OULUN AMMATTIKORKEAKOULU

Seila Laakso

ARTIFICIAL NEURAL NETWORKS AND DEEP LEARNING

Possibilities and Limits of its Use in Modern Software Development

ARTIFICIAL NEURAL NETWORKS AND DEEP LEARNING

Possibilities and Limits of its Use in Modern Software Development

Seila Laakso
Autumn 2022
Bachelor’s thesis
Information Technology
Oulu University of Applied Sciences

TIIVISTELMÄ

Oulun ammattikorkeakoulu
Tieto- ja viestintätekniikka, Ohjelmistokehitys

Nimi: Seila Laakso
Otsikko: Keinotekoiset neuroverkot ja syväoppiminen,
Niiden mahdollisuudet ja rajoitukset tulevaisuuden sovelluskehityksessä
Ohjaaja: Jukka Jauhiainen
Työnvalmistumislukukausi ja -vuosi: Syksy 2022 Sivumäärä: 39

Tämä tutkimusartikkelimuotoinen päättötyö käsittelee keinotekoisten neuroverkkojen historiaa, rakennetta ja käyttötarkoituksia. Päättötyön alku käsittelee tekoälyä yleisesti: mm. sen eri tasoja sekä sen historiaa. Lisäksi osiossa kerrotaan älyn, erityisesti tekoälyn määritelmästä.

Päättötyön kohta kolme käsittelee keinotekoisia neuroverkkoja. Kohdassa esitellään neuroverkon rakenne hyvin yksityiskohtaisesti sekä käydään läpi sen toimintaperiaate. Kohdassa käsitellään myös neuroverkon eri tapoja oppia sekä käydään läpi yleisintä keino-neuroverkkoteknologiaan liittyvää termistöä.

Viimeinen osio käsittelee keinotekoisten neuroverkkojen ominaisuuksia. Sen rajoituksia sekä sen käytön hyötyjä ja mahdollisuuksia. Osiossa esitellään myös käytännön esimerkkejä keino-neuroverkkojen käytöstä eri teollisuuden osa-alueilla.

Aineistona tutkielmalle on käytetty suuri määrä tutkimusaineistoa eri aloilta sekä viitattu muutamaan tekoälyn opiskeluun tarkoitettuun laajaan oppikirjaan. Aineistossa on käytetty niin painettua kirjallisuutta kuin verkko aineistoa.

Päättötyön johtopäätelmänä voitaneen todeta neuroverkkoteknologian kehityksen olevan kiihtymässä lähitulevaisuudessa ja että uusia innovaatioita syntyy kaiken aikaa. Kehitystä vaativia ongelmia teknologiassa on sen arvaamattomuus sekä mustan laatikon tapainen toimintaperiaate, joka vaikeuttaa mahdollisten ongelmien diagnosointia ja korjaamista.

Liitteenä päättötyössä on loppuraportti tekoälyprojektista nimeltä ProGAN DaliA. Projekti toteutettiin osana tätä päättötyötä. ProGAN DaliA on progressiivinen generatiivinen kilpaileva neuroverkko, joka luo annetusta kuva-aineistosta uusia taideteoksia.

Asiasanat: neuroverkot, syväoppiminen, tekoäly, kuvantunnistus, AI, perseptroni

ABSTRACT

Oulu University of Applied Sciences
Degree Programme in Information Technology, Software Development

---

Author: Seila Laakso
Title of thesis: Artificial Neural Networks and Deep Learning,
Possibilities and Limits of its Use in Modern Software Development
Supervisors: Jukka Jauhiainen
Term and year when the thesis was submitted: Autumn 2022 Number of pages: 39

This thesis is written in scientific article form. It addresses artificial intelligence, artificial neural networks and deep learning. Thesis introduces artificial intelligence, what it is, its history and different levels of AI. Third section addresses more deeply one of the implementations of AI, artificial neural networks. How does artificial neural network function and how it is structured comprehensively. Some of the most often used terminology of the subject will also be covered.

The thesis also covers the practical applications of artificial neural network technology and how it is used in different fields of industry. Positive and negative properties of artificial neural networks and how they should be developed in the future will be observed.

For this thesis a lot of research material from different research areas was used. A few AI and neural network related textbooks were also referred. The research material was both in electronic and printed form.

As a conclusion, it may be stated that the development of neural network technologies will be accelerating in the near future and that new innovations constantly arise. Biggest obstacles of developing of this technology are its unpredictability and its Blackbox like operation principle which makes problem diagnostics and solving difficult.

As an annex in this thesis is a final report of artificial intelligence project called ProGAN DaliA. The project was carried out as part of this thesis. ProGAN DaliA is a progressive generative adversarial neural network, which creates new artwork from a dataset of art pieces.

Keywords: Artificial neural networks, artificial intelligence, AI, deep learning, perceptron

PREFACE

I want to thank all that have contributed to this thesis by helping and advising me during the work. I feel that I couldn't have achieved so much without your help. Nothing is impossible if you have trusty people to hearten for you!

I want to thank Jukka Jauhiainen for directing this work, Michael Collins and William O'Mahony for my Irish exchange supervising and thesis review, English language supervisor Kaija Posio and Susanna Kujanpää for helping with the study planning.

I also want to thank Lea Salonen for mental support and my family for understanding the unexplainable, Cristina Pizziolo for inspiring support and lastly but most importantly my fiancé PhD. Jussi Putaala for the endless love and empathy. You are always there to catch me if I fall.

CONTENTS

VOCABULARY

AGI	Artificial General Intelligence
AI	Artificial Intelligence
ANI	Artificial Narrow Intelligence
ANN	Artificial Neural Network
ASI	Artificial Super Intelligence
BCE	Before Common Era
FFMLP	Feed Forward Multilayer Perceptron
GAN	Generative Adversarial Network
GELU	Gaussian Error Linear Unit
IQ	Intelligence Quotient
LSMT	Long Short-term Memory
ReLU	Rectified Linear Unit
RL	Reinforced Learning
RNN	Recurrent Neural Network
SL	Supervised Learning
SME	Squared Mean Error
UL	Unsupervised Learning

1 INTRODUCTION

"We can only see a short distance ahead, but we can see plenty there that needs to be done."
-Alan Turing [1]

Deep learning and artificial neural networks can be heard and read in the news all the time, but what do they actually mean. What is this "new" technology and why is it getting all this attention? In this thesis I will lead you to the realm of artificial intelligence and introduce you to artificial neural networks. What are artificial neural networks? Are AIs really that intelligent? And what does a coffee cup have to do with all of this?

This thesis will view AI and ANN from the practical point of view, not concentrating too much on the complex mathematical world of ANNs. We will, however, take a glimpse on how a multilayer perceptron works both figuratively and mathematically. I hope that after viewing this thesis you will understand what this media fuss is about and what we can expect to see in the future of ANN software development.

2 ARTIFICIAL INTELLIGENCE

2.1 What is Artificial Intelligence?

Artificial intelligence (or AI in short) does not have a universally agreed definition but generally the most agreed upon definitions do include the thought of humanlike intelligence in machines. AI is defined by the European union as an “ability of a machine to display human-like capabilities such as reasoning, learning, planning and creativity.” [2]. This does not mean that AI should be like human; process information like human or perform actions like human. Neither does this mean that scientists and engineers would try to build an artificial human brain but very different thing all together.

To understand the concept of neural networks and deep learning, understanding of the fundamentals of artificial intelligence is needed. But before anything, a word about intelligence. The first thing that may come into mind about intelligence is an ability to solve problems or simply have a high IQ. But when we talk about intelligent machines, we must view this matter in different perspective. Intelligence is a broad term and can be seen differently depending on your culture, education and your IQ. [3]. Therefore, because of the nature of the term, a directional definition of intelligence is needed. Using Artificial Intelligence textbook [4] as the guideline, intelligent abilities can be determined in four different ways.

- Linguistic and social intelligence: The ability to understand written and/or spoken language and the ability to generate such in a way understandable to receiver (usually human). For example, good grammar and ability to understand different spoken dialects of the same language. In more advanced programs this section includes detecting, understanding, foreseeing and adjusting to human behaviour, both in groups and as individual. For example: learning traffic patterns during rush-hours.
- Signal processing: The ability to filter and check received data for inconsistencies and to be able to detect obvious errors. Also, the ability to separate correct data from an error and ability patch and work around found errors. Storing and retrieving data and enabling the efficient use of this data for reasoning.

- Reasoning: Using the stored and received data for analysing, problem solving and making conclusions. Calculating possibilities and taking actions accordingly. Ability to adjust if failing in the given task. Also adjusting to the receiver by making the outcome comprehensible and everyway acceptable. This may mean simplifying, embellishing and shortening the outcome. For example, a chat bot for writing too much too quickly for the receiver to miss messages or by making the text grammatically correct, but too complicated for the receiver to understand.
- Learning: ability to learn from the data both received and reasoned to optimise the outcome. Learning can be done in three different ways which are covered in more depth later, in section 3.2 Deep learning.

Now by predetermining artificial intelligence it is easier to understand the premise of this topic. People and machine work in different ways even though the used terms here are the same. So, by using terms like learning or reasoning computer and human actions cannot be compared to one another.

2.2 History of AI

This section will briefly introduce the history of AI from the ancient times all the way to today. This is to get a little understanding of how the idea and technology of AI has developed over the years.

2.2.1 First Steps Towards Technology

The thought about artificial intelligence has existed since the classical antiquity. Homer, the presumed author of Iliad and the Odyssey, wrote around 850 BCE about living and speaking maidens made from gold. These android-like creatures were called *kourai khryseai* and were crafted by Hephaestus, a divine smith and the god of fire, metalworking, and technology in the Greek mythology. Similar legends can be found in early Chinese and Northern Indian writings, but what connects these legends together is the way how these androids were always told to be obedient and intelligent. A perfect laborers so to speak. [5] [6]

Ever since these writings, humans have kept on dreaming about building an artificial human, an android, with an artificial mind and artificial intelligence. AI has stayed as frequent topic for science fiction writers to this day but when it comes to actual innovations before the creation modern computers, the development of AI was closely linked to creation of engineering and technology overall. As, to create any form of intelligent item or being, a way to store and utilise information is needed. Salamis tablet was a white marble slab, used to aid Babylonians in calculus circa 300 BCE. It was the precursor of abacuses that were widely used eight centuries later in Europe and Asia. [7].The first automatons to keep track of information were the water clocks build around the 16th century BCE in Egypt and Babylon. The earliest water clocks were just bowl with a hole in the bottoms from where the water slowly flows to another bowl. This way early Egyptians could keep track of time and when this technology developed, advanced clocks with automated indicators were built. [6]

Innovations around this area of science progressed slowly for a long time and even though static electricity was known as a phenomenon since as early as the 6th century BCE, electricity was not discovered until the early 18th century. One of the most well-known scientists to dwell on this idea was Benjamin Franklin. Even if he only researched electricity as a phenomenon for 7 years, he left his mark to history books by proving that electricity not only occurs in his lab experiments but as a natural phenomenon as lighting. He used a kite and a hanging metal key to catch a lightning bolt thus proving that it is an electrical discharge. Alongside of his experiments he invented a lightning conductor, thus saving a lot of lives. [8]

For a such complex concept as AI, it is no wonder that many innovations and scientists have contributed to its development in the more recent future. Norbert Wiener introduced the cybernetics in 1948. It is a study of control and communications in machines and other organisms, such as humans. [9]. It is also closely linked to information theory and communication theory produced by Harry Nyquist, Ralph Hartley and Claude Shannon in 1920-1950. It provided mathematical tools and general understanding of information transferring to AI scientists. [10]. Also, studies on philosophy, logic, statistics, linguistics and mathematics have had a huge impact on the development of AI. [11]. When it comes to development of AI, most of these studies would have been useless without development of computers. In fact, the term ‘computer’ used to mean a person who did calculations, computed. And the term was associated with this activity until the first mechanical computers emerged around the 1950’s. [12]

2.2.2 The Rise of Computers

One of the early scientists to explore the thought of artificial intelligence was Alan Turing. Turing's machine was a mathematical model that resembled an early computer. And even though it does not look anything like the modern computers, it was done before any actual programmable computers were built in 1936. [13]. Besides his well-known mathematical model, Turing also started to wonder if computers could become intelligent. He wrote an article to Mind Magazine titled "Can a machine think?". Well after his unfortunate death in 1953, this model called an imitation game was renamed in his honour to the Turing test. Turing's article was a major step in the development of AI as it crystallised the possibility of artificial intelligence. [11]

The term Artificial Intelligence was invented by John McCarthy who was one of the attendees in a 1956 conference held at Dartmouth College in New Hampshire, United States. Called as The Dartmouth Summer Research Project on Artificial Intelligence (later referred as the Dartmouth workshop), it was held to gather up the most brilliant computer engineering minds of the time and brainstorm about Artificial Intelligence. What was achieved in the workshop was consensus of forming a new research field for AI. [14]. In an article published 2006 in AI magazine James Moor described well what happened in the Dartmouth workshop: "There was no agreement on a general theory of the field and in particular on a general theory of learning. The field of AI was launched not by agreement on methodology or choice of problems or general theory, but by the shared vision that computers can be made to perform intelligent tasks." [15]

After Dartmouth workshop, AI development has had its winters, springs and summers after another. Some of the winters were caused by lack of research funds and springs vice versa. It can be assumed that public opinion on Artificial Intelligence has been fuelling those events and that science fiction AI characters may influence the way public perceive AI. [16]. Today is considered a new springtime of AI development. Many new AI companies are emerging on the market, such as Duolingo [17], Utopia Analytics [18] and Valossa Labs Oy [19] (just to name few) and older tech-companies are starting to recognise AI as a way to improve safety, efficiency and profit.

