Some Thoughts on the Impact of Machine Learning on the Banking Industry – Beyond Myths

Authors: Farhad Khakzad, Dr. Sangmeng Li, Pablo Arboleda[1]

[1] Farhad Khakzad is the head of the "Analytical Regulatory and Risk" team at ifb group and Dr. Sangmeng Li is a member of his team. Pablo Arboleda graduated in statistics from Harvard College. This paper strictly reflects the opinion of the authors and not necessarily the view of the ifb group or any other institution. The authors reserve the right to use or publish parts of this article in other publications without the explicit permission of third parties. The authors would also like to express our thanks to all brains around the world who have supported us for developing this paper.

Content

ABSTRACT **3**

1. INTRODUCTION **5**

1.1. DEFINITION OF TERMS AND SELECTED KEY THOUGHTS ABOUT ARTIFICIAL INTELLIGENCE 5

1.2. AN OVERVIEW OF THE GERMAN BANKING INDUSTRY AS WELL AS SELECTED IMPORTANT ECONOMIC FRAMEWORK CONDITIONS 9

2. THE MATH BEHIND DEEP LEARNING **14**

2.1 GRADIENT DECENT ALGORITHM (GDA) 15

2.2 FEEDFORWARD NEURO NETWORK- ARCHITECTURE 20

2.3 FEEDFORWARD NEURO NETWORK – TRAINING PROCESS 23

2.4 FEEDFORWARD NEURO NETWORK – SOME IMPROVEMENT AND VARIATIONS OF TRAINING PROCESS 27

2.4.1 Momentum *27*

2.4.2 Nesterov Accelerated Gradient *28*

2.4.3 Adagrad *28*

2.4.4 Adadelta *29*

2.4.5 An excursion to Convolution Neural Network (CNN) *30*

2.5 DEEP LEARNING – THE TREND 33

3. EXCURSION TO THE TOPIC BLOCKCHAIN TECHNOLOGY AND THE INTERACTION TO MACHINE LEARNING **34**

4. CONCLUSION **36**

LIST OF REFERENCES **37**

Abstract

In 1890, a company was founded in the USA that eventually developed into a leading brand. 80 years after its inception, the company had a market share of approximately 90% in its core field of business: film and photography. The company's name is Kodak. Kodak was known for its top performance and the forward-looking view of its engineers. These engineers invented the digital camera in 1975. In spite of its long and successful company history, Kodak went bankrupt in 2012. The opportunities provided by the company's breakthrough technology and its engineering prowess may not have been fully leveraged by the management. Kodak may provide a useful lesson and an incentive for leaders to manage and leverage disruptive technologies.

From today's view, we believe that machine learning and deep learning are such disruptive technologies that should be leveraged. Advanced computing methods have become more and more popular alongside advances in the emergence of big data. In the finance and banking sector, increasing demand for more efficient approaches has been seen especially in the field of machine learning. The field was originally devoted to developing algorithms in artificial intelligence. In response to its significant advances, however, machine learning and its subfield of deep learning have emerged as breakthroughs with vast applications in a wide array of fields [1].

In our paper, we discuss the existing definitions as well as our thoughts on some of the applications of machine learning and deep learning. We will provide an insight into the significance of artificial intelligence in the banking industry, in particular and its response to big data market forces, with a focus on German Banks. To illustrate the breadth of the field's significance, we will also address the automotive industry in Germany and its connection to machine learning technologies. To deepen our understanding of the subject, we will explain the mathematical background behind deep learning and its intersection with artificial neural networks. We will provide a brief look into the subject of Blockchain and its relevance to artificial intelligence. A second part of this paper will analyze the practical implications (use cases) of the technologies and their expected developments in the short and long term.

Key Words

Artificial Intelligence, Machine Learning, Deep Learning, Data Quality, Automotive Industry, Banking Industry, Germany, Gradient Decent Algorithm, Artificial Neuro Network, Blockchain Technology

Artificial Intelligence is like teenage sex: everyone talks about it, nobody really knows how to do it, everyone thinks everyone else is doing it, so everyone claims they are doing it…*

** Based on the original quote by Dan Ariely from the Duke University*

1. Introduction

1.1. Definition of terms and selected key thoughts about artificial intelligence

There is no unequivocal definition of **artificial intelligence**. In fact, in the sciences, the concept of „intelligence“ has been widely discussed. Machine learning is a sub-field of artificial intelligence and deep learning is one of its “methodical” approaches. We believe **deep learning** is one of the most important future challenges of the century. We furthermore believe it to be a disruptive technology that will interact with our environment in a way that enables us to solve a broad array of sophisticated problems with higher speed and greater accuracy.

The raw material of our time consists mainly of data. Subsequently, there have been big changes in the whole industry over the last three decades, and new business models emerged from these changes. The social and communication behavior have been also affected by these changes. This can be seen for instance in the level of social coexistence of society, especially in the behavior of stakeholders.

The dependence on data driven decisions and the advancements of big data have urged businesses to create new products and to reconsider existing business models. A growing number of companies face the threat of massive changes, especially with regard to the disruptive business models of platform providers in the market.

According to our experience from the consulting industry and especially the assessment of the German banking industry, these difficulties are not due to a lack of understanding of the technologies. In fact, the banking industry lacks an overall understanding of how the market environment will change in cause of technological progress. In other words, some have not understood the nature and the extent of these changes yet. Unfortunately, this incomprehension leads to a digitalization aversion and culminates in a resulting lack of strategic planning at the highest management level.

Many companies are aware of the fact that the digital transformation will influence their business. However, most of them do not have an appropriate strategy to face the future challenges yet.

We strongly believe that the advances in artificial intelligence will accelerate this transformation process. Companies that for any reason do not face the challenges in the digital sector run the risk of dropping behind in the competition.

Especially the financial sector will be revolutionized by disruptive technologies, as evidenced by the following examples: Goldman Sachs no longer calls itself a "bank" but a "technology company". SocietyOne is one of the fastest-growing banks in the world [2]. The world's largest number of payment transactions are handled by a company that does not even have a bank branch and it has almost more customers than any German bank – PayPal.

In our opinion, however, these challenges are not limited to a strategic change of business models in terms of expertise and technology. In the corporate world, we need the same urge for change that has long since arrived in parts of the society. In other words, many companies must completely rethink their approaches and ~~a~~rtificial intelligence will be the basis of this rethinking process. Consulting firms could participate in this development with foresight, methodological excellence, practical and cross-company expertise.

Despite all the advantages of deep learning, the disadvantages should be also taken into account. Outstanding people of our time have already warned against some fears [3] [4] [5]. Regrettably, these discussions are increasingly ignoring the fact that the expansion of artificial intelligence into people's everyday lives raises new questions. The analogue and digital world is increasingly drifting apart. Our rule-based world collapsed in the usual way, for example, the existing laws such as intellectual property rights are being eroded in this way. Without the reflection on common values, it must be clear to everyone that artificial intelligence will not benefit all people in equal parts. Thus, the ethical questions with regards to the societal impact of deep learning will need to be discussed in the future. In order to substantiate these thoughts with concrete facts, the following three questions are concerned:

1. Why are not all people around the globe subject to one worldwide privacy protection?

2. Why is a film on an internet platform not subject to the same legal conditions as a film on television?

3. Why are identical life circumstances on online platforms subject to different legal conditions than offline?

These are supposedly simple but intellectually profound questions and ~~e~~ach of them is valid because they concern our basic values. Whoever takes the lead in artificial intelligence has a responsibility for humanity – this also applies to companies and their business relationship with their customers.

In our daily work practice, there is also pseudo-knowledge on these topics. One example is the assertion that predictive analytics is significantly less complex than deep learning. This is incorrect. While in predictive analytics success rates depend largely on expert knowledge and in particular, on the data scientist’s knowledge, deep learning simplifies the process because untransformed raw data are used and all assumptions about the distribution of data are taught more or less automatically. Overall, the solution of a problem with predictive analytics methods is considerably more complex than with deep learning. This presupposes that the initial problem is precisely understood by the users. Another example is the assertion that "artificial intelligence leads to robots taking jobs away from us". The truth is that our jobs are more likely to change completely in cause of further progress in the research area of artificial intelligence. Thus, our life becomes simpler and at the same time more volatile. However, that is logical because many questions in this context are currently unresolved. Furthermore, we have to differentiate very precisely between the short-term and long-term effects when we discuss about artificial intelligence. Finally, artificial intelligence is not a "machine that can be switched on and off".

First, we would like to introduce some details about our view of artificial intelligence. In the following Chapter 1.2 we will present some facts about the German banking market and the significance of artificial intelligence.

Algorithms are already part of our everyday life. For example, your phone uses speech recognition algorithms to make your everyday life easier; your mail web service uses algorithms to protect you from unwanted messages; your notebook uses algorithms to help you better manage your documents. These are just three examples where you can already benefit from algorithms and machine learning in your private life.

In the corporate world, for example in the banking industry, these algorithms will be generally used where there is an optimization of the business model. At present, the application concentrates on areas where costs can be saved. For example, in the insurance industry, Fukoku Mutual Life has decided to replace a significant number of employees with a machine learning system. In summary, the artificial intelligence system is intended to ensure the analysis and verification of relevant documents. The final decision will be done by human intelligence. This leads to a cost reduction for Fukoku Mutual Life of about 1.7 million euros per year so that the amortization of the project is already achieved after less than two years.

Unfortunately, there are some ambiguities in this context. Therefore, it is important to deal with the context of meaning in order to distinguish the different "methods" of the artificial intelligence from each other. For this purpose, the following points should be noted.

First, deep learning is similar to the implementation of an artificial neural network. It is an integral part of machine learning and at the same time the most common machine learning method in the corporate world. The global technology leaders each have their own specific technologies. The basis for this is an artificial neural network which tries to mimic the processes of the human brain. The learning process is driven by experience. For example, parents give the child feedback on whether it is a car or not. On this basis, the child learns to distinguish this object from the others over time. Now, if in the above example, we replace the parents with developers and the child with a developed software, we are already in deep practice of machine learning. Of course, this is a very simplified example. In fact, machines can learn human traits such as seeing, hearing, reading and speaking through deep learning and its methods. The most important thing here is to be clear about the target success rate with the method used. Deep learning always proceeds by training the mathematical models. If the result of the training matches the success rate, the job is done. The success rate has to be determined by the user.

Second, the distinction between deep learning and machine learning can be better explained as follows. In case of machine learning, additional information will be generated from the data analysis. The information is structured and linked with each other in order to identify causal relationships. From the recognition of this information, relevant and irrelevant information can be distinguished from each other. Based on the relevant information, relevant knowledge can be generated. On this basis, the quality and the speed of decision-making processes can be improved. Consequently, deep learning is always machine learning, but machine learning is not necessarily always deep learning.

In some papers, artificial intelligence has been wrongly interpreted as equal to machine learning. In fact, machine learning is only a sub-area of artificial intelligence; and artificial intelligence itself is a sub-area of computer science. Artificial intelligence is in fact an abbreviation for all automation processes of human behavioral characteristics.

Third, when we talk about machine learning, we should keep in mind what is essential and how it works in general. We already explained how deep learning works and found out that the basis of deep learning are artificial neural networks. This approach is equivalent to the learning process of the human brain. We also noted that deep learning is always machine learning. This background knowledge makes it easier to understand how machine learning works. In analogy to a human infant, the infant learns to recognize and distinguish objects.

However, we must always consider the methodological distinctions, in order to achieve useful results. For example, there are different methods and algorithms in deep learning to carry out pattern recognition procedures. All models have advantages and disadvantages, especially the success rates and complexity should be considered. Identifying the problem in concrete individual cases remains essential.

As we have already mentioned, machine learning is already a part of business practice: Spotify uses the technology to give its listeners recommendations for the next songs. Zalando uses the technology to show its customers recommendations for next winter season. Amazon uses this technology to provide its customers with product recommendations tailored to their needs. The only difference between the companies mentioned above is that Amazon also offers its own service area (machine learning as a service); a platform-based environment so that developers can build their own applications according to their needs (called "Amazon Machine Learning"). The world's largest technology leaders all have their own machine learning platforms. Amazon is undoubtedly one of them.

Finally, the topic of artificial intelligence becomes more exciting when we deal with the subject of economic success. Google receives more than 59,000 search queries every second or over 5 billion search queries a day. These search queries are basically data that will be transformed into usable information. All these searches will be conducted on the Google platform. To put it in the words of Twitter, we define a platform as a plug-and-play business model that allows multiple participants to connect to it, interact with each other to create and exchange value. Based on this platform, the company has developed an unprecedented product portfolio of more than 100 products. In this way, other companies could also earn money with machine learning. It is therefore true that the platform is the central element for all other Google products. This is an ingenious business model. Unfortunately, not everyone has grasped it yet.

