problem. The problem of this activation function is that all the negative values become zeros after this activation, which in turns affects the results by not taking negative values into account.

We use different activation functions when we know what characteristics of results we expect to see.

3.1.3 General Model Building

A neural network learns from patterns of data and tries to make predictions as accurately as possible. Assume that we already have a set of p data pairs containing the variables and the results, $(x^{(1)}, t^{(1)})$, $(x^{(2)}, t^{(2)})$,...,$(x^{(p)}, t^{(p)})$ where $x^{(i)}$ is the input value and $t^{(i)}$ is the target value for i= 1,2,...,p. We would like to build a neural net F so that ideally,

$$F(x^{(i)}) = t^{(i)}$$

However, typically we allow for error ϵ_i. Let $y^{(i)}$ denote the output of the neural net so that

$$y^{(i)} = F(x^{(i)}) \text{ and } t^{(i)} = y^{(i)} + \overrightarrow{\epsilon}$$

We know that $y^{(i)}$ depends on parameters, which are weights and biases, then it turns out as an optimization problem. We need to set up a neural net F that minimizes the error function, which is denoted below:

$$E = \frac{1}{N} \sum_{i=1}^{p} ||t^{(i)} - y^{(i)}||^2$$

where N is the number of training patterns. If it is a two-way classification problem, then N = 2. From this equation, we know that E is a function of the parameters in F, and we need to determine the values of weights that minimize the error by differentiating E. If we focus on only one term of the sum, then

$$||t - y||^2 = (t_1 - y_1)^2 + (t_2 - y_2)^2 + ... + (t_p - y_p)^2$$

Chapter 4

Figures, tables, enumerate and itemize

4.1 Figures

In almost every document figures will be needed in order to explain a concept or just present something. The package *graphicx* is needed for embedding figures.

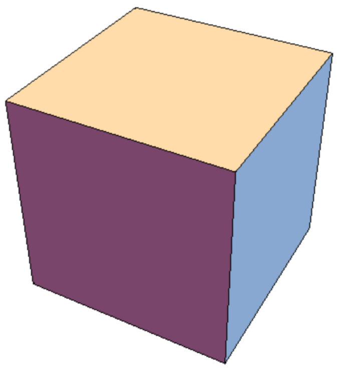

Figure 4.1: A figure caption beneath the figure for description of the depicted concept which sometimes can be very long

In Figure 4.1, for example, a PNG image is depicted (compiled with pdflatex). Alternatively, EPS figures can be embedded if dvips and ps2pdf compilation is used. All figures are listed in the list of figures with the command *listoffigures*.

4.2 Tables

Data can be presented in tables, e.g., as shown in Table 4.1.

	Property 1	Property 2
Criterion 1	764	23546
Criterion 2	3	34

Table 4.1: Exemplary table

Sometimes very long tables must be presented which may also go over pages. For this cases the packages *longtable* is useful, as used in

i^3	$2i^3$	$3i^3$
1	2	3
8	16	24
27	54	81
64	128	192
125	250	375
216	432	648
343	686	1029
512	1024	1536
729	1458	2187
1000	2000	3000
1331	2662	3993
1728	3456	5184
2197	4394	6591
2744	5488	8232
3375	6750	10125
4096	8192	12288
4913	9826	14739
5832	11664	17496
6859	13718	20577
8000	16000	24000
9261	18522	27783
10648	21296	31944
12167	24334	36501
13824	27648	41472
15625	31250	46875
17576	35152	52728
19683	39366	59049
21952	43904	65856
24389	48778	73167
27000	54000	81000
29791	59582	89373
32768	65536	98304
35937	71874	107811
39304	78608	117912
42875	85750	128625
46656	93312	139968
50653	101306	151959
54872	109744	164616
59319	118638	177957
64000	128000	192000
68921	137842	206763
74088	148176	222264
79507	159014	238521
85184	170368	255552
91125	182250	273375
97336	194672	292008
103823	207646	311469
110592	221184	331776
117649	235298	352947
125000	250000	375000

Table 4.2: Long Table

All tables are listed with *listoftables*.

4.3 Enumerate and itemize

If important sequential points are to presented the environment *enumerate* can be used as follows:

1. Some important stuff
2. More stuff

With the package *enumerate* some options can be used, e.g.,

a) Some important stuff

b) More stuff

or

1) Some important stuff
2) More stuff

Alternatively, point can be just presented without any enumeration with the environment *itemize*

- Some important stuff
- More stuff

Chapter 5

Appendix, footnotes, todos and index

5.1 Appendix

For many reasons some concept may be important for the document but too long for the main text. In this kind of cases these concept can be presented with the environment *appendix* in appendices, e.g., as in Appendix A and Appendix B.

5.2 Footnotes

You may want to give additional information to some points[1] in the text[2].

5.3 Todos

With the package *todonotes* comments pointing to their place can be embedded into the text. These comments are veeeery useful if you are writing something for the first time or are working on a draft. The todos can be listed with *listoftodos* where you want it to appear in order to see what is unfinished or needs some more work.

like this one

5.4 Index

If the document is very long, it may be very useful for a lot of readers to have an index for searching key words and certain concepts (Crtl+F is usually very helpful in PDFs but not always the best solution). For this, the package *makeidx*, the commands *makeindex* and *printindex* and the compiling option *make index* are needed. You may want to index different words like heterogeneous materials, effective properties and homogenization.

[1]Bla bla

[2]Blu blup

Appendix A

Just an example appendix

A.1 Bla blup

Sme stuff

$$f(x) = \int_{\Omega} g(x) dx \ . \tag{A.1}$$

Appendix B

Another example

B.1 More stuff

Bla bla.

Bibliography

R. Hill. The elastic behaviour of a crystalline aggregate. *Proceedings of the Physical Society. Section A*, 65:349–354, 1952.

E. Kröner. Bounds for effective elastic moduli of disordered materials. *Journal of the Mechanics and Physics of Solids*, 25(3):137–155, 1977.

List of Figures

List of Tables

Index