In more recent times AI development has taken huge leaps due to the fact that the development is no longer bound to the development of computers' processing speed or memory usage. But it was not until 1997 that a machine could defeat the world chess champion Gary Kasparow. The program was named Deep Blue and was developed by IBM. [11]. Other mentionable AI milestones to mention could be Spirit and Opportunity which are NASA's autonomous Mars rovers [20], Google's first autonomous car in 2009 [21] and smartphone assistant apps, such as Apple's Siri, Google Now and Microsoft's Cortana [22]

2.3 Different Levels of AI

Artificial intelligence can be divided in three different levels: Artificial Narrow Intelligence (ANI), Artificial General Intelligence (AGI) and Artificial Super Intelligence (ASI).
So far only ANI has been successfully developed in many different forms, but scientists are taking huge steps towards developing artificial general intelligence. [23]

2.3.1 Artificial Narrow Intelligence

If conversed about AI in action and outside of the scope of science fiction, it is usually referred to artificial narrow intelligence or ANI. ANI is a form of AI that is designed to perform singular and highly concentrated tasks, for example, playing chess or do image recognition [23]. Even the most sophisticated AI today is considered ANI. Voice recognition software, such as Apple Siri or Microsoft Cortana, while very human like in certain situations, these programs consist of are just many pieces of ANI-software brought together to perform more complicated tasks [22].

2.3.2 Artificial General and Super Intelligence

If referred to more complex AI systems, an AGI or artificial general intelligence need to be mentioned. AGI would be a machine that would not only be capable of fully interact and cope with many different environments but could also understand and solve complex assignments. Shortly, what separates ANI and AGI from each other is problem solving. Where ANI can already beat the best human chess or Go (game) player [24], a single AGI would do this and complete any other given task better than any human would. Therefore, calling AGI a human like intelligence is problematic as AGI would be in fact superior to human intelligence. If taken this thought even further, when or if AGI would ever gain consciousness and became self-aware, it could self-improve to a point where it interprets and understands human behaviour but would also surpass our capability to understand it. This machine would then be called ASI or Artificial Super Intelligence. [25]

The thought of creating general or super artificial intelligence has been pretty much abandoned nowadays because of its sheer complexity and lack of financial support. And therefore, engineers are concentrating on a more subtle “divide and conquer” way of thinking of AI. More complex and intelligent structures that exceed the capability of singular ANI can be created by uniting multiple ANI applications to one software. [26]

3 NEURAL NETWORKS

One of the earliest biology inspired approach to AI is an artificial neural network or ANN. The basic vanilla ANN is a very crude, simplified model of brain neural structure and it can be used to various types of AI software structures. The structure presented here is a feedforward multilayer perceptron (FFMLP) and it has been used as a foundation to many architectural variations after its invention. [4, pp. 727-729]

3.1 Artificial Neural Network Structure

Every neural network consists of units called artificial neurons, also called nodes. These artificial neurons consist of components that represent typical human nerve cell. Parts of the node have been named after the biological counterpart and their purposes have a lot of similarities (see Figure 1.) In ANN each of these parts have their designated tasks and mathematical functions. Dendrites are used for accepting inputs. A cell body, also called Soma, processes the input through activation functions. Axon transforms the processed input into outputs and delivers it (through axonal arborization) to synapses which are the connection between neurons. [27]

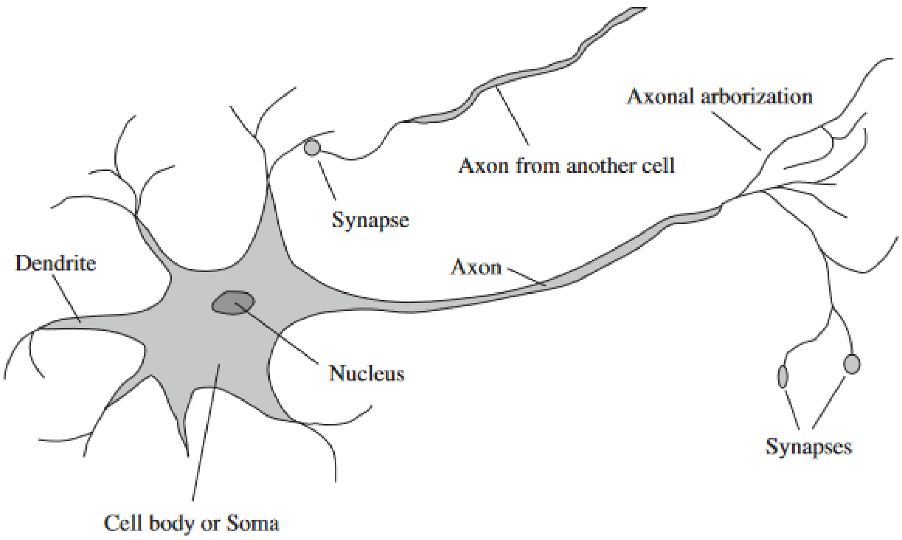


FIGURE 1. Composition of human neuron [4, p. 11].

These neurons are the basic building blocks of an ANN that is one of the ways to build AI. By combining these neurons into multiple layers, an Artificial Neural Network is created. ANN can be divided into three separate layers: input layer, output layer and hidden layers. ANN with only the input layer connected directly to the output layer is called a single layer perceptron where ANN with at least one hidden layer is called a multilayer perceptron. These hidden layers form the so-called depth in deep learning and are presented visually in the figure below (figure 2). Each of the circles in figure 2 represents a single artificial neuron (presented in figure 1) which processes the data according to its predetermined adjustment (see chaper 3.3 Learning Algorithms).

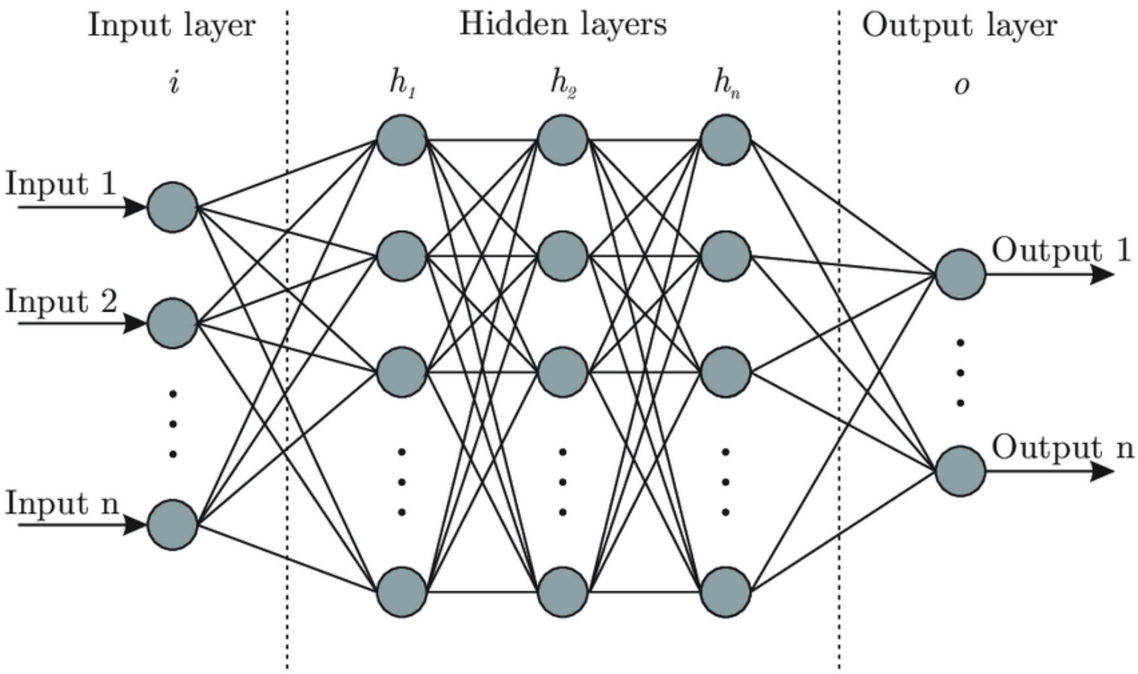


FIGURE 2. Architecture of Feedforward Multilayer Perceptron [28].

Depending on the project sizes, these layers can be adjusted. The amount of inputs in the input layer can be adjusted depending on the used data. For example every pixel of a picture can be used as an input in image recognition. The amount of outputs is determed by the wanted outcome. For example: Does the picture represent a cat, dog or neither would be 3 outputs. Is this spoken word any of these 10 commands? would make 10 or 11 outputs depending on the system. [29]

The amount of hidden layers and how many nodes are in each hidden layer is a bit trickier to determine. A rule of thumb is to have the equal ammount or less nodes than in the input layer but more than what is in the output layer. Usually by testing few different settings and picking the best outcome is the way to go. But it has to be kept in mind that the more you have hidden layers and nodes the more challenged tasks the ANN can solve, but this also means that more computing power is required to perform the task. [30]

The figure 2 above represents a multilayer perceptron feedforward neural network. By combining several layers and backfeeding the output data, a number of different neural network architectures can be created. In a feedforward neural network information moves only to one direction where in backpropagation network information moves in both ways creating a feedback circle enabling for the network to learn on its own.

3.2 Deep Learning

A crucial part of creating intelligence is learning. There are many different algorithms to train a neural network. For example, the Levenberg-Marquardt algorithm [31], the Newton Method [32] and a gradient descent which will be examined later in section 3.2.2 The Cost Function. New algorithms are constantly being developed and combinations of algorithms are tested, such as Gram-Gauss-Newton method, which combines the Gram matrix, the Gaussian process, and the Newton method [33].

Deep learning can be casted into three different possibilities based on the feedback method of the system: unsupervised, reinforced and supervised learning [4, pp. 693-695]. Each of these will be briefly introduced next.

3.2.1 Unsupervised Learning

Like name suggests, in unsupervised learning (or UL for short) there is no feedback supply. Therefore, for the system there is no knowledge of right or wrong answer, but the system can still learn possibly useful data patterns by clustering. Clustering means that the system tries to find similarities in the given data and groups it accordingly. This is not to be mixed with classifying as there is no labels in the data. [4, pp. 694-696]. Different methods can be used in clustering. k-Means represents the mean (arithmetic average) of each cluster (an example is represented in figure 3) and k-Medians use medians instead of means. Also, density and grid-based methods can be used to cluster data. [34]

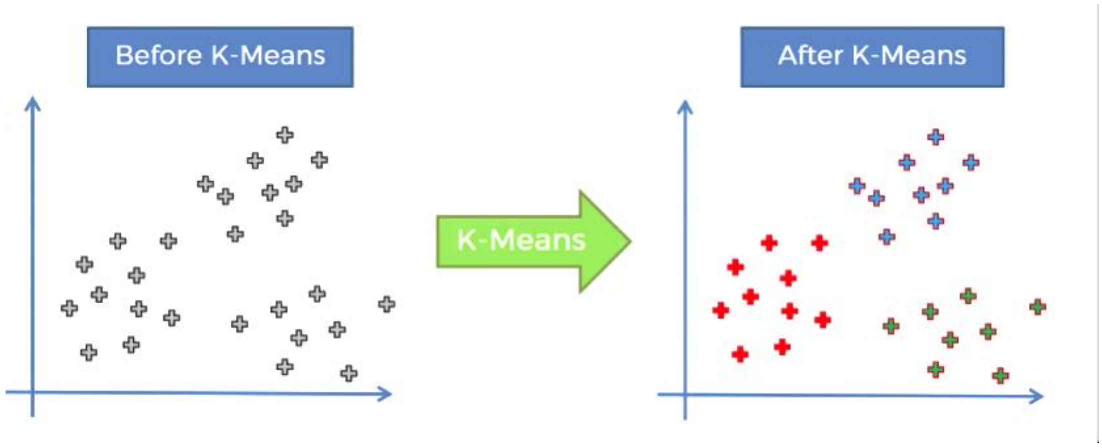


FIGURE 3. Representation of K-means clustering [35].