1.2. An overview of the German banking industry as well as selected important economic framework conditions

The German banking market is easy to understand in terms of selected figures for the financial year 2016 [6][2].

The total industry sales amounted to € 25.7 billion with sales growth 1.6%[3] and a CAGR 1.8%[4]. € 13.5 billion of € 25.7 billion arises from lending business with private and business customers. Credit banks and branches of foreign banks are the largest sub-sector of the total industry sales (53.7%). More than 586,000 people are employed in 1,626 banks[5] with salaries on an average of about € 69.000 per year. The number of banks has declined from 2010 to 2016, with future falling tendency. Measured in terms of sales volume (> € 93 billion), Germany is only the fifth most important banking market in the world.

The *one and only* German bank with international significance is **Deutsche Bank** with a market capitalization amounting to about € 23.8 billion by the end of the financial year 2016. The market capitalization has more than halved over the past 10 years. The bank's equity capital amounted to around € 59.8 billion, tier 1 capital amounted to € 46.8 billion with earnings before taxes to around € - 0.81 billion, a negative equity on return to around - 2.3% and earnings per share amounted to € 0.11 [7][6]. The current figures for the first three quarters in 2017 give hope of improvement [7][7]. In conclusion, the **innovation intensity** of the financial services (banking) industry in Germany is **very low** in comparison to other sectors.

The banking industry is sitting on a treasure trove of data which could be subject to artificial intelligence approaches. Unfortunately, the potential has not yet been exploited or even recognized in the German banking sector. In fact, the banks know their customers better than many other industries, as they have insight in the consumer’s payment and saving behavior, travel and investment preferences to name just a few examples. With artificial intelligence, banks could create new business models with offering new services for their customers. Only few of them currently actually do this. In our view, the obstacles consist in both the special legal situation and the technological landscape of many banks in Germany. However, this unfavorable situation should be remedied promptly. In our opinion, artificial intelligence is likely to be a key driver for ensuring competitiveness of the German banking sector in the long term. The following graphic clearly illustrates the top 10 out of 30 worldwide banks.

[2] Page 7, 16, 17, 18 and 20.

[3] Estimated Growth from 2016 to 2017.

[4] Estimated Growth from 2016 to 2021.

[5] In 2015 (= a decrease of 1% compared to the previous year).

[6] Page 13, 16-20.

[7] Page 26-32.

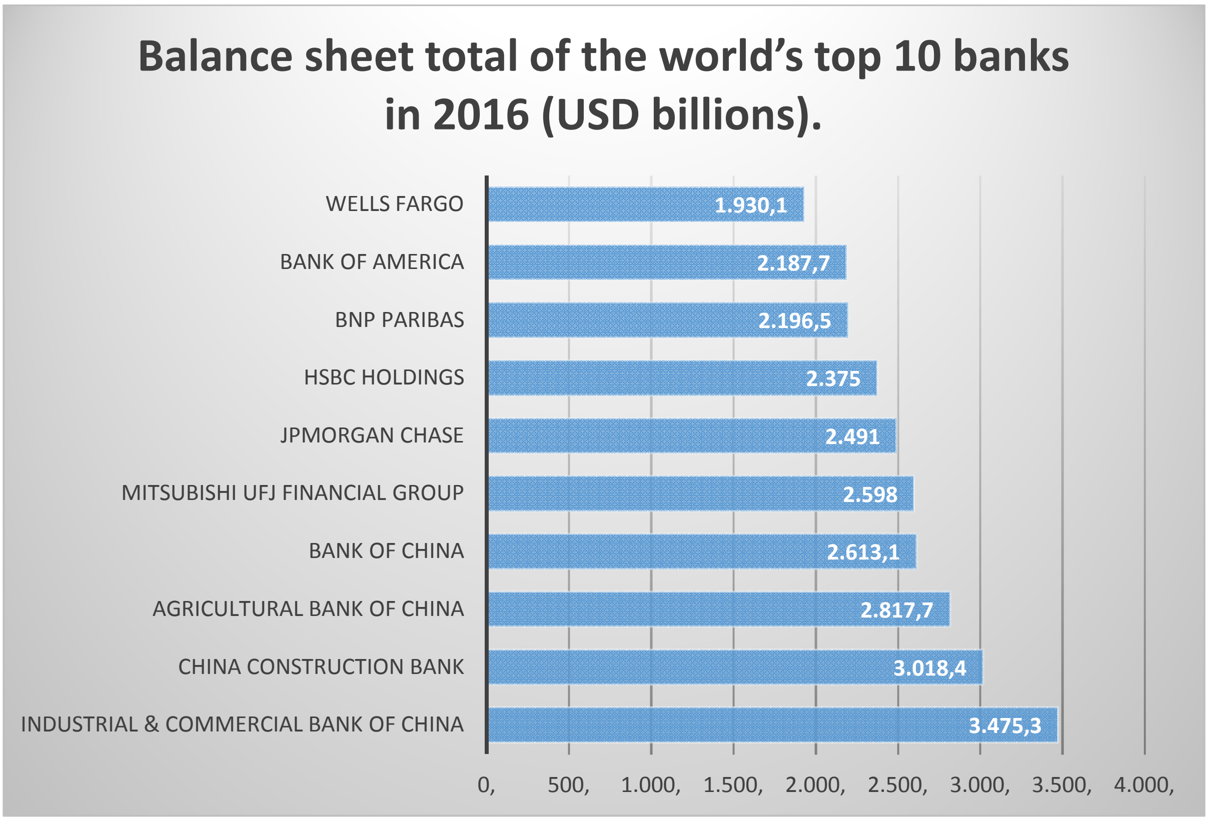

Figure 1-1

Among the top ten of the world’s largest credit institutions no German bank can be observed. Only Deutsche Bank (16th) ranks among the top 30 credit institutions by the end of 2016 [7][8].

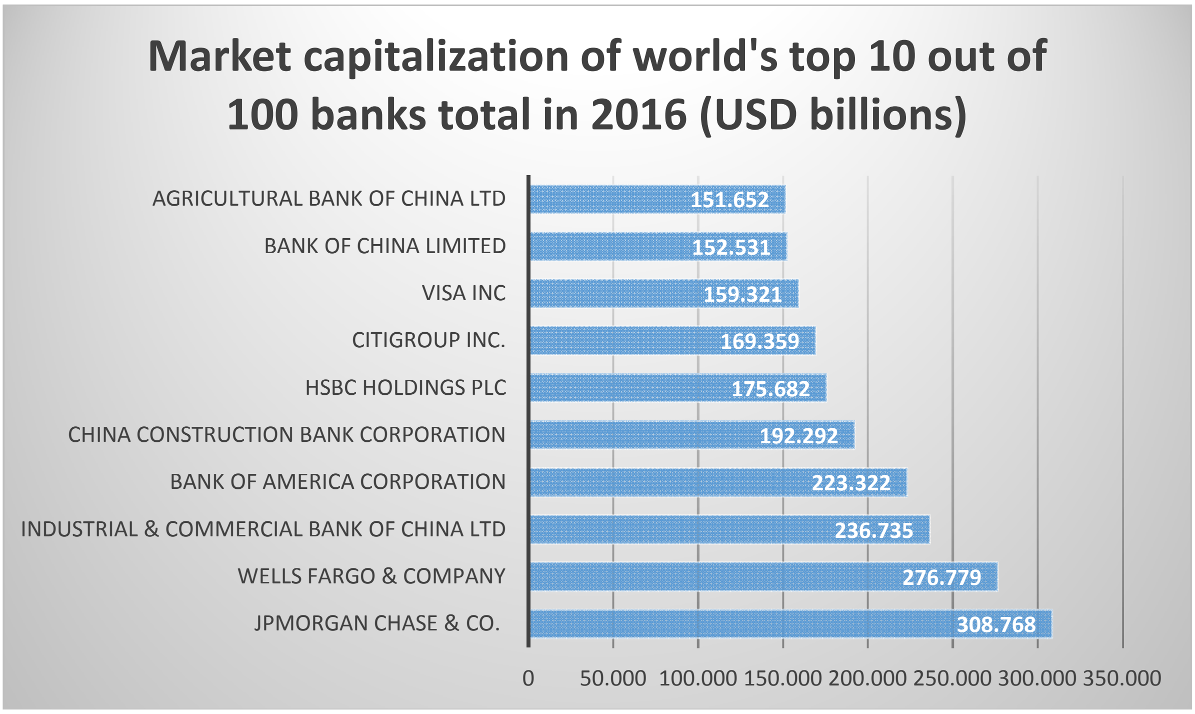

Figure 1-2

[8] Page 7.

The total market capitalization of the world's top ten credit institutions is displayed. In this view, Deutsche Bank covers the 67th rank among the top 100 credit institutions by the end of 2016; JP Morgan Chase & Co. is almost thirteen times “more valuable” than Deutsche Bank [8].

In addition to the above-mentioned facts, German banks' investments in the FinTech sector are also very restrained. While Citigroup, Banco Santander, Goldman Sachs, Mitsubishi UFJ Financial Group, UBS and Sumitomo Mitsui Financial Group, for example, have participated in total at 34 ventures between 07/ 2015 and 07/ 2016, no German bank was involved here [9][9]. It seems that these banks have understood that they cannot cope with FinTech companies, but they try to participate in their success based of disruptive business models.

In 2016, the total volume of venture capital investments in Germany amounted to approximately € 934 million [10], which is relatively low in international comparison. Up to the second quarter of 2017, € 21.8 billion of venture capital across 1.963 deals was recorded for the USA, whereas for the same period „only“ € 4.1 billion of venture capital across 589 deals was recorded for Europe [11]. Thus, Germany is too weak in comparison with other countries within the EU. This applies even more in comparison with the USA.

The gap becomes more obvious when we are looking at the following example; 7 out of 15[10] of the most important platform companies financed by venture capital originate USA, another 5[11] come from China, another 2[12] come from India and 1[13] come from Sweden [9][14]. In total, these more or less privately held companies have a combined market capitalization of more than US$ 300 billion. Only one of these companies originates from Europe. One of the most important competitors of Uber, Didi Chuxing from China, has received in 2016 US$ 7.3 billion additional venture capital. Thus, a single financing round in 2016 exceeded more than seven times the volume of the total venture capital investment in Germany in 2016.

[9] Page 158.

[10] Uber, Airbnb, WeWork, Pinterest, Dropbox, Stripe, Lyft.

[11] Xiaomi, Didi Chuxing, Lu.com, China Internet Plus, Toutiao.

[12] Flipkart and Snapdeal.

[13] Spotify.

[14] Page 36.

The following chart illustrates the top 15 privately held platform companies based on market capitalization.

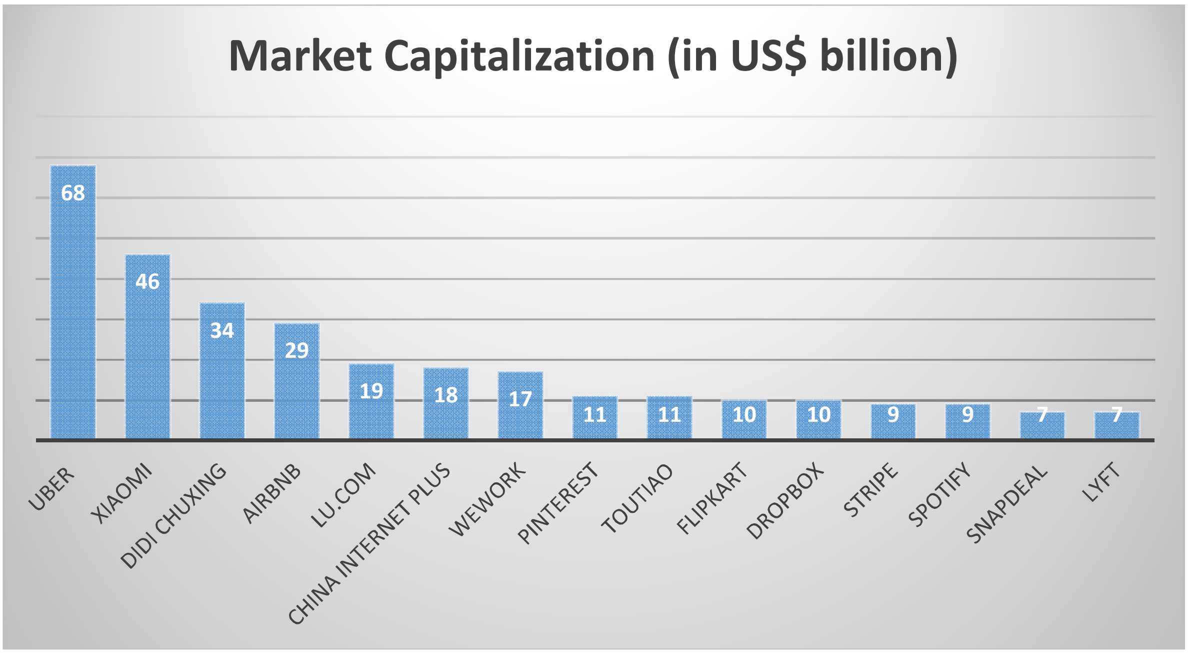

Figure 1-3

For example, now we selected the ridesharing/technology companies from the above graphic and compared them with the market capitalization of the leading car manufacturers.[15]

