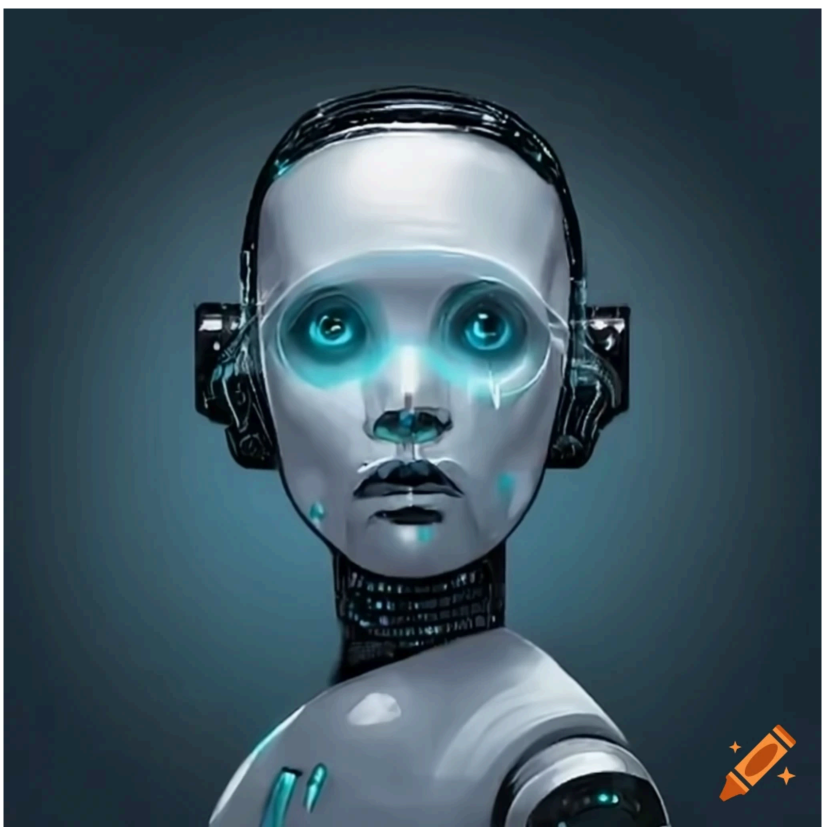
The Artificial Intelligence Journey

version 1.0
January-2024

By Dr. Shlomi Boutnaru



Created using Craiyon, AI Image Generator

Table of Contents

Introduction

Artificial Intelligence has become an integral part of our lives. Think about chatbots, image recognition, NLP (natural language processing), voice assistants, GenAI and more. Thus, it is something that many folks want to understand more from a technical perspective.

Overall, I wanted to create something that will improve the overall knowledge regarding Artificial Intelligence using writeups that can be read in 1-3 mins. I hope you are going to enjoy the ride.

Lastly, you can follow me on twitter - @boutnaru (https://twitter.com/boutnaru). Also, you can read my other writeups on medium - https://medium.com/@boutnaru.

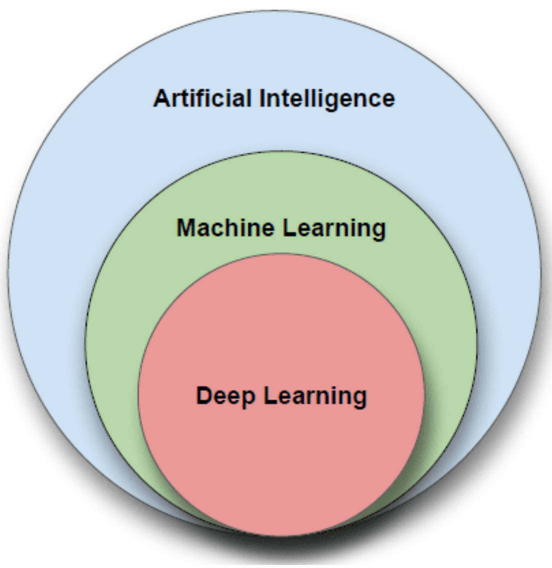
Lets GO!!!!!!

Artificial Intelligence (AI)

AI (Artificial Intelligence) is defined as the ability of machines/systems to enhance/replicate human intellect, think about reasoning and learning from experience. AI is based on the theory of probability, linear algebra and algorithms. We can break AI into two main fields: Machine Learning (ML) and Deep Learning (DL) - as shown in the diagram below[1].

In his paper "What is artificial intelligence?" from 2014 John McCarthy gave the following definition for "Artificial Intelligence": "It is the science and engineering of making intelligent machines, especially intelligent computer programs. It is related to the similar task of using computers to understand human intelligence, but AI does not have to confine itself to methods that are biologically observable"[2]. By the way, John McCarthy is an American computer scientist which received in 1971 a "Turing Award" for his contribution to the field of AI[3].

Thus, we can say that the goal of AI is to preserve, synthesize and infer information by computer systems in order to solve problems, represent knowledge, process natural languages and more. Think about tasks such as: computer vision, speech recognition, language translation, summarizing text and more. It is believed that AI was founded as an academic discipline in 1956[4].