Because labelling data is time consuming and therefore expensive, clustering is an excellent tool to find trends from the big data. Another commonly used unsupervised learning technology is autoencoders. This system can be as simple as a single perceptron which simplifies and reconstructs, for example, a given handwriting number image seen in figure 4. Autoencoders can be used to eliminate insignificant data, so called “noise” from data. [36]

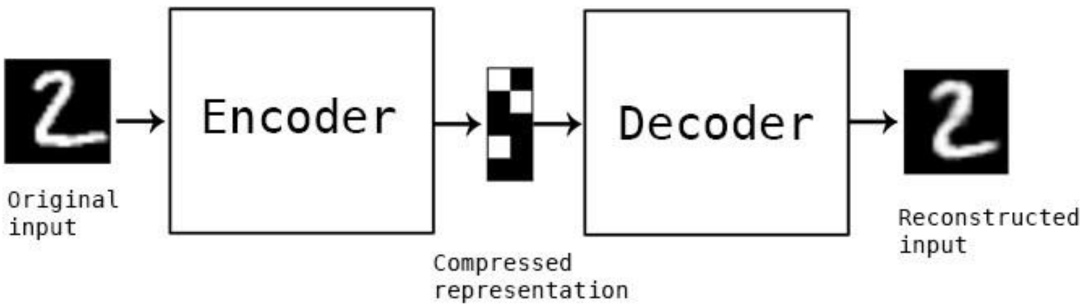


FIGURE 4. Architecture of a simple autoencoder which consist of three parts. Encoder, bottleneck, and decoder. Difference between output and original input can be measured and is called reconstruction loss. [36]

3.2.2 Reinforced Learning

In reinforced learning (RL) feedback is given in reinforcements, as punishments and rewards. This observation can be done automatically, for example scoring the performance after every attempt. The system will learn that by doing the task in a certain way is more rewarding than the others.

This way by repeating the process multiple times the system will learn the most efficient way of doing the given task in this environment without the teacher telling exactly what to do.

One of the most used approaches in RL are State-action-reward-state-action or SARSA-algorithm which gives the system a probability for an action to result positive feedback, a so-called policy, and Q-learning which gives the system no hints about how to achieve positive feedback.

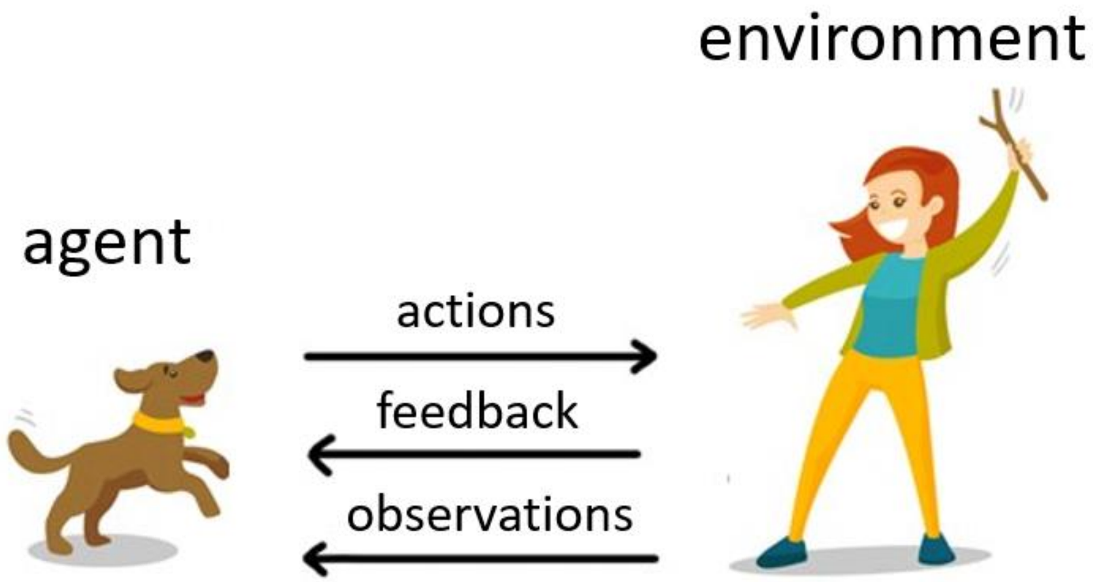


FIGURE 5. Reinforced learning examples can also be found in real life situations for example teaching a dog how to play fetch. [37]

3.2.3 Supervised Learning

Classification, which was mentioned earlier, is considered supervised learning as labels is used for a system to tell the apples from oranges. By giving the system simple “right” or “wrong” feedback when finishing the task is what SL is in its simplest form, binary classification. The idea of SL is for the system to learn the correct response for the given data but also hopefully making it provide the correct answer from the nontrained data. This can be expected because of the reinforced pathways that systems have learned from the teaching data. [38]. In a case where the system needs to give user a number estimate instead of label, regression can be used. If regression provides a continuous numerical estimate value, it is called linear regression. As a great example to tell classification from regression would be a weather machine. Using classification, a system would provide information whether the weather is sunny or cloudy, using a regression system gives a temperature estimate using previous data from similar weeks in the same month from years before. [4, p. 696]

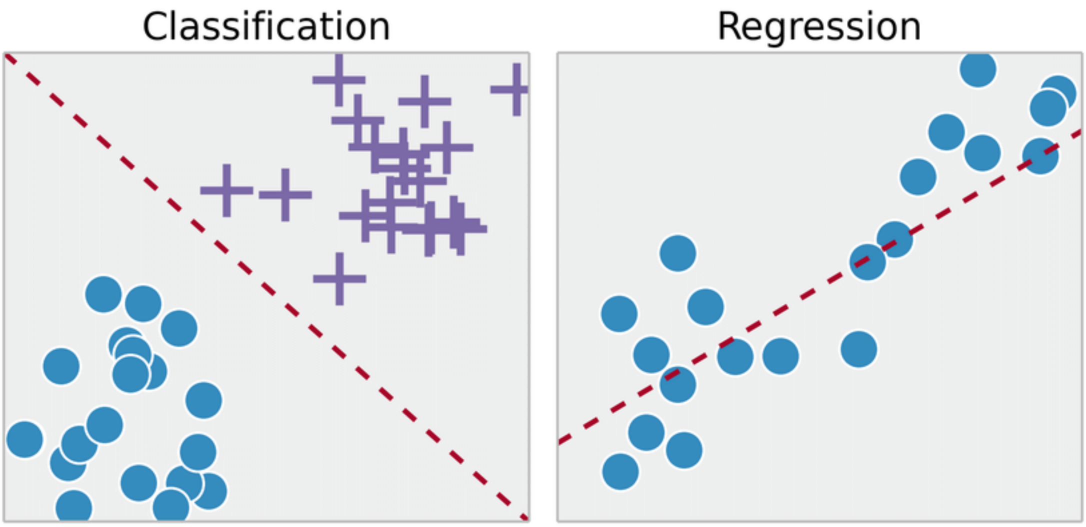


FIGURE 6. Visualisation of the difference between classification and regression. [39]

Usually, the answers to whether the completed task was right or wrong is not binary. In semi-supervised learning the system must make what it can from the collection of data where even the labels or correct answers are not always to be trusted. For example, data sets of pictures of persons with their ages and the system task is to guess the persons' age form the picture. There may be cases where a person has lied their age in the training data set, therefore the task is semi-supervised.

3.3 Learning Algorithms

The wisdom in the neural networks lies in the mathematics of the structure. As the information moves through the network from one layer to another, it activates the nodes to a certain level. This activation can represent, for example in picture recognition, the input layer grayscale value of the pixel from 0 to 1. Zero being a black pixel and one being a pure white pixel. This process in the node is called activation. In the hidden layer, as each of the nodes are connected differently by strengthening and weakening different connections, the ANN is able to learn from the training data. This strength difference is called the weight. Each of the nodes in the hidden and output layers can be understood as the sum of the activation and the weights of the previous layer. To make the ANN more efficient on learning and amplifying the weights and activations a threshold function, such as sigmoid or ReLU (Rectified linear unit) can be added to the function. Both are presented in figure 7. If up or down scaling of the value is needed, a number can be added before calculating the threshold function. This is called a bias. [29]

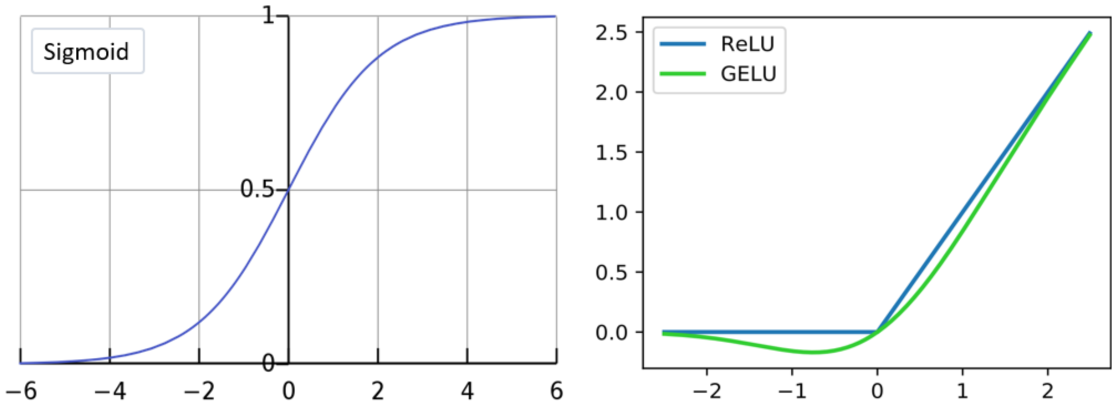


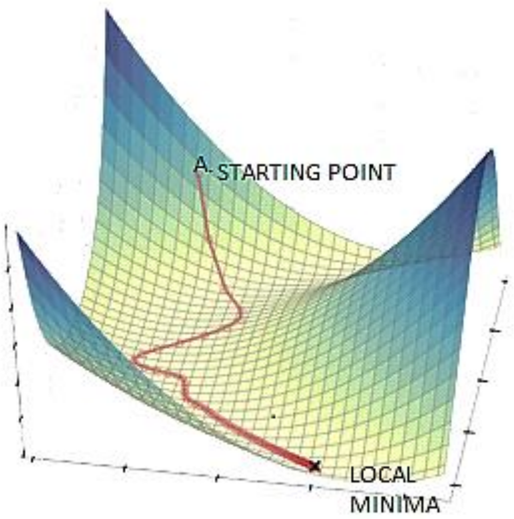
FIGURE 7. Visual presentation of classic threshold functions. Left one represents sigmoid [40] and on the right Rectified Linear Unit (ReLU) and Gaussian Error Linear Unit (GELU.) [41].

3.3.1 Fuzzy Logic

As mentioned in the previous section, not all answers in the world are binary. For this problem in Lotfi Zadeh, an Azerbaijani scientist developed what is called fuzzy logic. Instead of asking “Is there coffee in my mug?” the question can be formulated as “How much coffee is in my mug?”. By thinking logic problems this way, the certainty or uncertainty of the AI’s answer can be projected. Fuzzy logic is a very helpful tool for AI developers when simple binary answers are not available. And fuzzy logic works in both ways. Fuzzifying (fuzzification) means converting binary data (also known as a crisp value) to a scale between zero and one or even verbal answers like “a lot”, “some” or “little”. And defuzzifying (defuzzification) converting data vice versa. [42]

3.3.2 The Cost Function

To teach anything, and in this case a neural network, a way to measure the performance of the “student” is needed. In ANN training, a cost function can be measured. The Cost function is the sum of the difference of the system output and the wanted output. So, in this approach of teaching the ANN, the smaller the cost, the closer is the wanted output result. And like all learning, this goes through trial and error by gradually minimising the cost after multiple circles of training. Now, usually the data is not simple black and white, therefore by finding the so-called local minima is the most efficient way to go. To calculate this, either batch or stochastic gradient descent can be used. Where the batch gradient descent takes a step towards the minimum after each training cycle, the stochastic gradient descent takes a multiple samples of training data and uses the average of each patch to leap towards the local minimum hence speeding up the gradient process. But by using the stochastic method finding the local minimum is not certain as it can oscillate around the minimum without finding the lowest point. [4, pp. 719-720] [43]



FIGURES 8 Visualisation of the gradient descent. A is the starting point of the training. In this example local minima is also the global minima. [44]

Besides the gradient descent, there are other ways of minimising the cost. The Mean Squared Error uses error values of datapoint to minimise to overall cost by calculating the squared mean of the error (by this way eliminating negative numbers) between the predicted and actual datapoints. A regression line can be drawn closest to each datapoint after several training cycles.