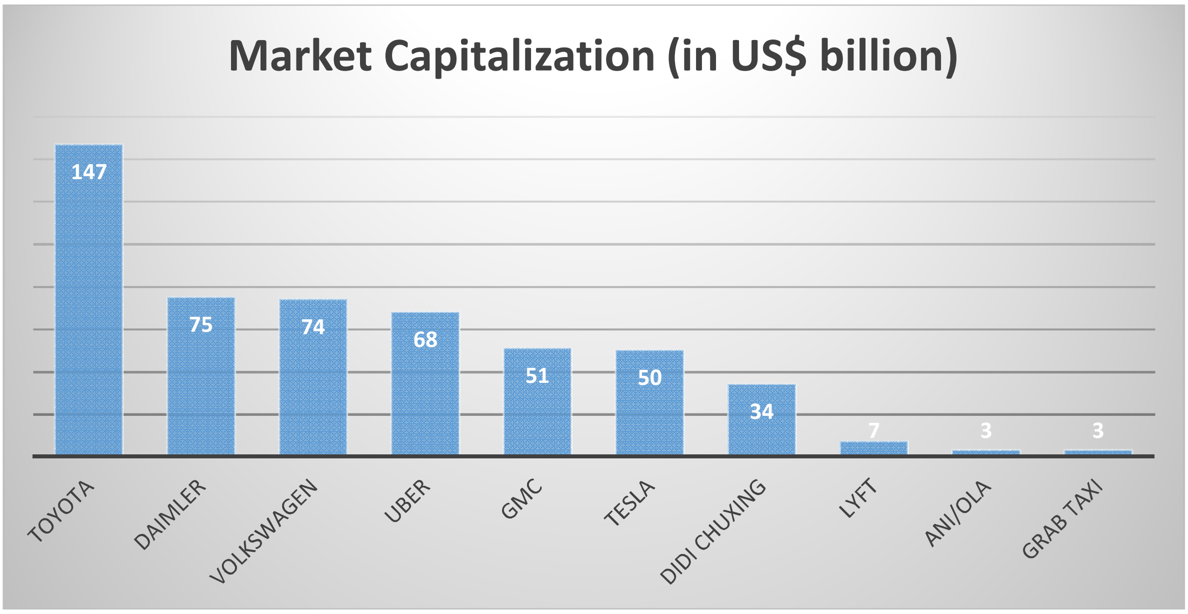

Figure 1-4

Six out of 10 companies with the highest market capitalization in the automotive industry are technology companies.

[15] As of the reporting date April 2017.

The automotive industry is the most important sector in Germany and none of these technology companies originated from a huge challenge from a German perspective.

Disruptive technologies are increasingly changing the market. More and more start-ups in the market enter traditional banking areas such as payment transactions, lending, retail banking or asset & wealth management. This is one side of the coin. The other side is that with Brexit new opportunities will rise for FinTech companies in Germany.

In our view, the banking market will be increasingly subjected to digital transformation, whereas the current economic conditions accelerate this process. Further key challenges for the German banking industry are the European sovereign debt crisis, the increased risk of terrorism and the associated consumer confidence in the banking sector, the increased regulatory requirements by the supervisory authorities and the need to ensure that the industry is able to meet these challenges [6][16]. **In view of these facts, it is a challenging outlook for the German banking industry.**

In the next chapter we will explain some mathematical basics of deep learning.

[16] Page 8, 9.

2. The math behind deep learning

In this chapter, we will introduce the fundamental mathematical theory of deep learning. As mentioned in the last chapter, data analysis is highly required in the rise of big data revolution. A classical problem for data scientists is **pattern classification (pattern recognition)**, which aims mainly at building a predictive model to classify data based on features from input data. Two examples are illustrated in Figure 2-1. The Decision Trees create a predictive model consisting of decision rules referred from statistical information of training data; Data points are separated by a nonlinear boundary in the second example.

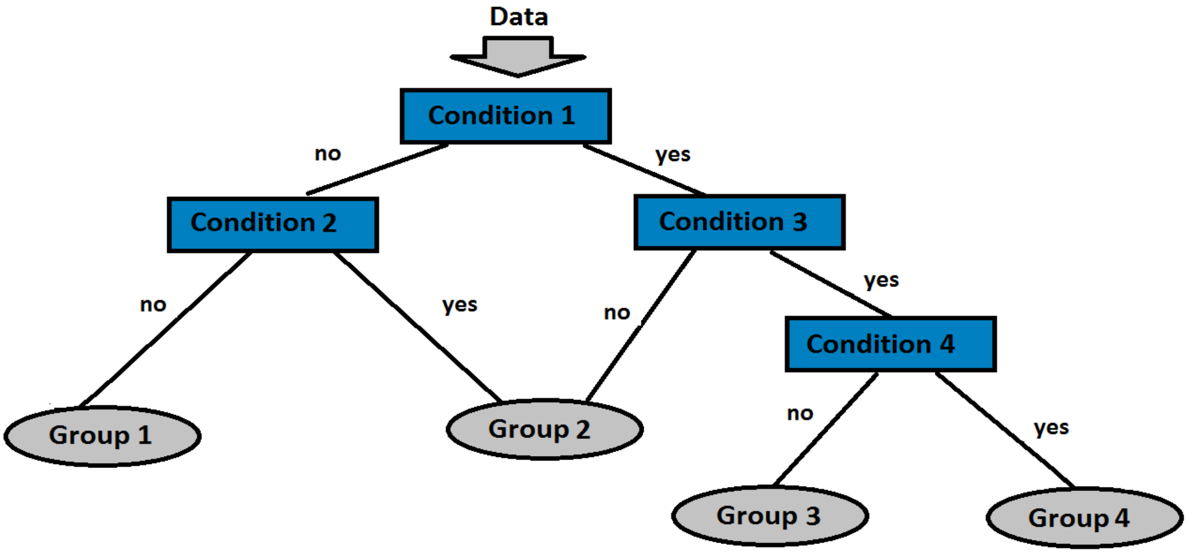

Figure 2-1-1   Example of pattern classification – Decision Trees

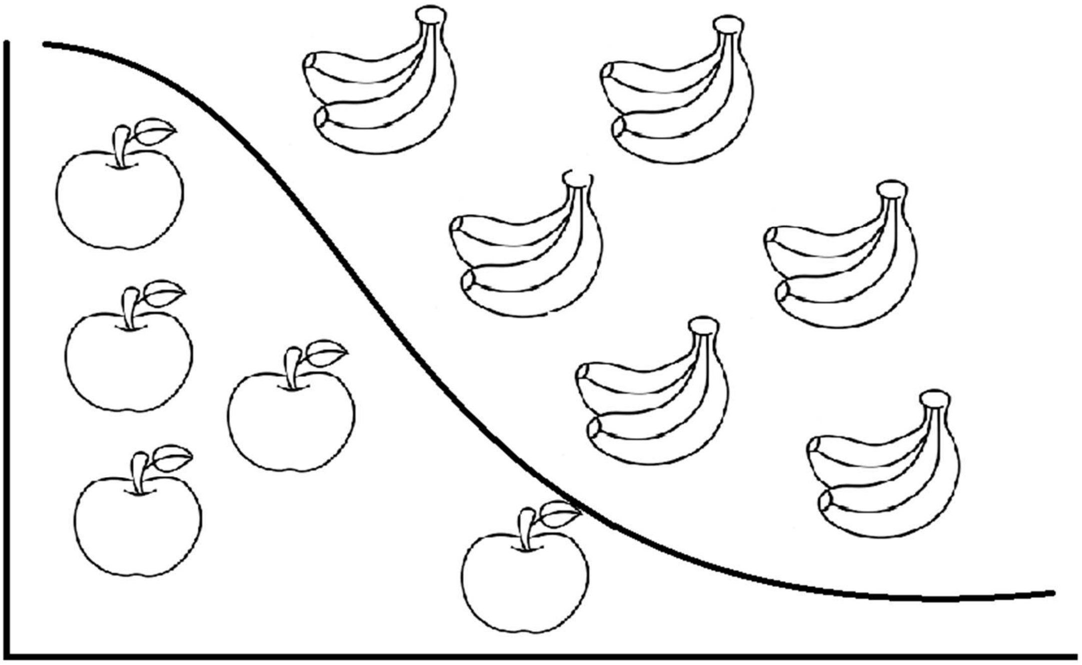

Figure 2-1-2   Example of pattern classification – Nonlinear Classification

Machine learning is a technique where a model automatically trains itself on a given dataset. This technique plays a key role in solving classification and recognition problem and it is widely studied in the last couple of years.

Some well-known algorithms are given as N-neighbor-algorithms, decision trees, logistic regression, etc. As a new trend of machine learning, deep learning showed up at the end of last century and became one of the most efficient machine learning algorithms. As the name "deep" suggests, deep learning subtracts features and represents data automatically by means of **multiple hierarchical levels**. The more levels are used, the more precise are the deep learning algorithms, however with a longer training time.

We continue this chapter by introducing **the classical gradient descent algorithm approach**, which is a popular mathematical approach in solving optimization problems and fundamentally supports training processes for most of the deep learning algorithms. **Artificial neuro network**, which has strongly powered deep learning in the last years, is going to be presented in the second sub-chapter. Inspired by the structure of a human brain, neural networks are built up by a number of neurons, where each neuro receives and transforms information based on its own activity. The neuro network trains itself automatically by adjusting the (transformation-)parameter iteratively until the performance reaches a user-defined boundary. However, training and optimizing a neural network is always challenging with regard to complexity and accuracy. In the third sub-chapter, we introduce some variations of training algorithms, which improve the classical gradient descent algorithm and strongly speed up the training process of deep learning.

2.1 Gradient Decent Algorithm (GDA)

Before starting to introduce GDA, we need to run through the following mathematic definitions:

- What is a Derivative ?
- What is a Partial Derivative?
- What is a Gradient?

Definition (Derivative): Let $f(x)$ be a function and $x0$ be a variable. The derivative of function $f(x)$ at point $x0$ is denoted by $f'(x0)$ and it is defined as the limit of,

$$f'(x0) = lim_{x \to x0} \frac{f(x)-f(x0)}{x-x0}.$$

Simply interpreted by using *Figure 2-2*, the derivative measures the slope of the tangent line (red line Tx) of the function $f(x)$ at point $x0$. On the other hand, the derivative represents the sensitivity of changes of the function f(x) with respect to the input x. With greater derivative, the function increases stronger as the input x increases.

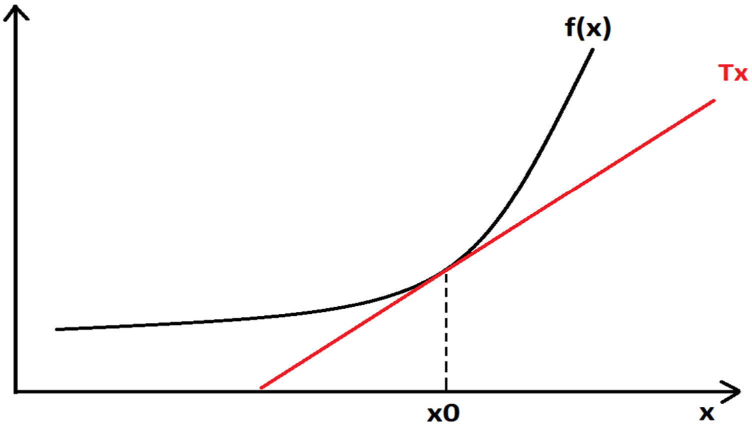

Figure 2-2 Derivative of the function f(x)

We move on to a two-dimensional function $f(x, y)$ (see *Figure 2-3*). Different from the one-dimensional case, there exists an infinite number of tangent lines at one point $(x0, y0)$. In order to measure the sensitivity of changes of function $f(x, y)$ with respect to the input x and the input y, we insert the second definition of partial derivative.

