Artificial Intelligence

KS Im Lee

Kantonsschule Im Lee

Banuijan Yogananthan

Maturarbeit

Contents

0 Introdction

This text-paper discusses the fundamentals of artificial intelligence, which includes processes like machine learning, deep learning and their mathematical/geometrical background. Also, we will take a closer look at different types of so called "neural networks" and deduce the advantages/disadvantages of the important ones. It is meant to be used as an introduction to a Maura project dealing with the construction of AI's. By the end, one should receive a brief overview about this topic

1 What is Artificial Intelligence

AI, or artificial intelligence, is prevalent in our world. Its use is versatile: voice/text recognition such as Siri or Alexa, finance, video games, military, art, and even the government make use of this technology. With AI, processes that could take ages when done by a human can be completed within seconds by an external device. That could be a definition of AI: It's a human, but just better. AI is basically exceeding or matching the capabilities of a human. So, it tries to match the intelligence, whatever that means, and capabilities of a human subject. There are different abilities most AI's have in common :

- ability to discover
- ability to infer
- read information
- ability to reason
- ability to figure things out

For the construction of such systems, different types of frameworks can be used. We need such a framework in order to make sure that our AI is deciding in the most convenient way for our project. Now, we will take a closer look at such frameworks.

2 Machine Learning

The term "Machine Learning" was introduced in 1959 by the computer scientist "Arthur Samuel". It's a subset of AI and contains these two fundamental abilities:

- classification of data
- accurate predictions based on data

How it works:

In order to train our model, we need to collect data which will be fed to it. A diagram of the data could be represented in a coordinate system:

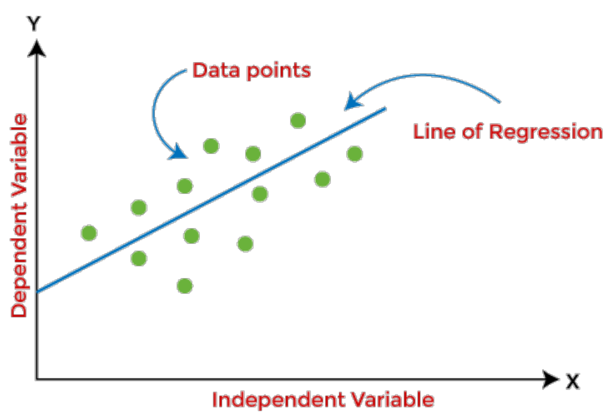


Figure 1: Data Points visualized

There are different facots which could influence the quality of the model. First of all, the data point should contain all the necessary information. If not, the accuracy will decrease. The same is for the "varianz", of the variance is high, there will be a change in the "line of regression" for high variances, more data is needed to get an accurate output. Other methods like normalising, converting or randomising values are also used for preparing data sets, which we won't go in detail. To avoid an increase in weight when a data point occurs twice, you just simply delete one.

After preparing the raw data, we need to come up with a technique which we could train our model with. the machine . For this, there are 3 different concepts that allows us to build such machines depending on its abilities :

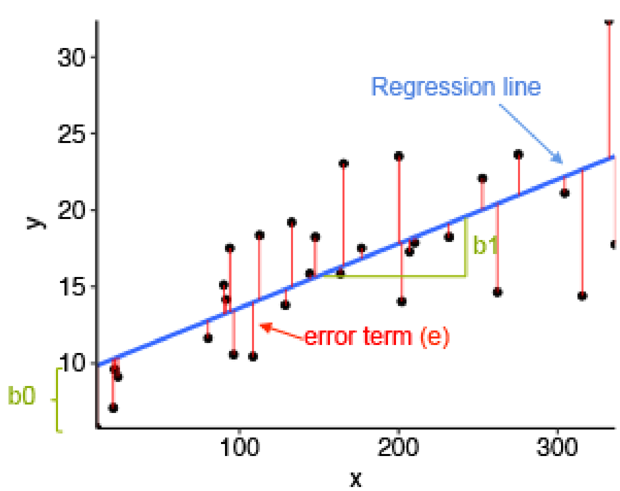
Supervised (Machine) Learning

In supervised learning, the model will be trained with labelled data. Labelling here means that the size which one wants to preach is contained in the data. It is not enough to put in the data and let it train. It needs certain information about the object or whatever is preached. Therefore, the connection between the data sets is already given before