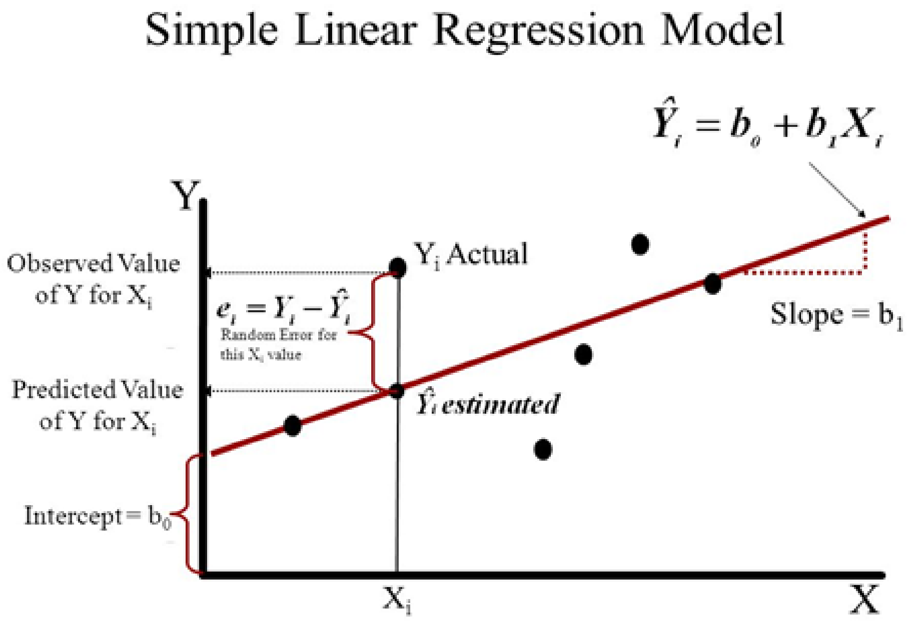


FIGURE 9. Model of linear regression. Red line representing the regression prediction line. Each dot represents an actual training datapoint from which the SME could be calculated. [45]

3.3.3 Backpropagation

Thinking about the number of connections between the input layer, hidden layers and the output layer, changing the weights and biases of every connection to every neuron would be utterly impossible by hand. By creating a feedback loop or backpropagating, these values can be automatically changed to closer approximate of the wanted value, minimising the cost. What this means in action is that after every training data circle, the ANN adds the value adjustment to the connections of each hidden layer connection iterating backwards one layer at the time. In this way after multiple training circles, the network should be able to navigate through data that it has never seen before. Therefore, no hand in hand training is required and the network can adjust the values automatically by moving backwards in the network. [4, pp. 733-736]

3.4 Adaptations of Neural Network

This neural net prescribed before is not the only adaptation of this technology. By combining layers, the quantity of nodes, adding memory elements and varying the calculus in the connections, a number of different network architectures can be created. Each of which are useful for completing different kinds of tasks. This next list introduces some ANN adaptations which most closely resemble the previously shown FFMLP.

Before this multilayer perceptron was introduced in 1969, a single layer perceptron was created. It only consisted of input nodes that were directly connected to output nodes. It was the first architecture that was considered a neural network as it contained more than just one neuron. [4, p. 22]

The Recurrent neural network (RNN) is on the other hand a bit more advanced version of the FFMLP as it backpropagates [4, p. 729].The RN Network with a bigger memory capacity is called The Long short-term memory (LSTM) [46].

A Convolutional neural network has much in common with the FFMLP but it has been optimised for image and video recognition by adding convolutional layers, pooling layers, normalisation layers and fully connected layers between the input and output layers. Word convolution refers to a mathematical operation between two functions that produce a third function. Convolutional layers significantly reduce the processing power that would be needed to calculate all those accumulating weights and biases of the network. [47]

The Generative adversarial network (GAN) architecture is two networks competing. The other one tries to create authentic datapoints while the other finds and tries to recognise between real and generated datapoints, thus the name. This architecture is able to create very realistic looking photographs and deepfake-videos. [48]

4 PROPERTIES OF NEURAL NETWORK TECHNOLOGY

4.1 Why are Artificial Neural Networks not an answer to everything?

There are many possibilities how ANNs can be used but there are also problems that show that ANNs are not the optimal solution.

4.1.1 Blackbox

As seen in the ANN structure section before, the ANN architecture resembles a lot like a black box. There is usually no way of predicting the output quality of the experiment. Even if the structure would be examined node by node, there is no certain knowledge of how the ANN is actually solving the problems. Much like our own neural network. That is why there is also no way of diagnosing where the problem is if something is wrong with the output. Therefore, building an ANN relies a lot on trial and error and that is why training an accurate neural network is very time consuming.

While an optimised ANN can reach a very high classification accuracy, it is still hard to consider it reliable for its black box nature. For example, it can be imagined a doctor explaining to a patient that a computer said that the patient has cancer. “We don’t know how computer came to this conclusion, but we are 98,7% sure that you have one, somewhere in this x-ray.”

4.1.2 Experimental Nature

If you are looking for an easy to fix, reliable, predictable and fast solution for your software problem, ANN is surely not the first thing on the list. Even though AI and ANN have been already developed for more than 60 years [4, p. 22], most of the projects can be portrayed as experimental because of the Blackbox nature explained above. There is yet no way of standardising a paradigm that works for every or even some of the problems ANN needs to solve. What works with other problems usually do not work with another and every time ANN must be trained with plenty a of data to be accurate.

4.1.3 Hardware Consuming

Because of the parallel processing work that happens in ANN while solving anything more complex, it requires a lot of processing power. [49]

4.1.4 Finding the Perfect Fit

Overfitting is a phenomenon that happens when ANN is too well trained (see figure 10). ANN starts to imitate the random oscillation of the training data thus creating a "background noise". By teaching the ANN too little it fails to deliver the wanted outcome more often. Both cases cause inferior performance on new data than an adjusted learning curve. Finding the balance between over and underfitting is found through trial and error and is so for every new ANN project. [50]

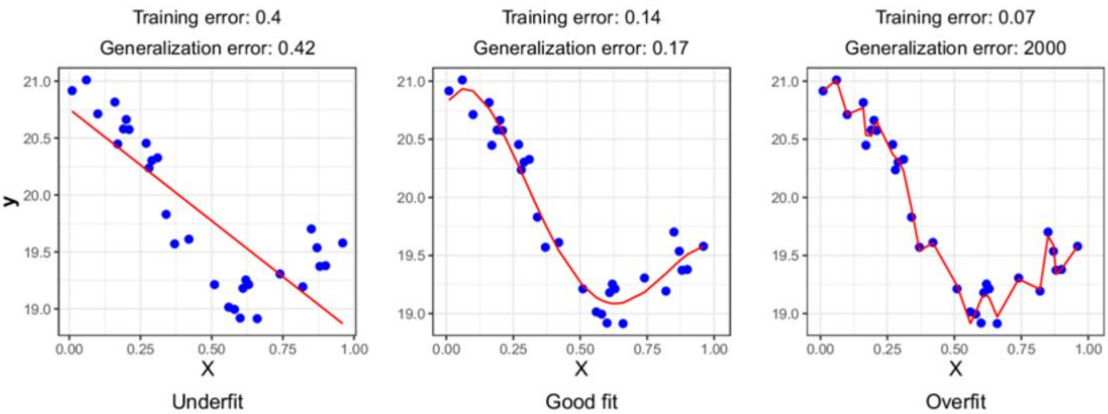


FIGURE 10. Representation of under- and overfitted learning curve. Good fit represents an adjusted learning curve (in the middle) [50].

4.2 Advantages of Artificial Neural Networks

4.2.1 Ability to Overcome

Compared to traditional software, where all possible errors and shortcomings usually need to be considered beforehand to avoid them, ANN can withstand and overcome obstacles that were unexpected. This is one of the biggest advantages of ANN. ANN works with data that is new, random and fuzzy. Therefore, ANN has an advantage over real life applications where not all situations can be foreseen. [51]

4.2.2 Lessens Human Error

In applications where the ANN technology is used as a tool a beside human expert, for example a doctor, laboratory technician or even just a casual car driver, using ANN as a so called third eye can significantly lower the chance of human error. For example, ANN can find in an x ray image a tumour that the doctor missed. Or ANN saw and identified the ponytail of that little girl running in the parking lot before the car driver and possibly prevented an accident from happening. And it must also be taken into account the fact that a computer is much faster to make decisions than humans. [52]

4.2.3 Distributed Memory and Error Tolerance

When traditional software solutions and ANNs are compared, one obvious difference is the memory usage. Where traditional software stores data into databases, ANNs knowledge is stored into the connections between nodes as weights and biases. Because of this nature of ANNs distributed knowledge, it is not as vulnerable to a memory loss. Even losing a node or two does not a cause significant damage to the network. This is not the case in traditional software as loosing part of the software may cause a serious malfunction or even paralyze the whole system. This is not only useful in cases where the actual hardware must perform in hazardous environments but it also works as a safety feature because no actual visible data is stored into the system, just numbers and equations that are no way usable for criminals. [51] [53]

4.3 Real-life Applications

As widely known, AI applications are already around all of us and they are becoming more and more part of our daily lives. Some of the most interesting and recently advanced approaches are introduced here.

4.3.1 Datamining

From a vast amount of data that is produced everyday it is getting harder and harder to find the essential data. Finding the relevant data from the so-called big data is called datamining and ANN is a useful tool for this purpose as it can cluster, classify and filter the data for any purpose needed.

Industries using this kind of technology would be classifying measurements for industrial companies or for any company that wants to know more about their customers preferences. [54]

A good example of ANN use in datamining on the financial field is a loan applications risk analysing system, which rates customers' loan applications between good and bad investments. Each characteristic of the customer is valued by a number, as ANN can only use numerical values in computing. After training of the ANN, software should be able to predict customers' credit rating. See an example of the ANN architecture of this application in figure 11. [55, pp. 15-18]

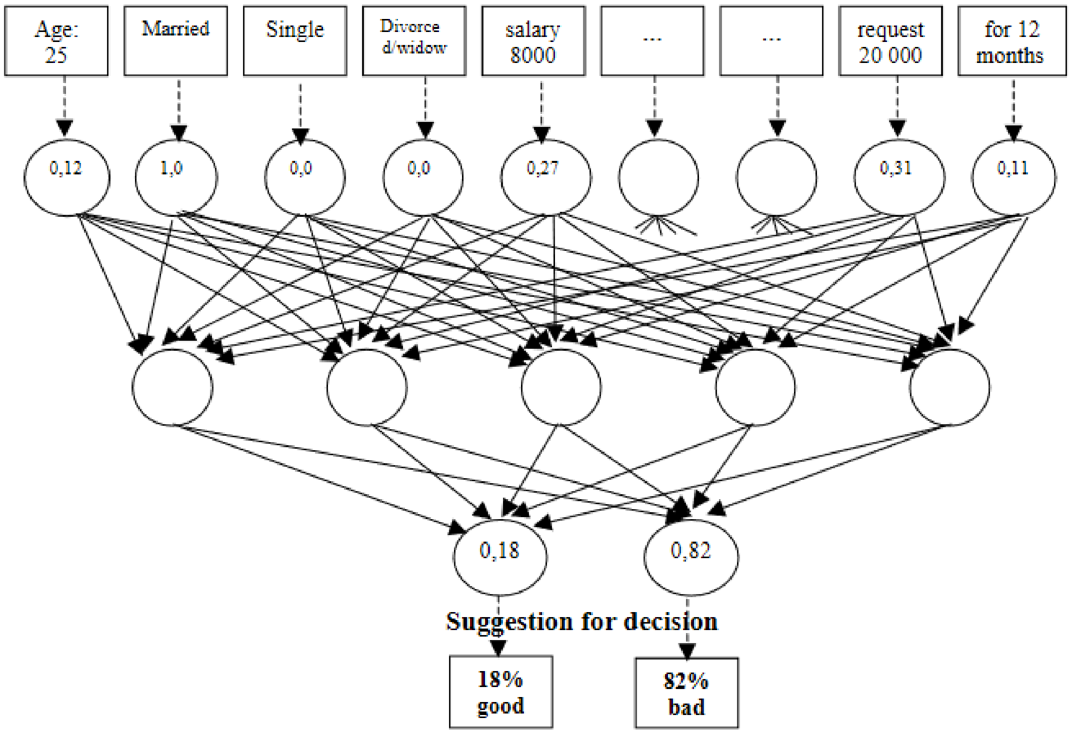


FIGURE 11. A loan application's risk analysing system architecture. Information input flows from the top to bottom. A multilayer perceptron is used for this application. [55, p. 18]

4.3.2 Image and Sound Recognition

As used as an example earlier, image recognition is a very suitable task for ANN. Sophisticated ANN architectures can not only recognise objects from images, recolour black and white images, automatically sharpen grainy images but also create images and videos that are incredibly realistic, so called deep fake images. ANN can also be used to detect and decompose deep fake images and videos. In figure 12 is a set of ANN generated celebrities. People in the pictures are not real people but fabricated using the most sophisticated GAN technology. Image recognition features are also useful in computer vision projects, such as self-driving cars. [56] [57]

FIGURE 12. Generated images of non-existent celebrities. Images were generated by generative adversarial network (GAN). This image generation experiment was done by Nvidia researchers in 2018. [58]

Deep learning algorithms are also used in speech recognition applications, such as Apple Siri, Microsoft Cortana and Amazon Alexa [48].