Definition (Partial Derivative): Let $f(x, y)$ be a function and $x0, y0$ be variables. The partial derivative of function $f(x, y)$ at point $(x0, y0)$ with respect to x (or y) is denoted by $\frac{\partial f}{\partial x}(x0, y0)$ (or $\frac{\partial f}{\partial y}(x0, y0)$) and defined as the limit of,

$$\frac{\partial f}{\partial x}(x0, y0) = \ lim_{x\to x0}\frac{f(x, y0) - f(x0, y0)}{x - x0}$$

$$\text{(or } \ \frac{\partial f}{\partial y}(x0, y0) = \ lim_{y\to y0}\frac{f(x0,y) - f(x0,y0)}{y - y0}\text{).}$$

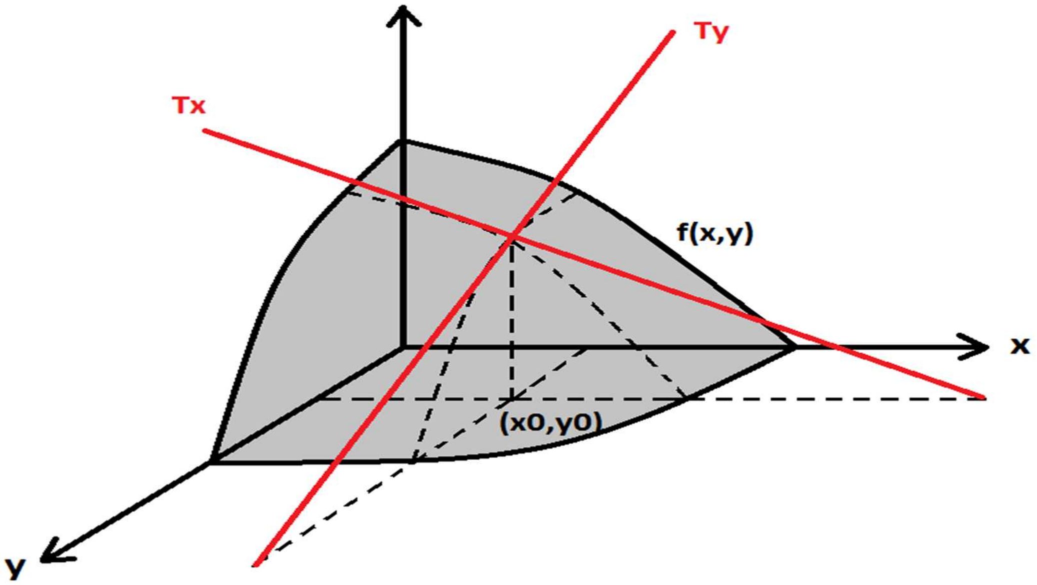

Figure 2-3 Partial derivative of function f(x,y)

The partial derivatives measure the slopes of the tangent lines that are parallel to the x and y axes respectively (red lines Tx and Ty in *Figure 2-3*).

As a simplified interpretation, partial derivatives represent the **change rate** of function $f(x, y)$ at point $(x0, y0)$ **in the direction of x and y axes, respectively.**

Note, partial derivatives are restricted to the directions parallel to x and y axes. A general question then arise: How to measure the change rate of the function $f(x, y)$ in any arbitrary direction and further how to find out the direction of the greatest change rate?

Definition (Gradient) Let $f(x, y)$ be a function and $(x0, y0)$ be variables. The gradient of function $f(x, y)$ is a vector-valued function denoted by $\nabla f(x0, y0)$ and is defined as

$$\nabla f(x0, y0) = (\frac{\partial f}{\partial x}(x0, y0), \frac{\partial f}{\partial y}(x0, y0))$$

where $\frac{\partial f}{\partial x}(x0, y0)$ and $\frac{\partial f}{\partial y}(x0, y0)$ are partial derivatives of $f(x, y)$ at point $(x0, y0)$ with respect to x and y.

The most attractive fact of gradient is presented in the following theory.

Theory (Gradient) [12][17] The gradient $\nabla f(x0, y0)$ points in the **direction of the greatest change** of the function $f(x, y)$ at point $(x0, y0)$.

As illustrated in the following Figure, the gradient (shown as the blue arrow) represents a direction at which the function $f(x, y)$ has the greatest change. Trivially, the direction against the gradient$-\nabla f(x0, y0)$, is the one in which $f(x, y)$ **decreases most steeply.**

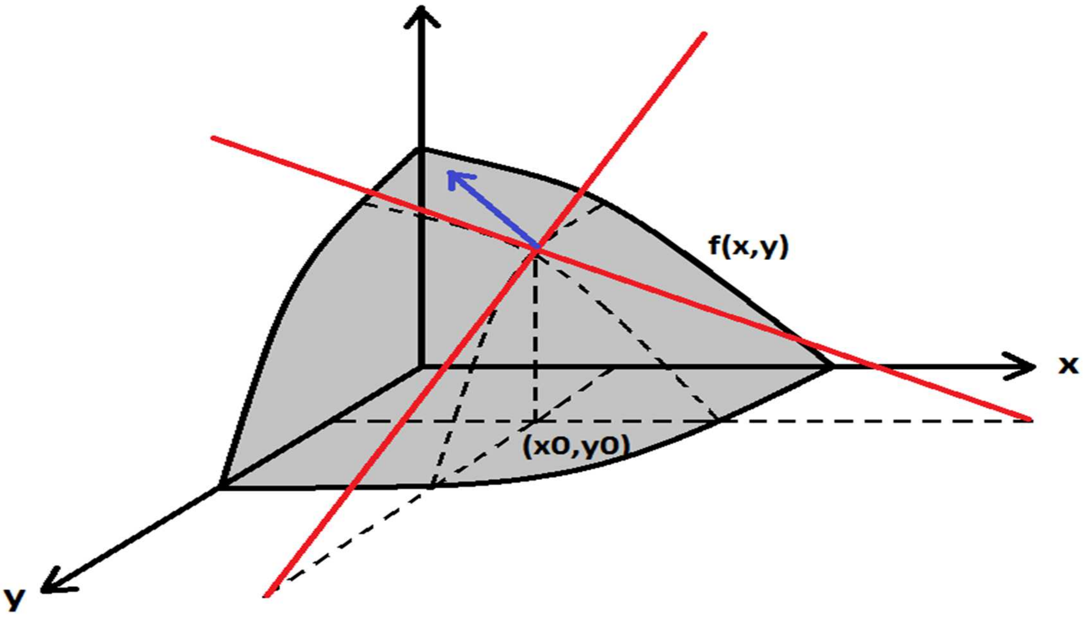

Figure 2-4 Gradient (blue arrow) of function $f(x, y)$

[17] Page 201.

Based on the mathematical background presented above, we move on to introduce the classical gradient descent algorithm. In general, the gradient descent algorithm is an **iterative optimization algorithm for obtaining the minimum of an objective function**. In order to go through the iterative updating process step by step, we provide two pictures in *Figure 2-5*, where the left one is a three-dimensional surface and the right one illustrates the corresponding two-dimensional diagram of contour lines. Assume that we start at point a0 and aim at looking for the minimum of the objective function.

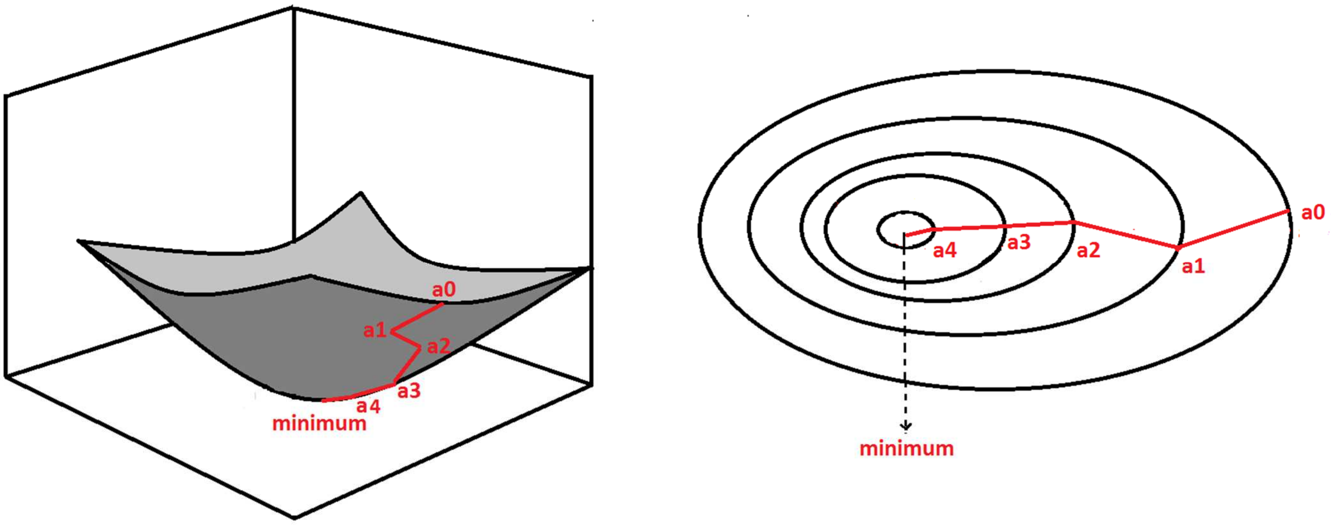

Figure 2-5 Gradient descent algorithm of function f(x,y)

The gradient descent algorithm is proceeding iteratively as follows:

- Step 0: Getting a start point $\alpha 0$

- Step 1: Compute the gradient at $\alpha 0$, namely $\nabla f(\alpha 0)$ and choose a constant γ>0. We adjust the point by

$$\alpha 1 = \alpha 0 - \gamma \nabla f(\alpha 0).$$

- Step 2: Compute the gradient at $\alpha 1$, namely $\nabla f(\alpha 1)$ and choose a constant γ>0. We adjust the point by

$$\alpha 2 = \alpha 1 - \gamma \nabla f(\alpha 1).$$

- Step 3: Compute the gradient at $\alpha 2$, namely $\nabla f(\alpha 2)$ and choose a constant γ>0. We adjust the point by

$$\alpha 3 = \alpha 2 - \gamma \nabla f(\alpha 2).$$

- Step N+1: Compute the gradient at αN, namely $\nabla f(\alpha N)$ and choose a constant γ>0. We adjust the point by

$$\alpha N + 1 = \alpha N - \gamma \nabla f(\alpha N).$$

In *Figure 2-5*, the red polylines (in both subpictures) represent the iterative process introduced above. In each step the gradient determines the direction towards the steepest slope of the surface and a positive constant γ is chosen for the step size of movement.

Note, we always subtracted the gradient since we are going to move down **toward the minimum**, namely **against the gradient.** The sequence αN is updated iteratively and **converges to the minimum** of the surface.

It is possible that the gradient decent algorithms may converge into a local minimum instead of the goaled global minimum. A local minimum is a minimum located around a set of neighbor points (see Figure 2-6). There are some approaches for allowing the algorithms keep updating and escaping from the local minimum, for example stochastic gradient decent[18].

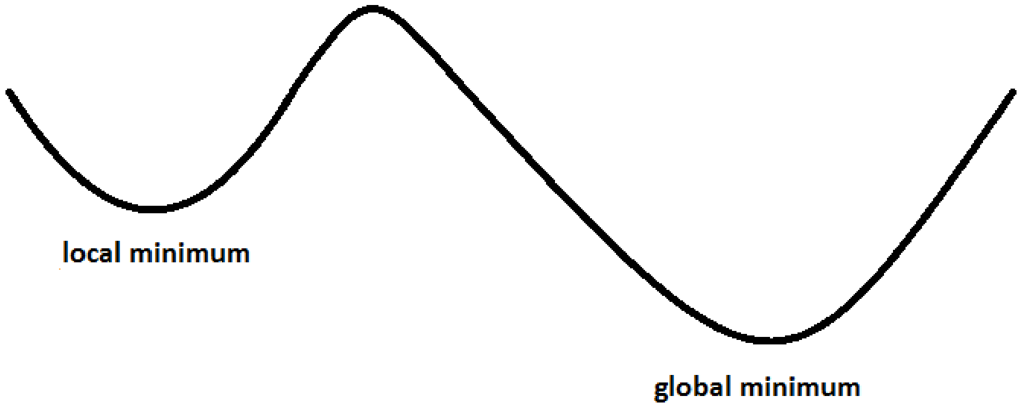

Figure 2-6 local minimum/global minimum

The classical gradient descent algorithm presented above always faces challenges in the requirement of performance. Several variations were developed in the last couple of years in order to improve the convergence rate. We will introduce some of them at the end of this chapter.

[18] [22].