Again, supervised learning is based on 2 concepts

Regression

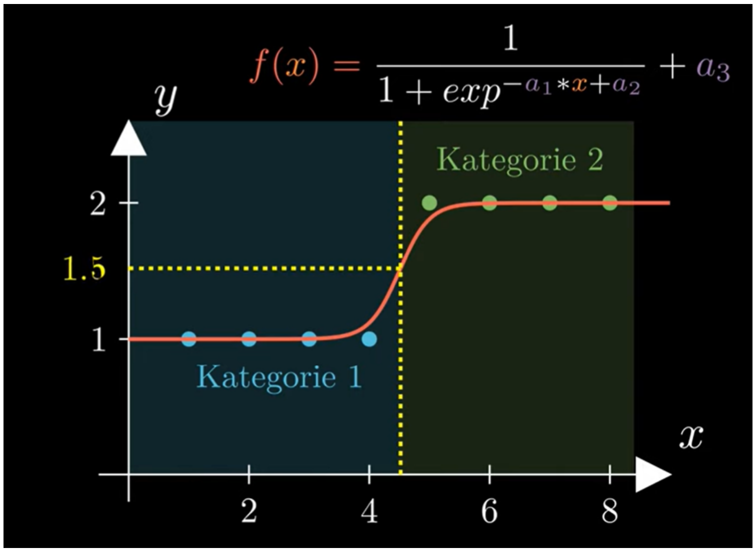
Consider a regression line: $f(x) = mx + q$. The line in this case will be determined by minimizing the distance from the line to all the data points by adjusting the parameter m and q:



$$min \sum_{i=0}^{N} d_i^2 \quad (1)$$

Classification

consider following function : $f(x) = \frac{1}{1+exp^{-a_1*x+a_2}} + a_3$:



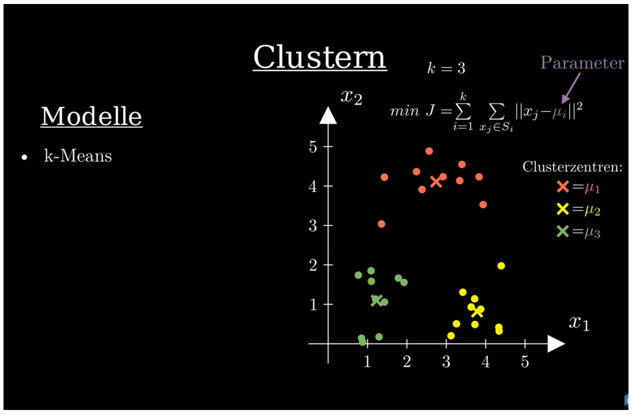
As you can see, depending on your ouput, it determines wether it belongs to category 1 or 2. This technique is used to classify data and is also part of supervised learning.

Unsupervised (Machine) Learning

In contrast to supervised, learning, this learning technique uses unlabelled data. This means, the size in the dataset is not given already before. The connection of the data is determined by the algorithm by its own.

The idea of unsupervised learning is again based on 2 concepts :

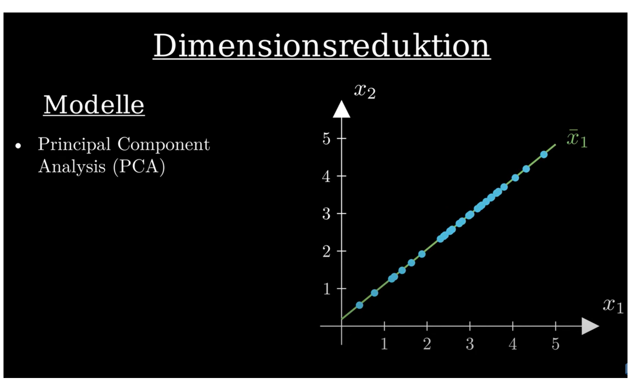
Clustering



Clustering is a concept where data points are categorized through so-called cluster-centres (by formular shown above). One could imagine these cluster-centres to span a circle where all the repsective points are included.

Reduction of Dimensions

This concepts yields (as the name tells us) a reduction of dimensions. This is not achieved by leaving out dimensions, it rather depends on the goals which are aimed onto (e.g reducing information loss). This is done by transforming the base of the coordinate system so that the points lie on one straight line



3 Deep Learning

Deep learning is a subsequent field of machine learning. In fact, deep learning is also based on the same concepts we have discussed above. The major difference of ML and DL are the amount of layers a neural network consists of. When ML- neural networks consist of 3 layers (Input-, hidden-, Output layer), deep learning networks consist of more than 3 layers. Whereas Machine learning is rather used for human interventions (e.g recognition of food), deep learning is more used with the concept of unsupervised learning, therefore with the ideas of clustering and reduction of dimensions:

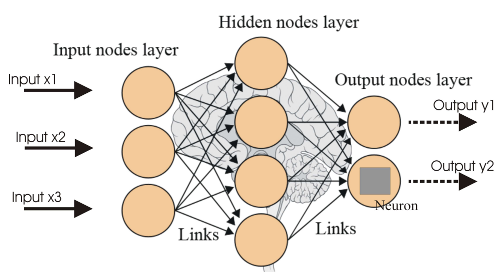


Figure 2: ML NN

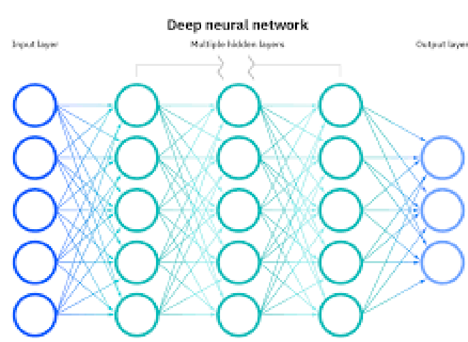


Figure 3: DL NN

Neural Networks

As this concepts is one of the fundamental parts when dealing with deep learning, we will focus on the concept of neural networks and try to understand how they function.

How it works

In order to understand the mechanism which lies behind Neural Networks, we will focus on understanding the single (2) components, namely neurons and weights. We will use a "Plain Vanilla" model, since it's the easiest one.

Neurons

Neurons are placeholders for specific values. These values are called "activation". In a way, you can imagine that if this value is high, the neurons will be more intensive, meaning this value has a high importance.

Weights

Weights are used to connect the layers of a neural network. With higher values, you can determine the importance of each neuron. As we dig deeper in examples, you can observe how certain patterns can be determined by setting up specific weight values

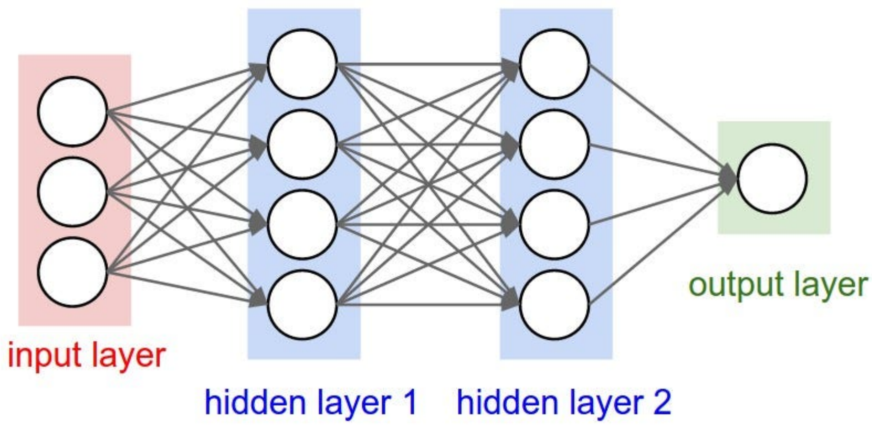
bias

A bias is a number you add after taking the scalar of all weight and "activations" to make sure the output is higher than the minimum

activation function

An activation function is used because normally, you want your values to be between 0 and 1. A common function which would to this is the sigmoid function.

Example: Plain Vanilla NN

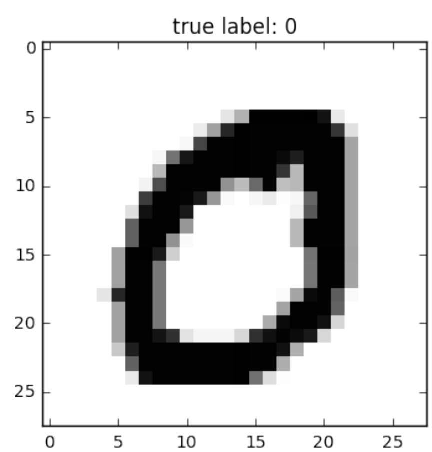


Let's say we have this type of neural network, and we train this network with the MNIST dataset. This dataset contains several thousand of handwritten numbers, so that in the end, the neural network can tell us which number is represented.