4.3.3 Forecasting and Simulation

The ability of neural network to forecast from previous data is a very useful ability and it has been used successfully in many different fields of technology from material engineering [59], to environmental sciences [60] and medicine [57]. As an example of ANN used in simulation, in material engineering, a trained ANN can be used to predict material properties, such as hardness very accurately. Without ANN, accurate load-displacement curves are difficult to create because of the geometric and material nonuniformity and contact interactions in the testing situation. [61]

Another example of practical ANN applications is electrical load forecasting systems. The ability of ANN to use nonlinear data for forecasting has been proved to be more accurate than traditional statistical methods in forecasting electricity usage. Scientists tested several different deep learning architectures of ANN and trained it with real load datasets from Germany, America and Australia. [62]

5 SUMMARY AND CONCLUSIONS

Artificial Neural Networks have been gaining more and more interest both in science community (and not only computer science) and industrial sector and it can be clearly seen why. It has taken many decades of baby steps and advance in computing power for communities to starting to realise what AI can achieve and the possibilities of the uses in the future. And it is easy to believe that this is only the start of the development.

Even if ANN does not understand the data it processes, as it can only find similarities and compute figures and even though the development of AGI is still far ahead, ANNs can already be applied as a tool in so many different applications, in so many different fields that it is certain that ANN's will become even more common in the future. Because of their accuracy, ANNs are suitable for applications that operate in crucial circumstances such as aviation industry [63], military technology [64] and healthcare [65].

By combining ANNs with other software technologies accuracy, speed and detail of computing never seen before can be reached. Developing a financially efficient ANN will still take some time. And for everyday applications, it might not be the best solution. But where you need big data filtering, accurate predictions or just something that is really good at "Where's Wally?", ANNs are something to consider as an option [66].

The development of ANN will trend towards deep neural networks in the future years because the complexity of tasks they are needed to solve will increase over time. Therefore, single layer perceptrons and small multilayer perceptrons will be the thing of the past. And as computer processing power increases further, even deeper ANNs will be developed to take on the challenge. Also, different architectural solutions, e.g. spiked ANNs, are being tested [67].

Major obstacles to solve in the future ANN development will be technique standardising making ANN development more of a routine than a guessing game when it comes to layer structures and depth. There is also work to be done in the predictability of the system accuracy and output. That is why solving the Blackbox structure is a key factor for the future AI development. Not only for the developers to be able to fix, copy and adjust systems better, but for customers and end users as transparency of computing and use of data is important for them, too.

6 DISCUSSION

My aims for this thesis were to learn about AI, Artificial Neural Networks and deep learning and learn what future software development options ANNs and deep learning have. During making of this thesis, I started to feel that I have taken a bit too much to chew because the options where ANNs can be used seem to be unlimited. What comes to the learning part of AI and ANN, I feel I have gained a huge amount of knowledge about these subjects. So much so that I feel that this thesis does not give the full credit of the lengths of learning that I have done in these past few months.

I was fascinated about this subject from the start. The thought about thinking computers felt so unreal. But the more I learned, the more I understood that the truth was far much more fascinating. The fact that ANNs develop to function more and more like a real brain and that we can see to the future with forecasting ANNs is quite amazing.

I have high hopes of development of ANNs but I also understand the obstacles of this technology. And I truly hope that I have a chance to be a part of this development with this thesis and with my future work as a software engineer.

7 REFERENCES

[1] A. Turing, “Computing machinery and intelligence,” *Mind,* no. 236, pp. 433-460, October 1950.

[2] European Parlament, “News European Parlament,” 4 9 2020. [Online]. Available: https://www.europarl.europa.eu/news/en/headlines/society/20200827STO85804/what-is-artificial-intelligence-and-how-is-it-used. [Accessed 20 3 2021].

[3] R. B. Fletcher and J. Hattie, Intelligence and Intelligence Testing, London and New York: Taylor & Francis Group, 2011, pp. 5-6.

[4] S. J. Russel and P. Norvig, Artificial Intelligence A Modern Approach, Third Edition ed., 2010.

[5] P. McCorduck, “History of Artificial Intelligence,” 1977. [Online]. Available: https://www.ijcai.org/Proceedings/77-2/Papers/083.pdf.

[6] K. Kennedy, “Syracuse University,” 2017. [Online]. Available: https://surface.syr.edu/cgi/viewcontent.cgi?article=1018&context=wp.

[7] L. Fernandes, “The Abacus: A Brief History,” 11 January 2015. [Online]. Available: https://www.ee.ryerson.ca/~elf/abacus/history.html.

[8] I. Lindell, Sähkön pitkä historia, Tampere: Otatieto, 2009, p. 65.

[9] T. E. o. E. Britannica, “Cybernetics,” 10 March 2005. [Online]. Available: https://www.britannica.com/science/cybernetics.

[10] “Communication theory,” 14 January 2021. [Online]. Available: https://en.wikipedia.org/wiki/Communication_theory.

[11] B. G. Buchanan, “A (Very) Brief History of Artificial Intelligence,” *AI Magazine,* pp. 53-60, 2006.

[12] M. R. Williams, The First Computers, History and Architectures, 2002 ed., U. H. Raúl Rojas, Ed., London: Massachusetts Institute of Technology, 2000, p. 2.

[13] L. D. Mol, “Turing Machines,” 2019. [Online]. Available: https://plato.stanford.edu/archives/win2019/entries/turing-machine/.

[14] T. Boobier, “Advanced Analytics and AI,” in *Impact, Implementation and Future of Work*, vol. Artificial Intelligence, John Wiley & Sons, Incorporated, 2018, pp. 40 - 41.

[15] J. Moor, “The Dartmouth CollegeArtificial Intelligence Conference: The Next Fifty Years,” *AI Magazine,* no. 4, p. 87, 2006.

[16] E. Brennan, “Why Does Film and Television Sci-Fi Tend to Portray Machines as Being Human,” in *Communicating with Machines*, 2016.

[17] B. Marr, “The Amazing Ways Duolingo Is Using Artficial Intelligence To Deliver Free Leanguage Learning,” 16th October 2020. [Online]. Available: https://www.forbes.com/sites/bernardmarr/2020/10/16/the-amazing-ways-duolingo-is-using-artificial-intelligence-to-deliver-free-language-learning/?sh=7cf8e0785511. [Accessed 1st October 2021].

[18] Utopia Analytics, ”Utopia Analytics provides the finest AI services on the planet.,” 2020. [Online]. Available: https://utopiaanalytics.com/about-us/. [Haettu 1st October 2021].

[19] Valossa Labs Oy, “Increase Business Efficiency,” 2021. [Online]. Available: https://valossa.com/. [Accessed 1st October 2021].

[20] G. Webster, “NASA Mars Rover Getting Smarter as it Gets Older,” NASA, 2010. [Online]. Available: https://www.nasa.gov/mission_pages/mer/news/mer20100323.html. [Accessed 1st October 2021].

[21] B. Marr, “Key Milestones of Waymo - Google's Self Driving Cars,” 21st September 2018. [Online]. Available: https://www.forbes.com/sites/bernardmarr/2018/09/21/key-milestones-of-waymo-googles-self-driving-cars/?sh=1a37b6f85369. [Accessed 1st October 2021].

[22] N. Goksel-Canbek and M. E. Mutlu, “On the track of Artificial Intelligence: Learning with Intelligent Personal Assistants,” *International Journal of Human Sciences,* vol. 13, no. 1, 2016.

[23] H. J. E. Korteling, G. Van De Boer-Visschedijk, R. A. M. Blankendaal, R. Boonekamp and A. R. Eikelboom, “Frontiers in artificial intelligence,” 25 March 2021. [Online]. Available: https://doi.org/10.3389/frai.2021.622364.

[24] BBC, “Go master quits because AI 'cannot be defeated',” 27 November 2019. [Online]. Available: https://www.bbc.com/news/technology-50573071. [Accessed 4th October 2021].

[25] N. Bostrom, “How long before superintelligence,” *Int. Jour. of Future Studies,* vol. 2, 1998.

[26] A.-M. Rusanen and P. Ylikoski, “Algoritmit, tekoäly - tieteen murros?,” *FUTURA,* vol. 2, no. 17, pp. 15-25, 20th July 2017.

[27] S. Kohli, S. Miglani and R. Rapariya, “Basics of artificial neural network,” *International Journal of Computer Science and Mobile Computing,* vol. Vol. 3, no. Issue.9, pp. 745-751, September 2014.

[28] F. Bre, J. M. Gimenez and V. D. Fachinotti, “Prediction of wind pressure coefficients on building surfaces using Artificial Neural Networks,” *Energy and Buildings,* vol. 158, 1st November 2017.

[29] G. Sanderson, “But what is a Neural Network?,” 5th October 2017. [Online]. Available: https://www.3blue1brown.com/lessons/neural-networks. [Accessed 9th August 2021].

[30] B. M. Wilamowski, “Neural Network Architectures and Learning Algorithms,” *IEEE Industrial Electronics Magazine,* pp. 56-63, December 2009.

[31] A. Shrestha and A. Mahmood, “Review of Deep Learning Algortihms and Achitectures,” *IEEE Access,* vol. 7, pp. 53040-53065, 22nd April 2019.

[32] D. T. La Crote, “Newton's Method Backpropagation for Complex-Valued Holomorphic Neural Networks: Algebraicand Analytic Properties,” University of Wisconsin-Milwaukee, Wisconsin-Milwaukee, 2014.

[33] T. Cai, R. Gao, J. Hou, S. Chen, D. Wang, D. He, Z. Zhang and L. Wang, “A gram-gauss-newton method learning overparameterized deep neural networks for regression problems,” *arXiv preprint,* 28th May 2019.

[34] C. C. Aggarwal and C. K. Reddy, Data Clustering, T. &. F. Group, Ed., CRC Press, 2014, pp. 5-7.

[35] H. Pandey, “K-Means Clustering,” 28th March 2020. [Online]. Available: https://medium.com/analytics-vidhya/k-means-clustering-ee17930c177d. [Accessed 4th October 2021].

[36] W. Badr, “Auto-Encoder: What Is It? And What Is It Used For? (Part 1),” 22nd April 2019. [Online]. Available: https://towardsdatascience.com/auto-encoder-what-is-it-and-what-is-it-used-for-part-1-3e5c6f017726. [Accessed 4th October 2021].

[37] E. Tzorakoleftherakis, “Three Things to Know About Reinforcement Learning,” 2021. [Online]. Available: https://www.kdnuggets.com/2019/10/mathworks-reinforcement-learning.html. [Accessed 5th October 2021].

[38] J. Delua, “Supervised vs. Unsupervised Learning: What's the Difference?,” 12nd March 2021. [Online]. Available: https://www.ibm.com/cloud/blog/supervised-vs-unsupervised-learning. [Accessed 5th October 2021].

[39] Y. Mataga, *Analysis of Control Attainment in Endogenous Electroencephalogram Based Brain Computer Interfaces,* 2017, p. 20.

[40] “Wikipedia,” 14th September 2021. [Online]. Available: https://en.wikipedia.org/wiki/Sigmoid_function. [Accessed 14th September 2021].

[41] “Wikipedia,” 22nd July 2021. [Online]. Available: https://en.wikipedia.org/wiki/Rectifier_(neural_networks). [Accessed 14th September 2021].

[42] M. F. McNeill and E. Thro, Fuzzy Logic, Academic Press Limited, 1994, pp. 7-15.

[43] G. Sanderson, “Gradient descent, how neural networks learn,” 16th October 2017. [Online]. Available: https://www.3blue1brown.com/lessons/gradient-descent. [Accessed 9th August 2021].

[44] R. Jain, *3 Types of Gradient Descent Algorithms for Small & Large Data Sets,* 2017.

[45] Keerthana, “STATS101: Linear Regression,” 1st September 2018. [Online]. Available: https://medium.com/@keerthana21/stats101-linear-regression-684b67eff36b. [Accessed 26th October 2021].

[46] H. Sak, A. Senior ja F. Beaufays, ”Long Short-Term Memory Recurrent Neural Network Architectures,” 2014.

[47] Helsingin Yliopisto, Reaktor, “Elements of AI,” [Online]. Available: https://course.elementsofai.com/fi/5/3. [Accessed 24 August 2021].