2.2 Feedforward Neuro Network- Architecture

Artificial neuro network powered deep learning in the last years and it is one of the most popular machine learning techniques at present. In this sub-chapter, we present one of the most common artificial neuro network architectures: **Feedforward Neuro Network (FNN).** Some other artificial neuro network architectures, such as the convolutional neuro network, will be introduced in Excursion 2.4.5.

As a traditional neuro network architecture, FNN has been studied for more than three decades. A typical FNN architecture is illustrated in *Figure* 2-7, which consists of one input layer (Layer K), one (or more) hidden layers (Layer H) and one output layer (Layer L). The Input layer is the layer where the external training data enter into the network. The hidden layers in-between are hidden away from the “outside” and deliver the information flow “forward” from the left layer to the right one. In *Figure* 2-7, we have only one hidden layer. The Information flow ends at the output layer. Since the information flow moves in one direction, the neuro network in this architecture is usually known as **“Feedforward**”.

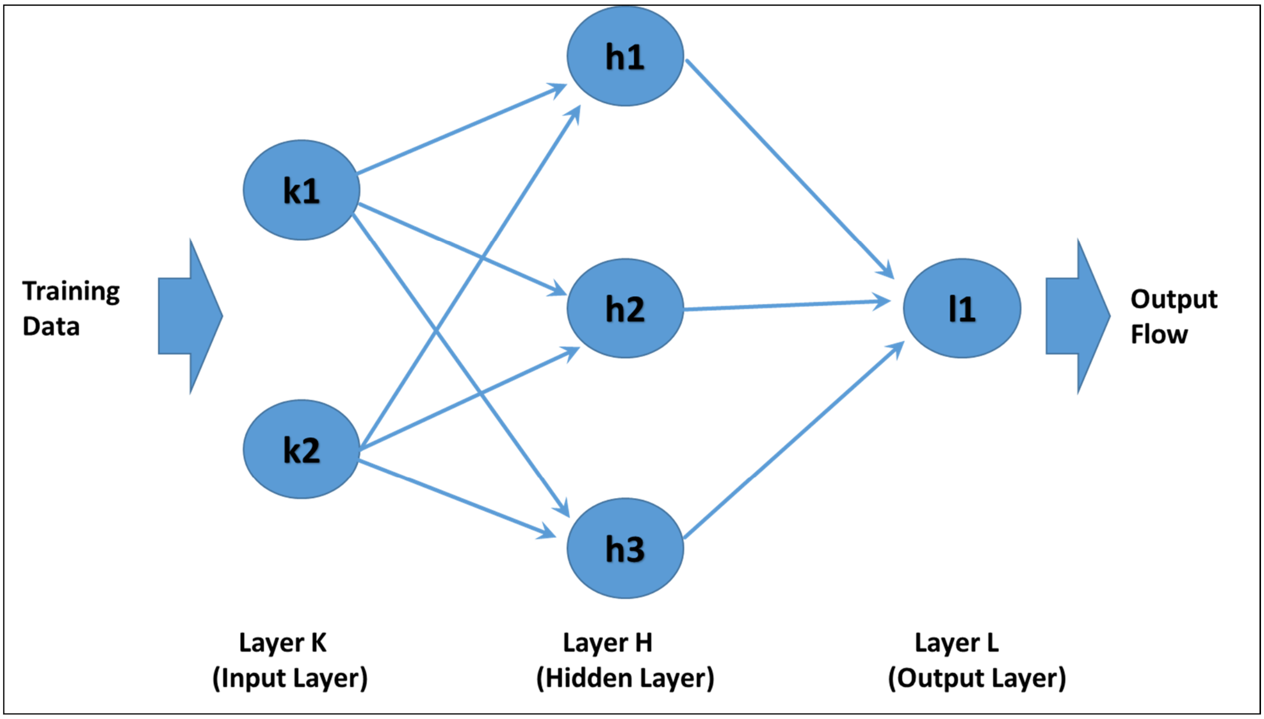

Figure 2-7 Architecture of Feedforward Neuro Network

In FNN, each layer is formed by several neuros and each neuro is able to obtain, adjust and deliver information to neuros in the next layer. Two successive layers are connected by a **weighted average procedure**, which averages the output of neuros from the previous layer according to their weights and provides a linear classification. Another important feature of FNN is the **activation function** inside each neuro, which represents the activation level of the neuro. The activation function is usually non-linear enabling FNN to produce non-linear classification. In the following, we will introduce both features in detail.

Some notations are required through the remaining part of this chapter: Let h be a neuro in FNN and k be a neuro from the previous layer of h, we denote,

- by $Input_h$ as the input flow of h
- by $Output_h$ as the output flow of h
- by w_{kh} as the weight between k and h.

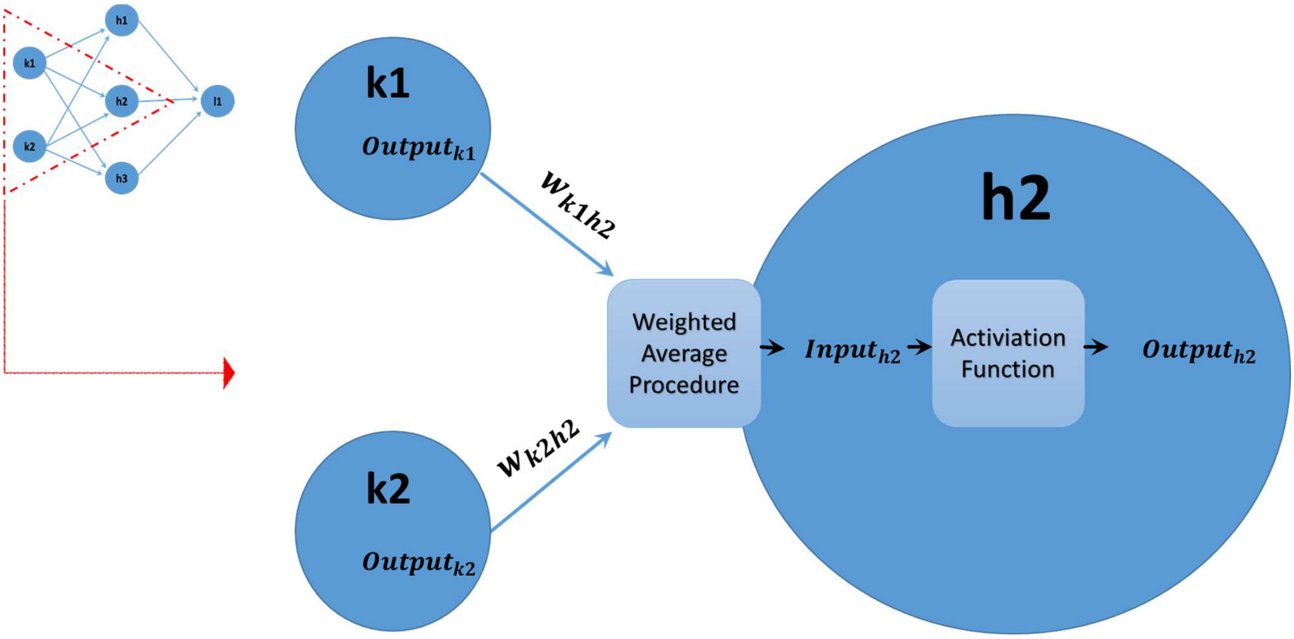

Figure 2-8 Data flow in Neuro h2

Taking neuro $h2$ as an example, *Figure 2-8* illustrates the input flow, output flow of $h2$ and weights w_{k1h2}, w_{k2h2} between $h2$ and neuros from the previous layer. We present the formal definitions of the Weighted Average Procedure and Activation Function.

Definition (Weighted Average Procedure) Given neuro $h2$, the weighted average procedure computes the $Input_{h2}$ by averaging $Output_{k1}$ and $Output_{k2}$ weighted by w_{k1h2} and w_{k2h2}, respectively. Accordingly, we have

$$Input_{h2} = Output_{k1} * w_{k1h2} + Output_{k2} * w_{k2h2}.$$

Definition (Activation Function) Given neuro $h2$, the activation function $Act()$ determines the output flow of a neuro given the input flow:

$$Output_{h2} = Act(Input_{h2}).$$

In biological neuro network, the activation function determines whether and how strong the neuro is turned on.

Some popular activation functions are listed as follows:

a) Step function:

$$Act(x) = \begin{cases} 1, & x \geq 0 \\ 0, & x < 0 \end{cases}$$

This function takes the value 1 (neuro is activated) for values of x>0 and the value 0 (neuro is deactivated) otherwise.

b) *Logistic function*:

$$Act(x) = \frac{1}{1+\mathrm{e}^{-x}}, \quad for -\infty < x < \infty.$$

This function rises monotonically and takes value between 0 and 1.

c) *Relu function* (**re**ctified **l**inear **u**nit):

$$Act(x) = \begin{cases} x, & x \geq 0, \\ 0, & x < 0. \end{cases}$$

This function rises linearly for values x>0 and it is 0 otherwise.

d) *Tanh function*:

$$Act(x) = \frac{2}{1+\mathrm{e}^{-2x}} - 1, \quad for -\infty < x < \infty.$$

This function rises monotonically and takes values between -1 and 1 [13][19].

Using a linear activation function in FNN has no sense, since non-linear classification problems, such as XOR problem, will be never solved correctly by using FNN with linear activation, no matter how deep the FNN is built and how well the network is trained. A simple example of non-linear classification was illustrated in *Figure* 2-1.

[19] Page 15-18.

2.3 Feedforward Neuro Network – Training process

From now on, we are going to study the learning process of FNN. In the training process, the weighting factors between the nodes are iteratively adjusted until a user-defined convergence level is reached.

Suppose that

- we have a set of training data $\{(X^t, Y^t), t \in T\}$, where T is the number of examples in the training set, X^t is the Input variable and Y^t is the target output variable to be predicted;
- we are aiming at training the FFN illustrated in *Figure 2-9*. The set of weights, denoted by $w = (w_{k1h1}, w_{k1h2}, w_{k1h3}, w_{k2h1}, w_{k2h2}, \dots \dots)$, are to be optimized in the iteration process, given the test input and output
- we denote each training example X^t by O^tas the corresponding output of FNN.

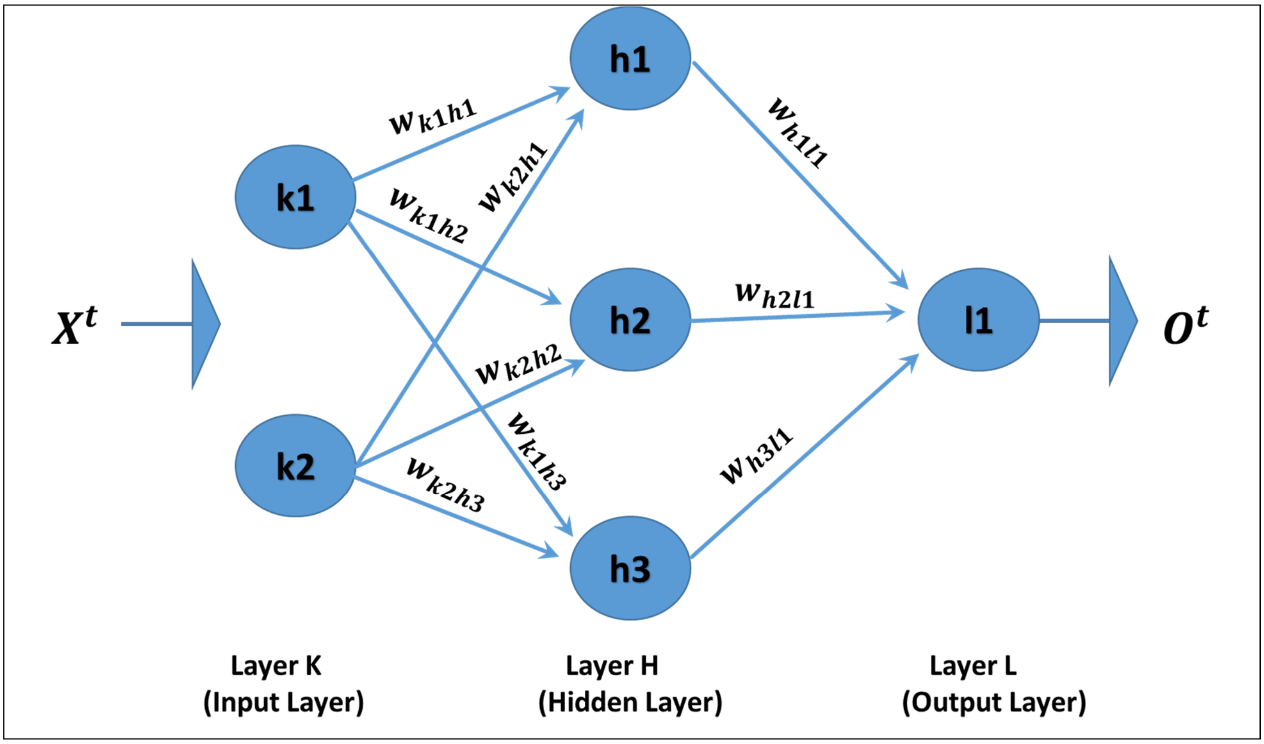

Figure 2-9 Feedforward Neuro Network

In order to evaluate the performance of a neuro network, we need to introduce the **loss function**. The loss function calculates the “gap” between the target output (to be predicted) and the output of FNN. Simply interpreted, the loss function measures how well the neuro network is trained and how well the weights are estimated. There are various choices of loss functions.