Input layer

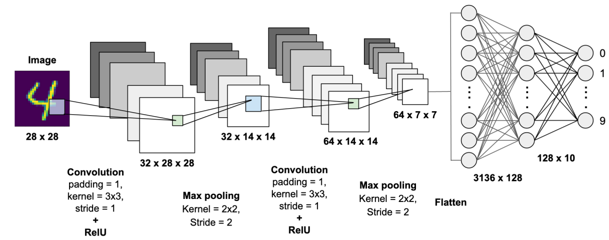
Let's say you have several examples of the number seven, one is depicted on the right. You could

consider every pixel of this image as a neuron which contains a certain value. In this case, how bright the cell is or not. This is contained in the activation number of this neuron. Like that, you could take a the value of a pixel and represent it as a neuron. This makes up the first layer of the Neural Network

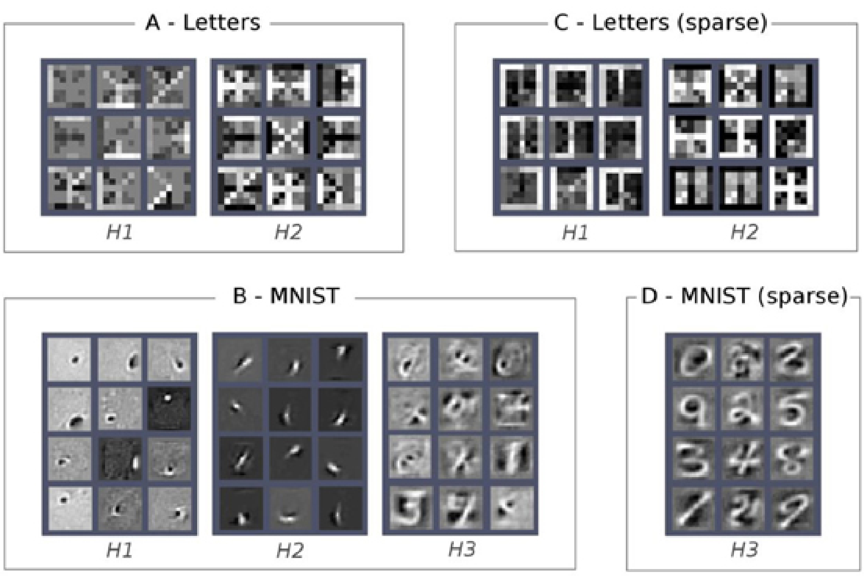


Hidden Layers

The Hidden layers in this case are responsible for determine specific patterns with the aid of weights. By increasing and decreasing the value of weights, one is able to determine



specific patterns. By increasing the amount of hidden layers, one is able to go more precise with these patterns, are represented on the right.



Output Layer

The output layer consists of all the scalars of weight and activation for the specific number. The neurons which shines the brightest (the neuron which holds the highest number, contains the right output and therefore the solution.

The value of the output neurons can be summed by the following formula :

$$sigmoid(w_1a_1 + w_2a_2 + w_3a_3 + ... + w_na_n - 10)$$

The following formula in words would be :

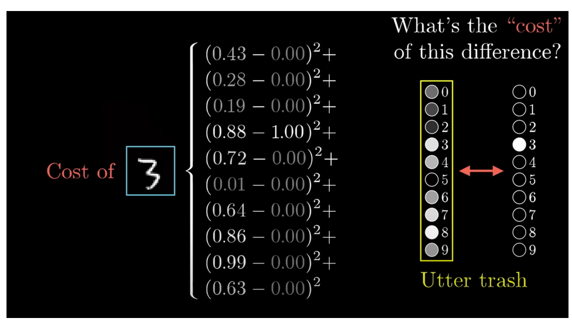
activation function(scalar product of all weight and sums - bias)

Learning process

A functioning neural network does not work perfectly without training. Indeed, the model will encounter many mistakes and therefore not the result one would expect. Basically, a neural network learns by its mistakes. The mistakes will lead to the adaption of weights (and therefore the sums), so that the weights are altered in the most convenient way.

The way this is done is with the aid of a so-called cost function. This is a way of telling the computer how extreme the mistake was.

By adjusting and minimizing the intensiveness of the mistakes, we would get the proper weights and therefore the right output You add up the squares of the differences of the "mistake" output and the value you want them to have. If we go back to our MNIST example :



I we want the number 3, all other values of the output layers should have a value in the near of 0. A way how to minize these mistakes can be done with the aid of the gradient and the first derivative, which we will not go in detail.

Backpropagation

4 Appendix

https://ai-leaders.de/portfolio/wo-wird-ki-eingesetzt/#:~:text=Regierungen%20und%20Beh%C3%B6rden%20nutzen%20KI,zur%20Erzeugung%20von%20Kunstwerken%20einzusetzen.