[48] R. Midya, “Medium,” 20th October 2020. [Online]. Available: https://medium.com/swlh/ai-revolution-voice-assistants-their-smartness-c7afe2580e74. [Accessed 30th August 2021].

[49] R. Ullah, “Computing power for neural networks,” 2nd April 2020. [Online]. Available: https://www.fasthosts.co.uk/blog/computing-power-neural-networks/. [Accessed 5th October 2021].

[50] S. Badillo, B. Banfai, F. Birzele, . I. Davydov, L. Hutchinson, T. Kam-Thong, J. Siebourg-Polster, B. Steiert and J. D. Zhang, “An Introduction to Machine Learning,” *Clinical Pharmacology & Therapeutics,* no. 107, March 2020.

[51] M. Imran, ”Advantages of Neural Networks - Benefits of AI and Deep Learning,” 9th March 2020. [Online]. Available: https://www.folio3.ai/blog/advantages-of-neural-networks/. [Haettu 5th October 2021].

[52] K. Sunil, “Advantages and Disadvantages of Artificial Intelligence,” Towards Data Science, 2019.

[53] M. Tomas, “Neural Networks: Advantages and Applications,” 18th April 2019. [Online]. Available: https://www.marktechpost.com/2019/04/18/introduction-to-neural-networks-advantages-and-applications/. [Accessed 5th October 2021].

[54] N. Smita, “Potential use of Artificial Neural Network in Data Mining,” in *2010 The 2nd International Conference on Computer and Automation Engineering (ICCAE)*, Singapore, 2010.

[55] E. Rinta-Runsala and C. Bounsaythip, “Overview of Data Mining,” VTT, Espoo, 2001.

[56] J. Brand, “Medium,” 29th October 2019. [Online]. Available: https://medium.com/kwadigoai/deep-learning-for-image-recognition-1d612be00bbb. [Accessed 30th August 2021].

[57] M. Chatterjee, “Top 20 Applications of Deep Learning in 2021 Across Industries,” 19th February 2019. [Online]. Available: https://www.mygreatlearning.com/blog/deep-learning-applications/. [Accessed 30th August 2021].

[58] T. Karras, T. Aila, S. Laine and J. Lehtonen, “Progressive Growing of GANs for Improved Quality, Stability and Variation,” 2018.

[59] J. Schmidt, M. R. G. Marques, M. A. L. Marques and S. Botti, “Recent advances and applications of machine learning in solid-state materials science,” *npj computational materials,* 8th August 2019.

[60] W. Zhang, Computation Ecology, World Scientific Publishing Co. Pte. Ltd., 2010.

[61] S. Swaddiwudhipong, K. Tho, Z. Liu and J. Hua, “Material Characterization Using Artificial Neural Network,” Singapore, 2004.

[62] Z. Li, Y. Li, Y. Liu, P. Wang, R. Lu and H. B. Gooi, “Deep Learning Based Densely Connected Network for Load Forecasting,” *IEEE Transactions on Power Systems,* vol. 36, no. 4, pp. 2829-2840, 2021.

[63] K. S. Byoung and A. J. Calise, “Nonlinear Flight Control Using Neural Networks,” *Journal of Guidance, Control, and Dynamics,* vol. 20, no. 1, January-February 1997.

[64] H. Bundarupally, H. R. Talusani and T. Sridevi, “Detection of Military Targets from Satellite Images using Deep Convolutional Neural Networks,” in *2020 IEEE 5th International Conference on Computing Communication and Automation (ICCCA)*, Greater Noida, India, 2020.

[65] P. Vähäkainu and P. Neittaanmäki, *Tekoäly terveydenhuollossa,* Jyväskylä: Jyväskylän yliopisto, 2018.

[66] T. Magajna, “How to Find Wally with a Neural Network,” 2nd December 2017. [Online]. Available: https://towardsdatascience.com/how-to-find-wally-neural-network-eddbb20b0b90. [Accessed 30th August 2021].

[67] S. Subbulakshmi Radhakrishnan, A. Sebastian, A. Oberoi, S. Das and S. Das, “A biomimetic neural encoder for spiking neural network,” *Nature Communivations,* 2021.

8 APPENDIX

As part of this thesis a practical demonstration work was carried out as proof of applied knowledge of the topic. Objective of this project was to create progressive generative adversarial neural network that creates new artwork from any kind of artwork dataset. Below as an attachment is the Final Report of this project.

OULUN AMMATTIKORKEAKOULU

Seila Laakso

Progressive Generative Adversarial Artificial Neural Network - ProGAN DaliA

Final Report of a Business-Driven Project

Progressive Generative Adversarial Artificial Neural Network - ProGAN DaliA

Final Report of a Business-Driven Project

Seila Laakso
Report
Spring 2022
Information and communication technologies
Oulu University of Applied Sciences

SUMMARY

Oulu University of Applied Sciences
Information and Communication Technologies, Software Development

---

Author: Seila Laakso
Project Name: Progressive Competing Artificial Neural Network - ProGAN DaliA
Supervisor: Anne Keskitalo, Jukka Jauhiainen and Manne Hannula
Term and year of completion of work: Spring 2022
Number of pages: 20

Project report related to a study module called a business-oriented project. This project produced an AI network that creates new pieces of art based on paintings by different artists.

The project is also part of the written bachelor's Thesis on artificial neural networks and deep learning.

Keywords: Artificial neural network, PROGAN, neural network, artificial intelligence, AI, GAN, generative adversarial neural network.

CONTENTS

1 INTRODUCTION

This report examines a Progressive Generative Adversarial Neural Network project called ProGAN DaliA. This project aimed to create an artificial neural network that would create new surreal works of art based on paintings by Salvador Dalí.

The report describes how the project was conducted and why surreal pieces of art works were used. The course of the project and the problems that occurred during it, as well as how they were resolved, are discussed in section three. The report also examines the architecture and principle of operation of the final neural network in section 4. Finally, the report also discusses the outcome of the project and presents the student's self-assessment.

2 PROJECT STARTING POINT

This project is part of a student's written bachelor's Thesis called 'Neural Networks and Deep Learning - Possibilities and limits of its use in modern software development. In addition to the thesis, a practical work was needed for the completion of the student's double degree at Technological University Dublin (later TUD). This previous study on neural networks and deep learning therefore also served as a starting point for the project, and therefore there was little need for brainstorming for the project. It was just a matter of deciding what kind of an artificial neural network the project would be and what its goal was. This project was produced as part of a study module called a business-oriented project for Oulu University of Applied Sciences.

2.1 Why Exactly Progressive Adversarial Neural Network?

As stated in the thesis mentioned, artificial neural networks come in many forms and their architecture variate greatly to achieve the desired results. The aim of the project was to select an architecture that would be related to the neural networks that were already discussed in the thesis and would be comparatively recent technology, but it would still be possible for one person to implement in the given time frame. The objective was ambitious. ProGAN CelebA neural network discussed in the thesis served as an inspiration for the project. Therefore, colour image data was used. Based on photographs of public figures, ProGAN CelebA creates new photographs of public figures that do not actually exist. The idea was quite simple but fascinating. The project was also wanted to be ground-breaking. A project that had never been done before, but it would be possible to develop with the application of previous examples. Thus came the idea of ProGAN DaliA, a progressive generative adversarial neural network capable of producing works of art based on works by Salvador Dalí.

2.2 Why Salvador Dalí?

The aim of the project was to use data that would be clearly distinguished from the data previously used in similar projects, such as previously mentioned CelebA neural network. The project ended up using data that presents people with things, situations and objects, but not in an environment that we perceive as traditional for example, in Salvador Dalí's work The Persistence of Memory (org. La sistencia de la memoria) (Fig. 1). The artwork reflects objects that are very clearly understandable to man, such as the eye, pocket watch, beach, rock, dried wood and ants. During the project, the aim was to find out whether an artificial neural network could distinguish these objects from images and, if so, what would be the works created with these images.

FIGURE 1 Salvador Dalí's work The Persistence of Memory is a good example of human's visual observation skill. In the image, objects are presented in an unusual environment for them. [1]

2.3 Programming Language and Selection of the Programming Platform

The previously discussed thesis, on which this project is based, explored several of the latest neural network applications. These applications were created almost invariably using python language. This is probably due to the existence of built-in libraries on Python and the fact that the language characteristics best match the features needed for the software, such as image processing, mathematical characteristics such as multiplying multiple-dimensional arrays, and computing power-enhancing features that are needed for neural network programming. The choice of language was therefore almost self-evident, but due to the lack of previous experience with a similar project, choosing a programming platform was a more challenging task.

Since there was no better knowledge of the matter at the beginning of the project, PyCharm, which was already familiar from previous Python projects, was chosen as the programming platform. PyCharm is a software installed on a PC that uses a local processor to run a program. This became a problem after the beginning of the project, and more information about this can be found later in the report.

3 PROJECT FLOW

3.1 In Search for the Head of Yarn

When starting a project like this, some people are surely familiar with the thought "where would I start this project?". At first, there was little to no previous practical experience in the creation of an artificial neural network. In theory, there was a lot of information, but based on the guide videos, online forum discussions and online examples, the project was still nipped in. During the first weeks of the project, the working hours consisted largely of searching for information and testing various examples on PyCharm.

3.2 PyCharm and Lack of Computing Power

Based on the thesis, it was known that it would take a lot of computing power to complete an artificial neural network. However, the current laptop, which could run even heavier PC games, was thought to be capable of such a performance when given enough time. However, this was not the case. While running the program a connection error (Figure 2) relating to the absence of CUDA Toolkit was encountered plenty of times. This Toolkit is software that allows computers with NVidia graphics cards and CUDA readiness to use a graphics card alongside a traditional CPU to improve computing power. The laptop that was used had full CUDA support. However, despite dozens of attempts, the program encountered compatibility issues with libraries used by the code that was already almost completed. When looking for a solution to the problems, it was noticed that the problem is quite common in this programming platform. Bringing independent libraries from several different actors into one code was challenging. Each of the libraries had to have specific versions. Compatibility problems were eventually discovered to be like a vicious circle. When one version of a library was used somewhere in the program, it would not be compatible with the Cuda Toolkit or another library.

The project had reached a stalemate. These libraries were necessary. Programming the layers, which make up the neural network, by hand would be weeks of work, if not months, of which, even if it had been completed, would it have finally helped the underlying problem, which was the lack of computing power. The software, which runs more than six hours, resulted in images that resembled ruined colour photographs. As a sum of all these problems, it was concluded that the direction of the project should be changed.

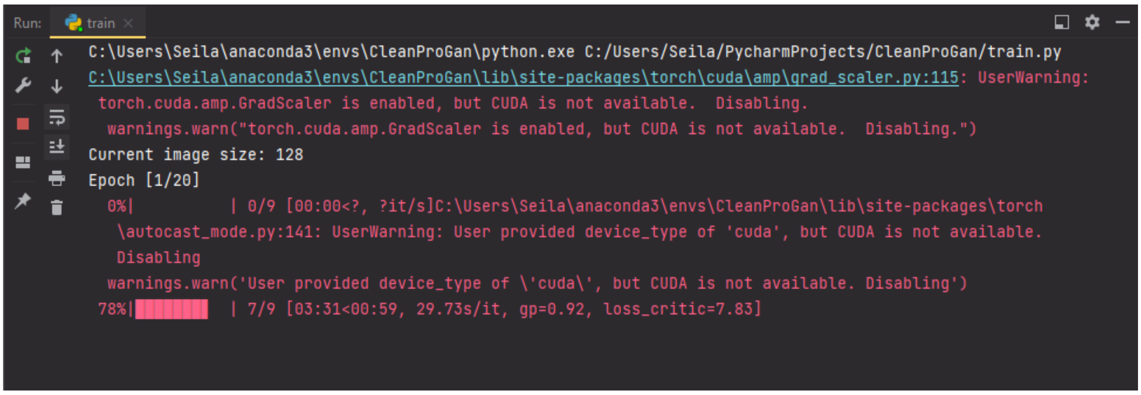


FIGURE 2 PyCharm notes on graphic card compatibility issues.

3.3 Google Colab and a Fresh Start

The project did not proceed. Despite several error corrections, the same problem occurred, and the neural network was unable to create a single image that would even remotely resemble a piece of art. As the problem persisted, it was decided that some advice was needed. Hence Jukka Jauhiainen, the Thesis supervisor was asked for help, who after hearing about the problem, suggested that the project would be moved to Google Colab environment. The author had no previous experience of this platform, except for few previously eyed online forum writings. From the very beginning, it was noticed that the platform was completely different from the previously used PyCharm. The code was run in sections and there was no need to bind multiple python files together. The program was run in a browser and Google's own servers were used to help with computing power. It was immediately concluded that it did not make sense to plant the previous code on a new platform. The resulting number of problems would certainly have exceeded the workload required to start from scratch. And there was no guarantee that the problems encountered in PyCharm had been rectified by just changing the platform. The project continued with a clean slate.