In this work, we introduce a most commonly used loss function based on the squared loss.

Definition (Loss function) The loss function is denoted by $Err()$ and is defined as

$$Err = \sum_{t\in T}(O^t - Y^t)^2.$$

Reading from the definition above, the smaller the loss function $Err()$, the better the neuro network is trained. Thus, the training process of FNN has the target of searching for optimal weights $w = (w_{k1h1}, w_{k1h2}, w_{k1h3}, w_{k2h1}, w_{k2h2} \dots \dots)$ to minimize the loss function $Err()$. The gradient descent algorithm, introduced at the beginning of this chapter, is one of the most popular and standard techniques for solving such an optimization problem. We recall that the gradient descent algorithm starts at a guessing point and adjusts the point iteratively against the direction of gradient until the objective function is minimized. Applying gradient descent algorithm for training neural network, we have:

The training process (update process) of FNN:

- Step 0: Choosing starting value of weights w^0

- Step 1: Compute the gradient of Loss function at w^0, namely $\nabla Err(w^0)$ and choose a constant γ>0. We adjust the weights by

$$w^1 = w^0 - \gamma * \nabla Err(w^0).$$

- Step 2: Compute the gradient of Loss function at w^1, namely $\nabla \; Err(w^1)$ and choose a constant γ>0. We adjust the weights by

$$w^2 = w^1 - \gamma * \nabla Err(w^1).$$

- Step N+1: Compute the gradient of Loss function at w^N, namely $\nabla Err(w^N)$ and choose a constant γ>0. We adjust the weights by

$$w^{N+1} = w^N - \gamma * \nabla Err(w^N) \qquad (1)$$

The sequence w^N converges to the minimum. Normally, we train the neural network until the loss function reaches a user defined preference boundary. In the world of neuro network, γ is usually called **learning rate.** Next, it is necessary to answer how to adjust weights by calculating the gradient of the loss function.

The next theorem, which has been mentioned almost in each publication of FNN, allows us to calculate the gradient of loss function by using **back propagation algorithm.**

Theorem (Back Propagation Algorithm) [14][20] The gradient of Loss function ∇Err is defined as

$$\nabla Err = (\frac{\partial \nabla Err}{\partial w_{k1h1}}, \frac{\partial \nabla Err}{\partial w_{k2h1}}, \frac{\partial \nabla Err}{\partial w_{k1h2}} \dots \dots .),$$

where each partial derivative above is computed by

$$\frac{\partial \nabla Err}{\partial w_{pn}} = Ouput_p \delta_n.$$

for each pair of two successive neuros (p, n) illustrated in *Figure 2-9,* where p stands for the one delivering data flow and n is for the one receiving data flow (For instance $p = k1$ and $n = h1$).

Further, $\delta_n$is called **Error Signal** and can be **back propagated** by

$$\delta_n = \begin{cases} \sum_{t \in T} 2 * Act'(Input_n) * (Output_n - Y^t), & if\ n\ stays\ in\ Output\ Layer \\ Act'(Input_n) * \sum_{for\ every\ neuro\ l\ of\ the\ next\ layer} w_{nl}\, \delta_l, & otherwise \end{cases}$$

where Act' is the derivative of the activation function.

In order to understand the mathematical formula above, we illustrate the process for propagating Error Signal δ_n in the following *Figure 2-10*.

We start from the Output Layer and compute the Error Signal δ_{l1} at first. The Error Signal is then propagated **from the right layer to the left layer against** the direction in which the data flow is delivered. For Example, the Error Signal for neuro $h1$, namely δ_{h1}, is computed by using Error Signal of Layer L by

$$\delta_{h1} = Act'(Input_{h1}) w_{h1l1} * \delta_{l1}.$$

The Error Signal for neuro $k1$, namely δ_{k1}, is computed by using Error Signals from Layer H as,

$$\delta_{k1} = Act'(Input_{k1})(w_{k1h1} * \delta_{h1} + w_{k1h2} * \delta_{h2} + w_{k1h3} * \delta_{h3}).$$

[20] Page 89-95.

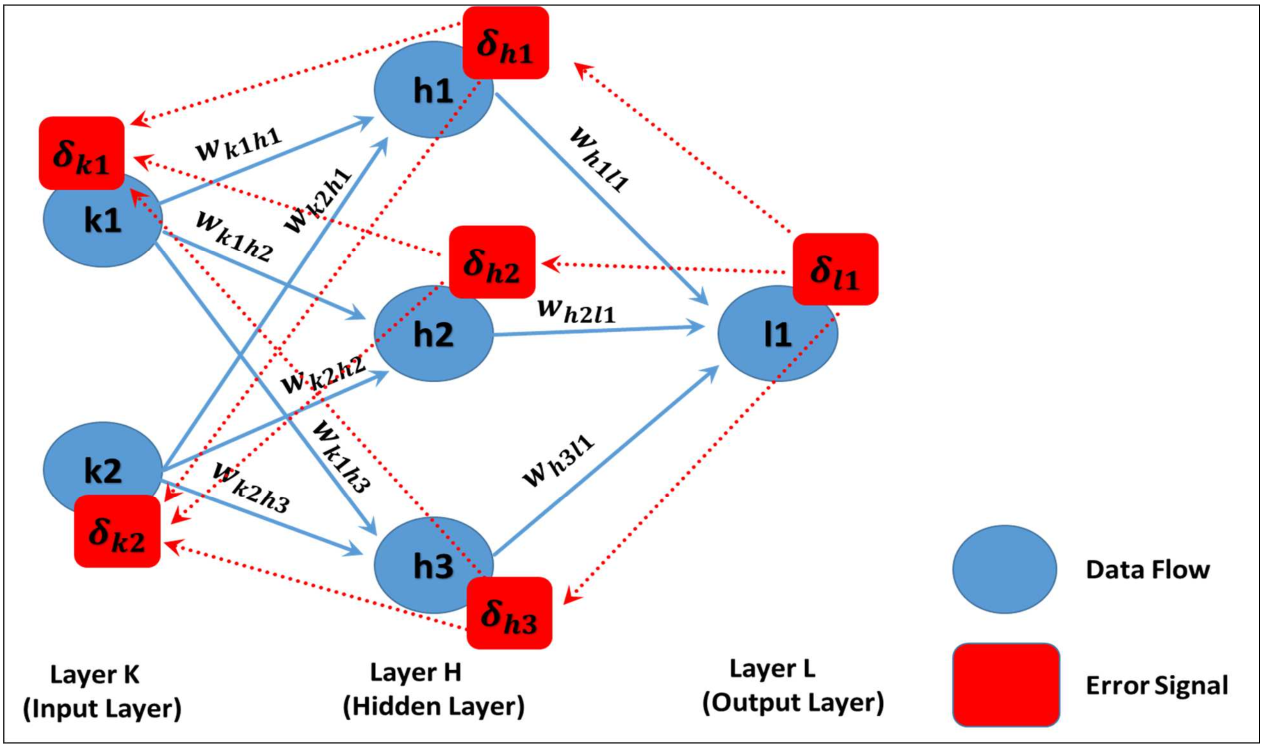

Figure 2-10   Back propagation of Error Signal

Note, the more layers are involved in a neuro network, the weaker the propagated error signal in layers closed to input layer. As a result, the training process becomes more expensive.

2.4 Feedforward Neuro Network – Some Improvement and variations of training process

Recalling the classical gradient descent algorithm, we move iteratively downhill against the direction of gradient until we reach the minimum of the surface. Accordingly, we have the update process,

$$w^{N+1} = w^N - \gamma * \nabla Err(w^N),$$

where ∇Err is the gradient of the error function at point w^N and γ is the learning rate that determines the step size we take. For the purpose of consistency, we rewrite the above equation into,

$$\begin{aligned} w^{N+1} &= w^N + v^{N+1} \\ v^{N+1} &= -\gamma * \nabla Err(w^N). \end{aligned} \tag{1}$$

with an additional **update term** v^{N+1}.

The classical gradient decent algorithm is simple to implement but performs sometimes not as well as we expect. One example is presented in *Figure* 2-11. The alternative approach (the one on the right) requires fewer iterative steps und reaches the minimum even faster. In this sub-chapter, we are going to introduce some variations of the classical gradient algorithm, which have significantly better convergence rates to the minimum and are currently worldwide used in most training processes of neuro network.

2.4.1 Momentum

Momentum enables the gradient descent algorithm to keep the memory about the previous update. Accordingly, the update process is given as,

$$\begin{aligned} w^{N+1} &= w^N + v^{N+1} \\ v^{N+1} &= \mu^N v^N - \gamma * \nabla Err(w^N). \end{aligned} \tag{2}$$

where $0 \leq \mu^N \leq 1$ is the momentum parameter.

In comparison to the classical gradient descent algorithm, the update term v^{N+1} is the linear combination of current gradient and previous update. The Momentum parameter μ^N determines how much the update information of the previous step is taken account into the current step. Comparing the equation (1) and (2), it is not hard to get that the classical gradient descent algorithm is just the case of $\mu^N = 0$. Normally, μ^N is set initially to 0.5 and increased to 0.9 when the learning process stabilizes.

As we illustrated in *Figure* 2-11, the classical gradient descent algorithm cannot guarantee a good convergence rate by suffering from a zigzag update path. Momentum smooths the optimization path by adding an additional term $\mu^N v^N$. This leads to a faster and more efficient convergence rate and improves the performance of gradient descent algorithm.

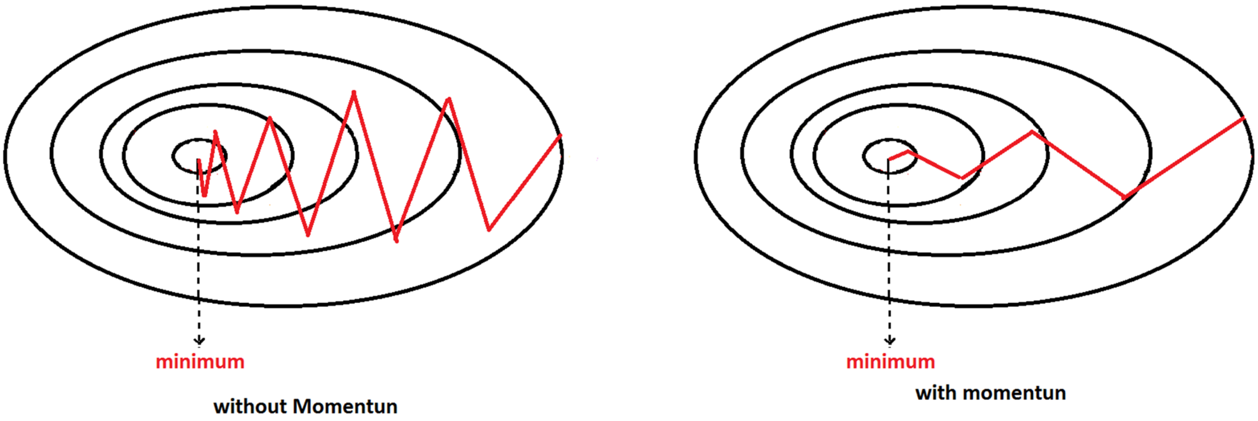

Figure 2-11 Comparison of the update process without/with momentum

2.4.2 Nesterov Accelerated Gradient

NAC is an improved gradient descent algorithm based on Momentum and has the update process given with,

$$w^{N+1} = w^N + v^{N+1}$$
$$v^{N+1} = \mu^N v^N - \gamma * \nabla Err(w^N - \mu^N v^N).$$

In comparison to the classical gradient descent algorithm, we compute the gradient at point $w^N - \mu^N v^N$ instead of w^N.

The intuition of NAC is to adjust the current update by looking ahead. In other words, we move our current point forward $-\mu^N v^N$ by using momentum and check around. If the surface sloped down there, we have to attend larger steps; on the contrary, we need to slow down [15][21].

2.4.3 Adagrad

It is always challenging to choose the learning rate for the update process appropriately. The well-known Newton-Method is able to provide the optimal choice of learning rate, however, it is computationally impractical since computing and inverting huge Hessian matrix is very expensive. Especially in multi-parameter cases, it is less efficient to set the learning rates consistent during the training process among all parameters. Adagrad is a method, which is simple to implement and enables to scale a global learning rate dynamically in each parameter based on its gradient history.

Assume that there are M parameters $w = (w_1, w_2, \dots . w_M)$ in neuro network. The update process of Adagrad for parameter w_i is given by

$$w_i^{N+1} = w_i^N + v_i^{N+1}.$$

[21] Page 1899-1903.

$$v_i^{N+1} = -\frac{\eta}{\sqrt{\sum_{n=0}^{N}\left|\nabla Err(w_i^n)\right|^2}} * \nabla Err(w_i^N),\ \text{for}\ i = 1,2 \dots M \tag{3}$$

where η>0 is the **global learning rate** used by all parameters and $\sqrt{\sum_{n=0}^{N}\left|\nabla Err(w_i^n)\right|^2}$ is the sum of square of all previous gradients.

In comparison to the classical gradient algorithms (see equation (1)), the term $\frac{\eta}{\sqrt{\sum_{n=0}^{N}\left|\nabla Err(w_i^n)\right|^2}}$ enables the update process to adapt the learning rate for each parameter according to its gradient history. Namely, it provides large update for parameters with small gradients in the past and oppositely small update for parameters with large gradients.

Unfortunately, the square sum $\sum_{n=0}^{N}\left|\nabla Err(w_i^n)\right|^2$ will increase continually as the training process proceeds. Because of this, the adapted learning rate $\frac{\eta}{\sqrt{\sum_{n=0}^{N}\left|\nabla Err(w_i^n)\right|^2}}$ will keep getting smaller und finally stop the training process by ending at zero. Adadelta method introduced in the coming part of this sub-chapter will overcome this problem.

2.4.4 Adadelta

As mentioned above, Adagrad adapts a learning rate based on an accumulation of squared gradients of the whole history but it has the problem of continual decay of learning rates. As an extension of the Adagrad method, the Adadelta method avoids the continual decay of learning rates by restricting the accumulation window to a fixed length ω. Instead of storing ω, Adadelta implements the restricted accumulation by computing a running weighted average with exponentially decaying weights [16].

The update process of Adadelta is given by,

$$w_i^{N+1} = w_i^N + v_i^{N+1}$$
$$v_i^{N+1} = -\frac{\sqrt{RMS[|\mathrm{v_i}|^2]_N + \epsilon}}{\sqrt{RMS[|\nabla Err(w_i)|^2]_N + \epsilon}} * \nabla Err(w_i^N),\ \text{for}\ \ i = 1,2 \dots M$$

where

- $\epsilon > 0$ is a constant, added to avoid the denominator by zero and usually has the order of 10^{-8};

- $RMS[|\nabla Err(w_i)|^2]_N$ is the exponentially decaying running average of the past gradients and is defined by,

$$RMS[|\nabla Err(w_i)|^2]_N = \rho * RMS[|\nabla Err(w_i)|^2]_{N-1} + (1-\rho) * \left|\nabla Err(w_i^N)\right|^2 \tag{4}$$

  with a delay rate $0 < \rho < 1$;

- $RMS[|v_i|^2]_N$ is the exponentially decaying running average of the past updates and is defined by,

$$RMS[|v_i|^2]_N = \rho * RMS[|v_i|^2]_{N-1} + (1-\rho) * \left|v_i^N\right|^2$$

with delay rate $0 < \rho < 1$.

We rewrite (4) into

$$RMS[|\nabla Err(w_i)|^2]_N = (1-\rho) * \left|\nabla Err(w_i^N)\right|^2 + (1-\rho) * \rho * \left|\nabla Err(w_i^{N-1})\right|^2 \\ +(1-\rho)^2 * \rho * \left|\nabla Err(w_i^{N-2})\right|^2 + \cdots + (1-\rho)^m * \rho * \left|\nabla Err(w_i^{N-m})\right|^2 + \cdots$$

Recalling the update process of Adagrade given in (3), the sum of squares in denominator of learning rate was given as,

$$\sum_{n=0}^{N} |\nabla Err(w_i^n)|^2 = \left|\nabla Err(w_i^N)\right|^2 + \left|\nabla Err(w_i^{N-1})\right|^2 \\ +\left|\nabla Err(w_i^{N-2})\right|^2 + \cdots + \left|\nabla Err(w_i^{N-m})\right|^2 + \cdots$$

Since $(1-\rho)$ is positive and smaller than one, $(1-\rho)^m\rho$ tends to zero as m increases. Comparing the above two equations, the accumulation window of Adadelta is no longer the whole gradient history due to the exponentially decaying weight $(1-\rho)^m\rho$.

2.4.5 An excursion to Convolution Neural Network (CNN)

A currently popular artificial neural network is the CNN, which improves the capabilities of the FFN mentioned above and is used especially for speech and image recognition. Convolution neural networks are organized in layers: an input layer is followed by a convolution layer and often by a subsampling or pooling layer (as shown in *Figure* 2-12). The mechanisms and architecture of these layers allow for the better capturing of spatial information of inputs and incorporate invariance properties into models involved- models are built in such a way that they are unchanged by certain transformations of inputs.

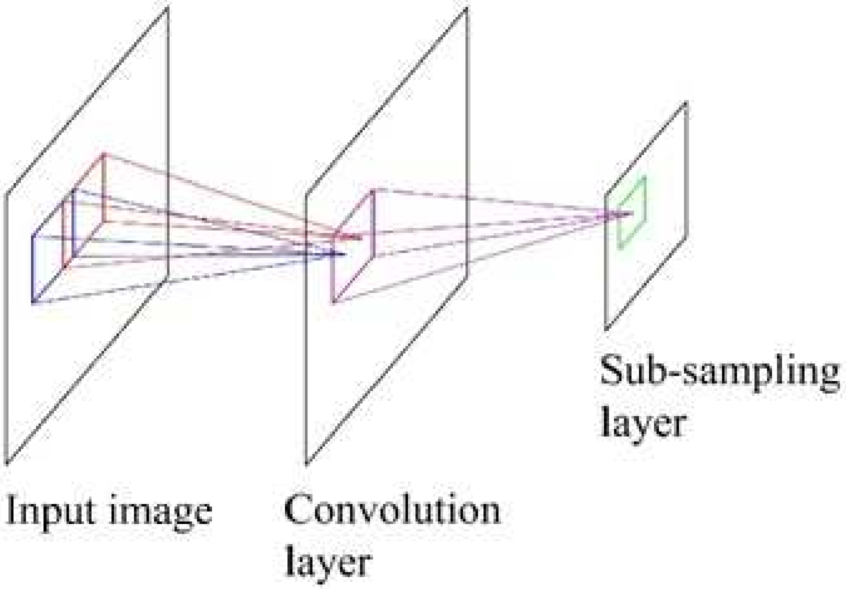

Figure 2-12 Convolution Neural Network

Furthermore, in a FFN, an input makes a full connection with the following hidden node in every neuron. This form of full connection between layers in a FFN prevents the capturing of the spatial information of an input. On the other hand, in a CNN, the input makes a connection with a small region of a hidden layer. This results in layers being connected by small fields called local receptive fields instead of being fully connected as in a Feed Forward networks, allowing for the capture of spatial information (shown in *Figure* 2-13) and beneficial uses such as speech and voice recognition [17].

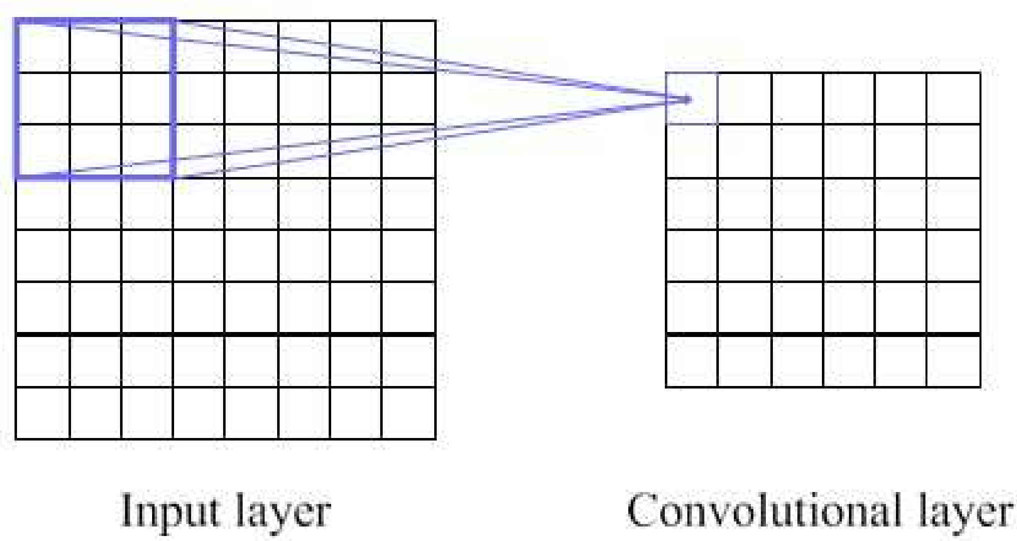

Figure 2-13

Another characteristic of a CNN is a feature map and its associated weight sharing. As mentioned above, layers in the network are connected through small regions called local receptive fields that allow for better capturing of information. More specifically, in the convolution layer, shown above, neurons are scattered in multiple parallel layers called feature maps, and each neuron in the feature maps is then connected to a local receptive field (shown in *Figure* 2-14) in a previous layer [17]. Within each feature map, all of the neurons share the same weigh parameter which contributes to the many computational benefits of these networks.

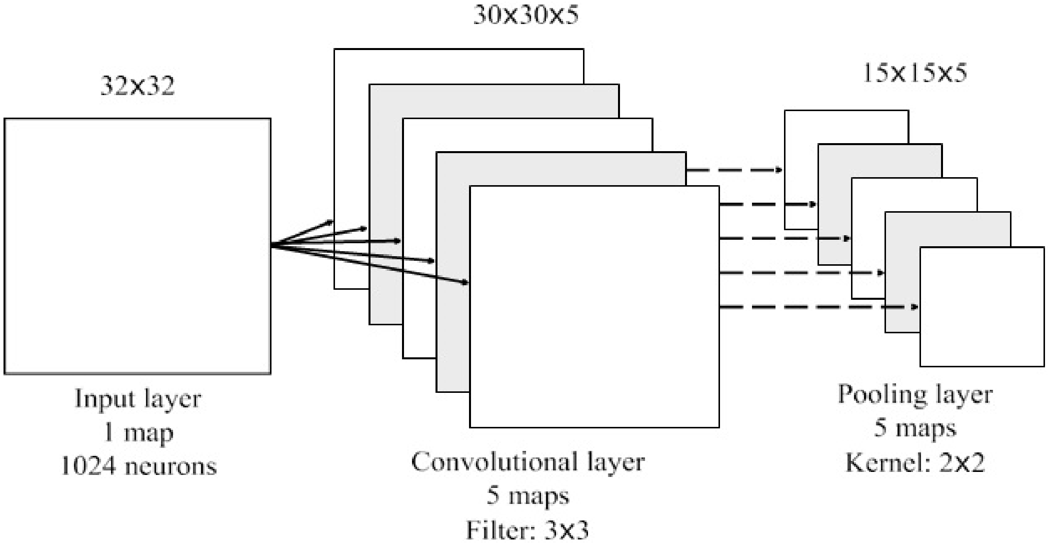

Figure 2-14

A final characteristic of a CNN is the concept of pooling and the translational invariance properties. As seen in *Figure* 2-14, CNN also may involve a pooling layer. This pooling layer receives outputs from the convolution layer (and associated hidden layers) and computes statistics based on the feature maps and localized regions, allowing for invariance properties to be present. *Figure* 2-15 below illustrates the connections between the convolution layer and the pooling layers [17].