It was therefore decided to start to look for new examples using Colab as the platform. At this time, it was discovered that the use of PyCharm had caused the project to stray on the very wrong track. Colab would now on be significantly more straightforward to use than PyCharm. And in just over a week, the finished GAN was producing its first images. However, the images were not as expected. A uniformly grey picture, in which the functioning of the code was confirmed only by variation in the shades of the individual pixels (Figure 3), and no development between the training cycle images was seen. We were faced with yet another problem.

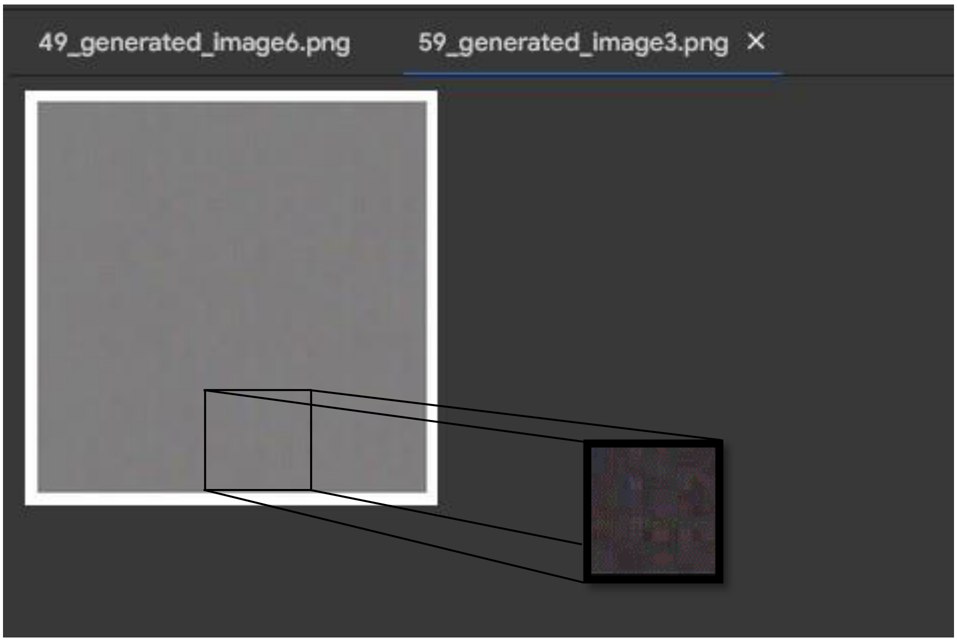


FIGURE 3 The first colour image produced with Google Colab. The differences in tone between individual pixels confirm that the code works but does not learn as desired. Excerpt on the right modified to highlight colour differences in the original image.

Jukka Jauhiainen was asked for guidance again. This time, however, it was an error in the code. After careful reviewing, two of them were discovered.

Problem one.

The sample code used a dataset called Cifar10 as its dataset. This did not correspond to the data needed for the project, as the data wanted to contain Dalí's works of art. The part of the code that processes the data, therefore, had to be completely rewritten. Although the lines look short (Figure 4), this expression firstly uploads the dataset art pieces into the Colab cloud and then places each of the images into four-dimensional tensors that are exactly the same size every time regardless of the original image size and are also the same size as the random data tensors that are used in the discriminator training. Data in these tensors include the index of each picture, the height and width coordinates and the RGB value of the pixel in question. The structure of this part of the code is illustrated in Figure 5.

```python
def load_images():
  filelist = files.upload()
  data_list = []

  for myfile in filelist:
    org_image = cv2.imread(myfile)[:, :, ::-1]
    resized = cv2.resize(org_image,(64, 64),
                         interpolation = cv2.INTER_AREA,)

    resized = resized.astype('float32')
    resized = (resized - 127.5) / 127.5
    data_list.append(resized)

  Xtrain = np.array(data_list)

  return Xtrain
```

FIGURE 4 The image upload clause contains the 4D matrix below for colour images formatted to the size of 64x64, which is uploaded from the cloud to the matrices one timespan.

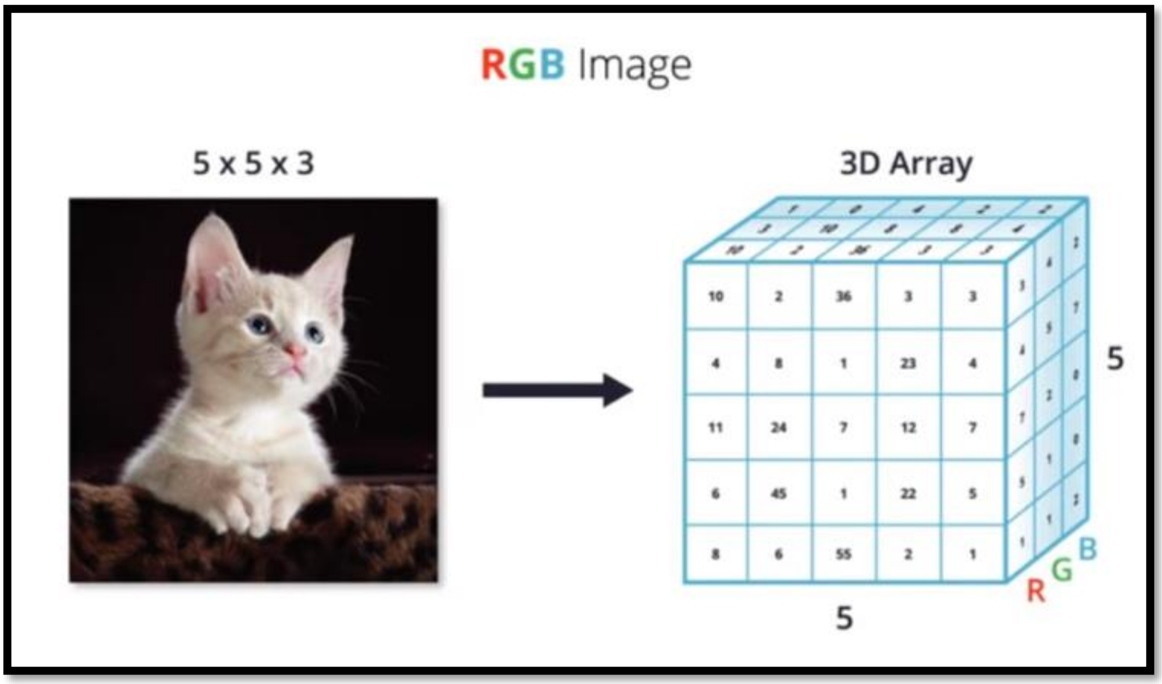


FIGURE 5 Description of the data matrix structure used in ProGAN DaliA. The 3D matrix in the image represents data from a single image. The fourth dimension consists of several images used. [2]

Therefore, in order for ProGAN DaliA to be able to use these tensors in their learning, they should be exactly a certain size. The original size of the art pieces also had to meet certain thresholds in order for the program to run successfully. However, the problem with why the images were grey was revealed in Figure 4 in a latent, orange-underlined row, in which the data from the image is edited from an integer (assumed by the program) to a fraction value (aka. float) that was successfully processed by other parts of the code.

Problem two.

After solving the problem of working with the images, another problem was identified. Now, even if the code was running smoothly and we got satisfactory images, in contrary to the assumption, the images did not "evolve" over time when the code was run for a longer period of time. The assumption for a neural network is that once one *batch* (as in batch of data) has been processed, the neural network updates the parameters to more accurately reflect the training data in question and thus learns. However, this learning event was not noticeable. The code was, ones again, carefully examined and even if it took some time eventually the missing clause seen in Figure 6 was found to be the cause of the problem.

```
def training(data, generator_model, discriminator_model, epochs, n_batch, start= 0):
  dimension_batch = int(data.shape[0]/n_batch)
  half_dataset = int(n_batch/2)

  # We iterate over the epochs
  for epoch in range(start, start + epochs):
    # We iterate over all batches
    for batch in range(n_batch):

      # Load all of the real data
      X_real,Y_real = load_data(dataset, half_dataset)


      # Train the discriminator with real and fake data
      cost_discriminator_real, _ = discriminator_model.train_on_batch(X_real, Y_real)
      X_fake, Y_fake =  create_fake_data(generator_model,half_dataset)

      cost_discriminator_fake, _ = discriminator_model.train_on_batch(X_fake, Y_fake)

      # Generate the input images for the GAN
      X_gan = generate_data_entry(half_dataset)
      Y_gan = np.ones((half_dataset, 1))

      # Train the GAN with fake data
      cost_gan = gan.train_on_batch(X_gan, Y_gan)

    # Every 10 epochs we show the results and cost
    if (epoch+1) % 10 == 0:
      evaluate_file(generator_model, epoch= epoch, half_dataset=half_dataset)
      show_generated_images(X_fake, epoch = epoch)
```

FIGURE 6 The missing part of the program (circled in red) was the reason why the neural network did not learn and evolve.

3.4 First Results and Fine Tuning

Once the problems in the code had been resolved, it was finally possible to explore the goals that were set at the beginning of the project. Was it possible to get the neural network to produce new surreal works of art? 139 works of art by Salvador Dalí were collected in the first imagery. However, the results of the code repeated again and again that the images did not represent any object or subject that could be clearly detected. In some cases, it also happened that the neural network decided to copy one of the original works. Which, although there was progress, did not meet the original goal, which was for the neural network to create similar works of art and not to directly copy original works.

A lot of tests were made. And although the structure of the Generator and Discriminator was modified, no better results were obtained than before. The dataset was also increased by adding surrealistic works by other artists. This dataset contained a total of 500 surrealistic works by various artists such as Joan Miro, Pablo Picasso, Rene Magritte and, of course, Salvador Dalí. Despite this, the results achieved in the smaller data were also repeated. The pictures didn't represent any observable objects. Examples of results can be seen in Figure 7.

FIGURE 7 Example of illustrations of surrealistic works of art created by ProGAN DaliA.

At a later stage, the aim was to increase the size of the images, which are 64x64 in size, to a higher resolution. However, these attempts were obstructed by the computing power limitations imposed by Colab. The computing power that would have been needed to produce larger images was behind the paywall, which meant that we had to settle for 64x64-sized images.

The original objectives had now almost been achieved. Now to conclude the work, a test was needed whether DaliA would be able to produce images that would actually represent an object when specific images were used in the training. This required a different kind of dataset to be used. It was intended to prove that the originally set goal of an artificial neural network that was able to separate objects in the way described in section 2.2 ‘Why Salvador Dalí?’, would not be achievable by this type of neural network technology.

The new tests ended up using two different subjects, painted flower arrangements (Figure 8) and painted facial images (Figure 9). The results, especially for flower arrangements, were beyond positive. When asked about the subject, four out of five reviewers said that the images clearly represent flower arrangements. In the results, it should also be taken to account the fact that the imagery used does not represent photographed flower arrangements but painted. Therefore, the artist's freedom of expression in the original works, such as contrast and use of shapes and colours is also seen reflected in the end result. The artist's view is particularly noticeable from the results of the dataset that illustrates facial images. The material used contained about 500 painted facial images with a mixture of images of women, men and children. Some of these faces are also very abstract. In addition, the person could be facing either right, left or straight. Pictures painted directly from the person's side so that only one eye was visible were omitted from the material. The data had also been modified so that the images showed one person's face up from the clavicle. However, there were also some variations in the paintings including different headgear, the genre of art, and also for example whether the person's hands were visible in the painting or not. Examples of the paintings used in the imagery are given in Figure 9.

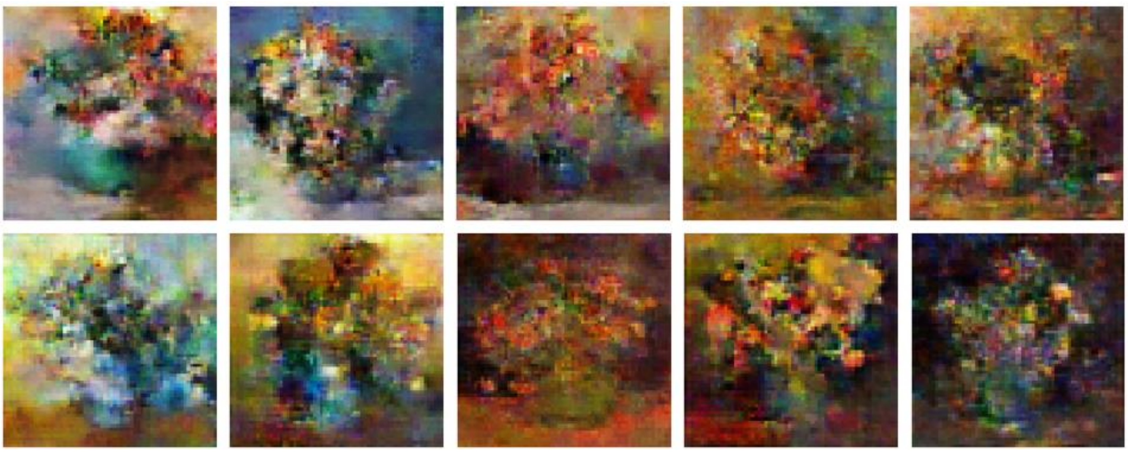


FIGURE 8 Works of art depicting the flower arrangement created by ProGAN DaliA.