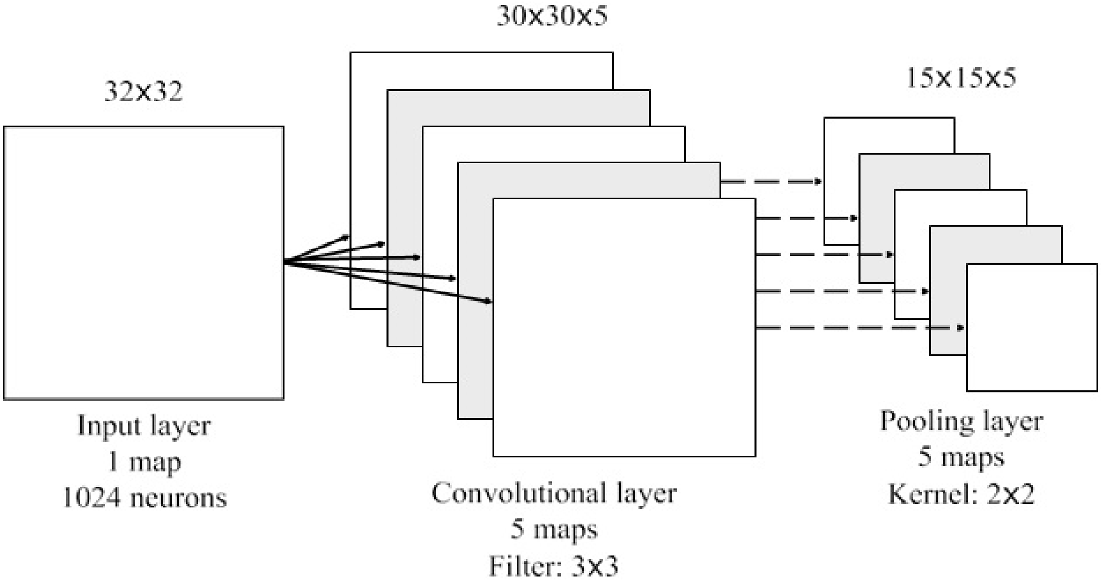

Figure 2-15

Generally, as discussed above, the use of these small regions to connect inputs to the convolution layers and feed the computations in the pooling layers allows for many of the computational and spatial information capturing benefits of the CNN.

2.5 Deep learning – The Trend

After the match between AlphaGo and Ke Jie, deep learning, especially artificial neural network, has become one of the hot topics around the world. Since then, more and more people would like to probe into the mathematics behind artificial neural network. In this chapter, we briefly introduced the mathematic fundamental theories within the artificially neural network. Inspired by the human brain, the artificial neural network has a layer-formed hierarchical architecture, where layers consist of neuros and interconnect with each other. The training of artificial neural network bases on providing a set of input data and telling the network what the output should be. An artificial neural network trains its internal connections between neuros by using gradient descent algorithms, where the connections are calibrated iteratively by using back-propagation until the performance (bias or error) reaches a user-defined level.

As more attention is paid to artificial neural network, people expect not only that the model could train by itself, but more efficient, fast and accurate training. In the past years, the classical gradient descent algorithms improved continuously and these variations **strongly speed up** the training process. We introduced some of them in chapter 2. Besides those, there are some other variants, for instance Adam [18], Nadam [19] and the conjugate gradient method [20].

In additional, the quick development of cloud environment and GPU hardware enable the technical implementation of deep learning**.** In the past years, a number of deep learning and machine learning frameworks were developed. Some of the most famous frameworks are given by Tensorflow developed by the Google Brain Team[22]; Caffe developed by the Berkeley Vision and Learning Center (BVLC)[23]; DMTK developed by Microsoft[24]. These frameworks are mostly open source, contain models with rich mathematical background and support for most of widely used language environment such as Java, C ++, Python and R. Furthermore, these frameworks are well designed for flexibility that allows users to develop and train their own task-specific models.

Successfully training artificial neural network requires at least a minimum amount of data. In the example of feedforward neural network, we are not able to calibrate weights between neuros without Input and Output Dataflow. As mentioned in the introduction, the banks are sitting on a treasure trove of data. This will enable the training of artificial neural network to proceed rapidly and effectively.

Thus, we believe intelligentization [21] will be a trend in the banking industry.

[22] https://www.tensorflow.org/

[23] http://caffe.berkeleyvision.org/

[24] http://www.dmtk.io/

3. Excursion to the topic Blockchain Technology and the interaction to Machine Learning

First, we explain very briefly, what is behind the Blockchain technology. We will then discuss the connection to machine learning.

We believe that misleading discussions about the Blockchain technology could be more of an obstacle than useful. If we take a closer look at the latest developments, we can see the following: Bitcoin also makes Blockchain part of an even bigger story. However, the story should be told in its entirety.

Major banks invest in their own technologies, participate in start-ups via venture capital, form consortia with other major banks and even conduct research on their own cryptographic currencies. This makes it clear that no one has a genuine vision. Nevertheless, no one wants to miss anything. In the media world, this creates more hype and myths than truth.

The particular appeal of the technology results from the variety of use cases. The technology enables real economic resources to be retained or used more effectively.

We have to distinguish between two main types of Blockchain; permissioned and permissionless block chains. The permissionless block chains are always suitable when it comes to ensuring a permanent, invariable and verifiable record of transactions that everyone can see. These cannot be edited. An example of an application could be in private use in a will, which would then no longer require notarial certification (Note that this would eliminate an important source of income for the typical notary in Germany). An essential element of the permissionless block chain in our example is that the beginning and the end would solely drive by the will of the testamentary with no more parties between.

The permissioned block chain is based on the element that generates a truth through algorithms, which can then be synchronized with any other parties or transaction points. In other words; the customer data could be directly notarized and synchronized with the land registry within the framework of a lending process and real estate acquisition. From the customer's point of view, the acquisition of real estate would thus be fully completed in the bank branch. This would result in a much more customer-friendly business model for a bank. Any customer in Germany would love that bank.

Before elaborating on the connection to artificial intelligence, let us explain the principle of “Smart Contracts”. In relation to Blockchain, "Smart Contracts" are often mentioned in the hype of the conversation; we already agree that the block chain as a technology can essentially contribute to keep records synchronized between multiple parties.

"Smart Contracts" based on Blockchain technology allow the automatic execution of essential contract terms and conditions. Less abstract, in the above-mentioned loan application process, at the same time as requesting his credit, the customer could schedule an annual repayment of the loan. Let us say the maximum possible amount is 12,000 € per year. Up to now the customer saves 1,000 € per month on a separate deposit account in order to be able to repay the unscheduled repayment each year.[25] In this case, the result could be that the "Smart Contract" is designed in such a way that all the described transactions are executed automatically. In other words, the contract is smart because all causal clauses in the contract would be subject to automatic execution.

Consequently, Blockchain Technology can produce great results. Even though we could only give a brief insight into the subject, we then should realize the huge opportunities and benefit for both, private and commercial sectors. For the commercial sector, we mentioned at the beginning that the golden source is the data. Data is the most important source for our business world - without data nothing works in the area of artificial intelligence. At the same time, a solid database is the substance of machine learning as a sub-area of artificial intelligence. In practice, companies have significant challenges with high-quality data. Blockchain could in fact be one of the key future sources of all data needed for anything.

From our point of view, Blockchain Technology could solve numerous problems at the same time by centralizing and digitizing the relevant data in a block chain. These datasets could then serve as an input layer for projects with deep learning implications. The advantage of this approach would be that the data quality of recognized block chains might be relevant for decision making when using artificial intelligence. Although banks sit on a large number of data, the fact remains that in practice these data are not available in sufficient quality (if data is available at all). In this respect, the centralization of data has an outstanding relevance in the banking industry; reduce costs, prevent the destruction of no longer meaningful business models, create new business models and be the main source of data so that algorithms can produce reasonable results. Thus, Blockchain could be a conceivable solution everywhere.

[25] An unscheduled repayment reduces, from the customer's point of view, the outstanding loan amount and at the same time its interest burden. From the bank's point of view, however, the unscheduled repayments are rather unpopular.

4. Conclusion

The first chapter provided an overview of the terms (artificial intelligence, machine learning and deep learning) as well as the economic framework conditions in Germany. In this respect, we have focused on the banking industry. The most important industrial sector for Germany (automotive industry) could not be ignored to clarify the competitive position in a context-related global comparison. From the point of view of the German banking industry, the result does not look particularly satisfactory. In the following second chapter, we have dedicated our attention to the math behind the myth of deep learning. We have devoted ourselves to this topic in more detail, because we believe that the majority of misunderstandings will disappear. In addition to theory, we have also dealt with practice. Especially with selected deep learning platforms, which are common in the corporate world. We are certain that deep learning will contribute to an erosion of current business models in the banking industry within the next five to ten years. This tendency can no longer be reversed. Although the prevailing literature is against the trend, we have ventured into a context between Blockchain and machine learning. We have understood the matter in this respect as follows: Data are the relevant factor for machine learning. We have shown the math behind deep learning. Whether difficult or simple to understand, the fact is that math is an open source to everyone. Thus, not primarily maths causes difficulties. However, the lack of data determines the result of artificial intelligence related projects. Exactly here we see the benefit of Blockchain[26].

In an overall view, we have no doubt that artificial intelligence will dominate our societies in the next decade. This applies in particular to companies and their existing business models. Even if the statistical figures put Germany in a rather remote place, we consider the potential for improvement to be high because of its intellectual resources. However, we see a mental transformation as a prerequisite for transforming the predominant risk perspective into a view of opportunities. Under the assumption of ideal social conditions, the existence of intellectual resources and the required technology, the chances for Germany are tremendously high.

In any case, all uncertainties about the theoretical foundations should be clear from now on. In an upcoming article, we will focus on practical use cases.[27]

[26] See Chapter 3.

[27] Next publication is already in preparation.

List of References

[1] „(ICLR), Dive Deep into Deep Learning: SAP at the International Conference on Learning Representations," [Online]. Available: https://blogs.sap.com/2017/05/22/dive-deep-into-deep-learning-sap-at-the-international-conference-on-learning-representations-iclr/.

[2] „Societyone," [Online]. Available: https://www.societyone.com.au/about-us.

[3] S. Russell, D. Dewey und M. Tegmark, „Research Priorities for Robust and Beneficial Artificial Intelligence," *2016arXiv160203506R,* 02 2016.

[4] A. Cuthbertson, „Elon Musk and Stephen Hawking Warn of Artificial Intelligence Arms Race," Newsweek, http://www.newsweek.com/ai-asilomar-principles-artificial-intelligence-elon-musk-550525, 2017.

[5] R. Cellan-Jones, „Stephen Hawking warns artificial intelligence could end mankind," BBC, Technology News, http://www.bbc.com/news/technology-30290540, 2014.

[6] „Statista, Branchenreport Deutschland," Statista GmbH , 2017.

[7] „Statista, Dossier, Deutsche Bank," Statista GmbH , 2017.

[8] „Statista, Top 100 Banking & Finance," Statista GmbH , 2017.

[9] „Statista, Digital Economy Compass," Statista GmbH , 2017.

[10] „Bundesverband Deutscher Kapitalbeteiligungsgesellschaften," [Online]. Available: http://www.bvkap.de/markt/statistiken.

[11] K. Enterprise, „Venture Pulse, Global Analysis of Venture Funding," Q2 2017.

[12] F. Sauvigny, „Analysis: Grundlagen, Differentiation, Integrationstheorie, Differentialgleichungen, Variationsmethoden," Springer-Verlag, 2013.

[13] T. Epelbaum, „Deep learning: Technical introduction," *2017arXiv170901412E,* 09 2017.

[14] D. Kriesel, Ein kleiner Überblick über Neuronale Netze, http://www.dkriesel.com/science/neural_networks: Kriesel2007NeuralNetworks, 2007.

[15] A. B. a. G. L. a. D. Barber, „Nesterov's Accelerated Gradient and Momentum as approximations to Regularised Update Descent," *2017 International Joint Conference on Neural Networks (IJCNN),* pp. 1899-1903, 2017.

[16] M. D. Zeiler, „ADADELTA: An Adaptive Learning Rate Method," *CoRR,* p. abs/1212.5701, 2012.

[17] S. M. Dewi Suryani, „Convolutional Neural Network," Binus university, school of computer science, http://socs.binus.ac.id/2017/02/27/convolutional-neural-network/, 2017.

[18] J. B. Diederik P. Kingma, „Adam: a Method for Stochastic Optimization.," in *International Conference on Learning Representations, 1–13*, 2015.

[19] T. Dozat, „Incorporating Nesterov Momentum into Adam," in *ICLR Workshop (1),* 2013-2016.

[20] J. R. Shewchuk, „An Introduction to the Conjugate Gradient Method Without the Agonizing Pain," Carnegie Mellon University, 1994.

[21] J. Zhou, „Digitalization and intelligentization of manufacturing industry," *Advances in Manufacturing*, pp. 1-7, Volume1, Issue 1 03 2013.

[22] A. T. Marco Gori, „On the problem of local minima in backpropagation,“ *IEEE Transactions on Pattern Analysis and Machine Intelligence (Volume: 14, Issue: 1,) ,* Jan 1992.