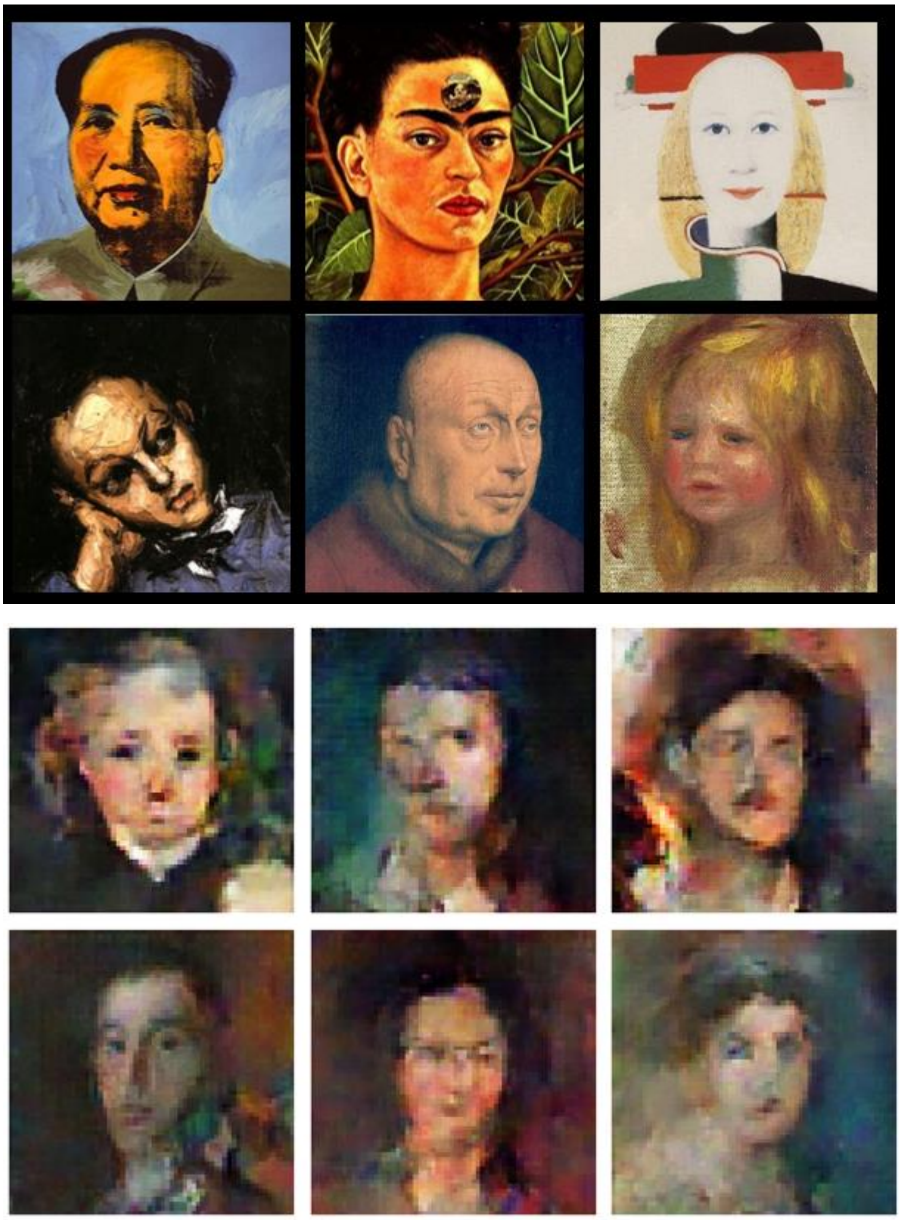


FIGURE 9 Above, (on a black background) example of the paintings used in the training dataset. Further down, (on a white background), paintings created by ProGAN DaliA portraying the faces of individuals. [2] [3] [4] [5] [6] [7]

4 PROGAN DALIA'S ARCHITECTURE

In reality, generative adversarial neural networks consist of two separate artificial neural networks. Generator (Fig. 10) and discriminator (Fig. 11). As one might playfully describe to the unfamiliar; ‘In this case, it could be said that the generator is an artist, and the discriminator is an art critic'. The generator creates images based on random noise, and the discriminator compares the result to the original works. The discriminator gives feedback to the generator on how much the images resemble the original art pieces, and the generator improves its performance based on this value, the so-called loss value. It could also be said that the generator and the discriminator compete with each other over whether the discriminator can distinguish the "fake" images created by the generator from the originals. This is where the architecture gets its name.

Artificial neural networks always consist of layers. The architecture of the neural network depends on what the network is to achieve. The layers reflect the mathematical performances that are performed on images packed into matrices so that GAN can learn and produce the desired result.

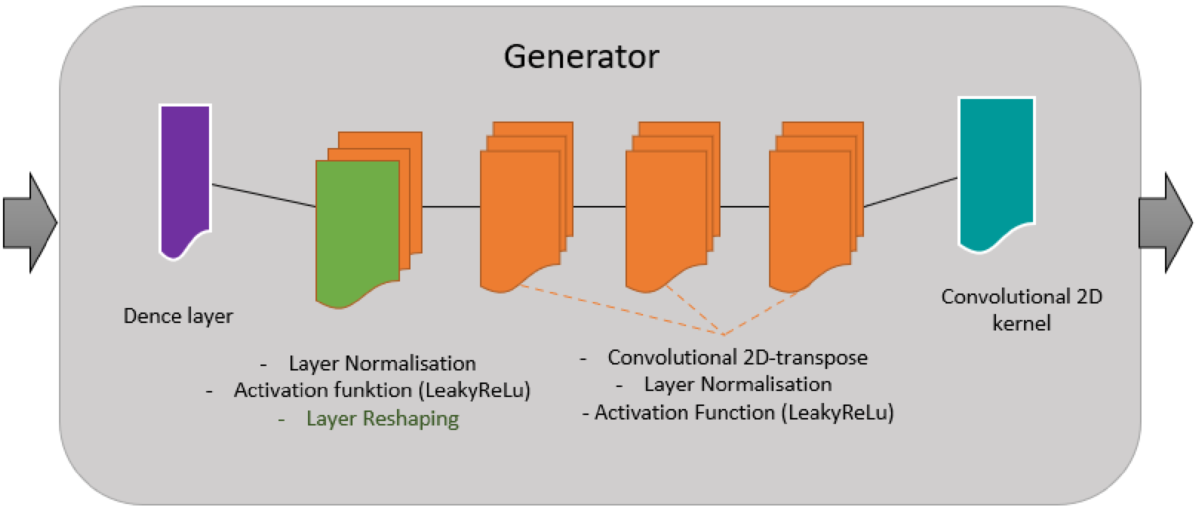


FIGURE 10 ProGAN DaliA generator structure visualised.

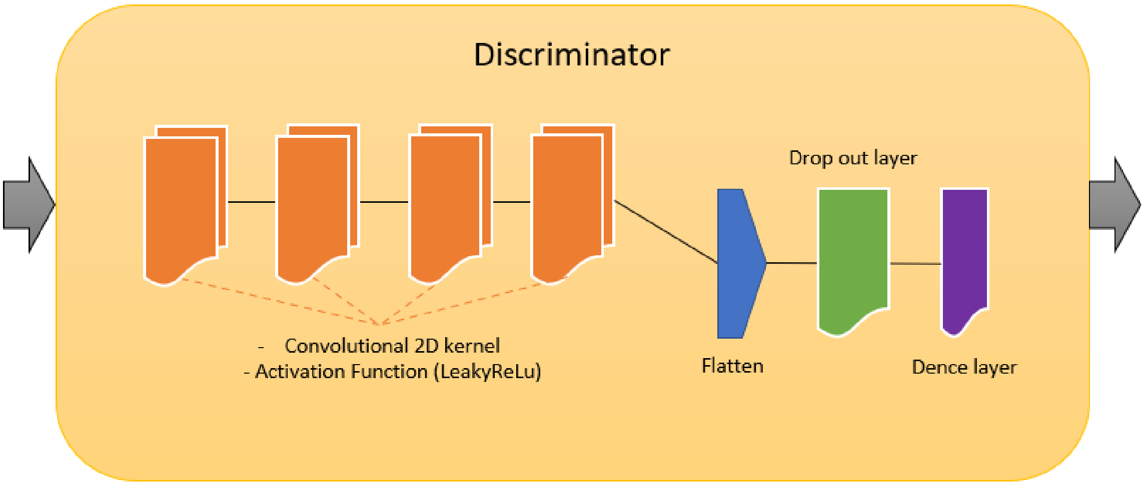


FIGURE 11ProGAN DaliA discriminator visualised.

By connecting the generator and the discriminator, we create a GAN (Fig. 12). The generator produces images and improves its performance based on the loss value produced by the discriminator.

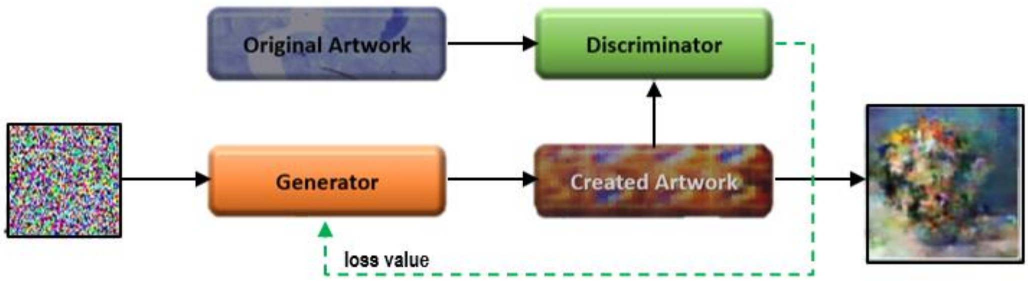


FIGURE 12 GAN's principle of operation.

5 END RESULT AND SELF ASSESSMENT

When I started the project, I took a conscious risk in choosing a job that no one else has done before. The work was significant in terms of both workload and objectives. There were a lot of challenges and problems, and sometimes also a piece of advice was needed to solve them. But I think the results are in line with that. I am very pleased with the results obtained and the project as a whole. Although the surreal works of art did not present any objects, they clearly show an artistic expression similar to the original art pieces. The works illustrating faces and flowers were also very similar to works such as the original imagery.

If I was looking for things I would do differently; at the beginning of the project, I would have looked for more information about the programming platform used for projects like this one. Moving the work from PyCharm to Colab caused hours of programming work to be wasted. But even though the program created at PyCharm could not be used at Colab, I had had time to gather a huge amount of know-how for the job, which was crucial in achieving the ultimate goal.

In the future, I want to continue working on similar projects in my working life. Work is very ground-breaking and experimental, which makes it fascinating. Artificial neural networks are now at the forefront of the wave of development. The potential for using this technology will only increase in the near future as the computing power of computers continues to increase. This project gave me valuable expertise in this technology on a practical level and I am sure it will be of enormous benefit to me in my working life.

The author:

Seila Laakso

6 REFERENCES

[1] S. Dalí, Artist, *The Presistence of Memory (La persistència de la memòria).* [Art]. Museum of Modern Art, 1931.

[2] A. Warhol, Artist, *Mao.* [Art]. 1972.

[3] F. Kahlo, Artist, *Thinking About Death (Pensando en la Muerte).* [Art]. 1943.

[4] K. Malevitš, Artist, *Girl with a Comb in her Hair (Девушка с гребнем в волосах).* [Art]. Tretyakov Gallery, 1933.

[5] P. Cezanne, Artist, *Portrait of a Young Man.* [Art]. Private Collection, 1866.

[6] J. v. Eyck, Artist, *Adoration of the Mystic Lamb (De aanbidding van het Lam Gods).* [Art]. St Bavo's Cathedral, 1432.

[7] P.-A. Renoir, Artist, *Coco's Head.* [Art]. Private Collection, Date unknown.

[8] S. Balachandran, "Machine Learning - Going Furthur with CNN Part 2," Dev, 17th March 2020. [Online]. Available: https://dev.to/sandeepbalachandran/machine-learning-going-furthur-with-cnn-part-2-41km. [Accessed 31st May 2022].