[1] https://www.red-gate.com/simple-talk/development/data-science-development/introduction-to-artificial-intelligence/
[2] https://www-formal.stanford.edu/jmc/whatisai.pdf
[3] https://amturing.acm.org/award_winners/mccarthy_1118322.cfm
[4] https://en.wikipedia.org/wiki/Artificial_intelligence

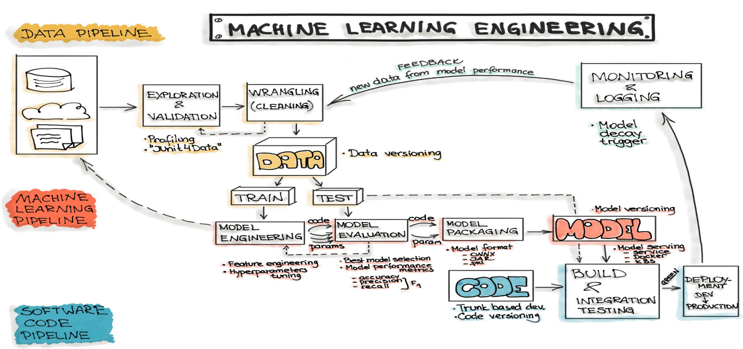
Machine Learning (ML)

Machine Learning is a subset/branch of artificial intelligence[5] which leverages the use of statistical methods in order to train algorithms for making classifications and/or peridications. This provides the ability to uncover key insights in data mining projects. This can be done using different frameworks like PyTorch and TensorFlow[6].

Due to the fact, we have not talked yet about "Deep Learning" (I will do that on future writeups) so for now we are going to focus on classical/"non deep" learning. Thus, machine learning is more dependent on human intervention like: selection of features, data cleaning and more. Think about the process of feature engineering which can include adding/mutating/combining data within the data set by experts for improving the results of the machine learning models[7].

Overall, we can categorize machine learning to four main types: "Supervised Learning", "Unsupervised Learning", "Semi-Supervised Learning" and "Reinforcement Learning". "Supervised Learning" is when we have a pre-labeled/classified dataset (like by users) which allows a machine learning model to measure its performance. "Unsupervised Learning" is when we are using raw datasets which are not labeled, in this case we try to find patterns/relations in the data without the user's help.

Moreover, "Semi-Supervised Learning" we have a dataset which has both labeled and unlabeled data which allows the models to learn how to label unlabeled data. "Reinforcement Learning" uses AI agents that try to find an optimal way to perform a specific task, when they take an action that goes towards the goal they get a reward[8]. More on those types in future writeups. Lastly, there are different phases in the machine learning pipeline like: data collection, data clearing, training a model, testing a model deploying it and more - as shown in the diagram below[9]. By the way, well known machine learning algorithms that you might have heard of are: decision trees, neural networks, linear regression, logistic regression SVM and more.



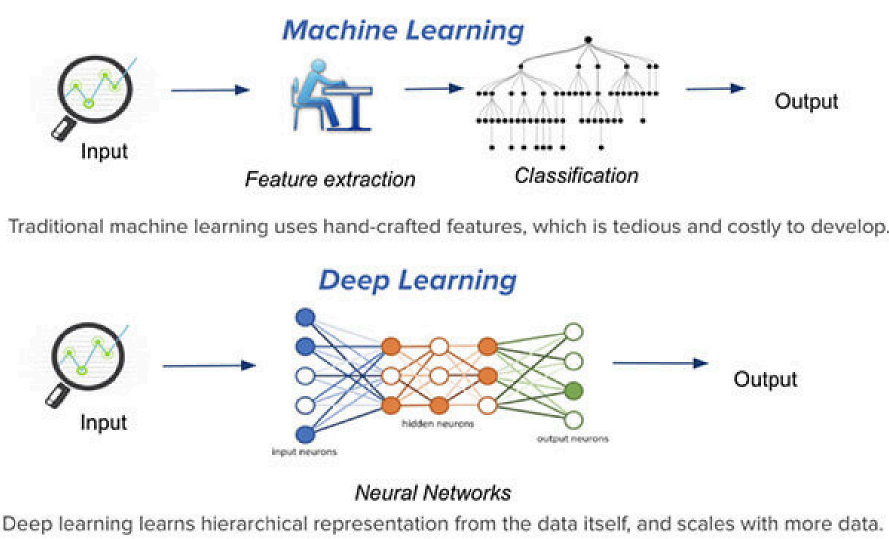
5 https://medium.com/@boutnaru/introduction-to-ai-artificial-intelligence-8c71d4d25320
6 https://www.ibm.com/topics/machine-learning
7 https://www.snowflake.com/guides/feature-extraction-machine-learning
8 https://ischoolonline.berkeley.edu/blog/what-is-machine-learning/
9 https://ml-ops.org/content/end-to-end-ml-workflow

Deep Learning

As stated in the "Introduction to Artificial Intelligence"[10], "Deep Learning" is a subset of "Machine Learning" - in the diagram below we can see a description on the difference between the two[11]. Moreover, "Deep Learning" leverages neural networks in order to try and achieve learning in the way in which it tries to simulate the behavior of the human brain (learning from large amounts of data). Usually it is done using neural networks with at least three layers[12]. By the way, more on neural networks in a future writeup.

Also, "Deep Learning" models allow us to learn complex patterns which are impossible/difficult to identify using traditional machine learning models. This is due to the fact "Deep Learning" models are able to learn hierarchical representations of the data, while also performing automatic feature extraction[13].

Thus, let us think about a "Deep Learning" model that is trained for identifying cars in an image. In this case the model might identify edges->shapes->objects->car. Thus, we can summarize that "Deep Learning" enables computational models that are based on multiple layers in order to learn representations of data with levels of abstractions. This is done by using backpropagation algorithms for tweaking the internal state of the model on every layer[14]. Lastly, "Deep Learning" is used in different fields such as fraud detection, medical diagnostics, speech recognition, computer vision, NLP (natural language processing), recommendation engines and more[15].



[10] https://medium.com/@boutnaru/introduction-to-ai-artificial-intelligence-8c71d4d25320
[11] https://www.merkle.com/blog/dispelling-myths-deep-learning-vs-machine-learning
[12] https://www.ibm.com/topics/deep-learning
[13] https://aws.amazon.com/what-is/deep-learning/
[14] https://www.nature.com/articles/nature14539
[15] https://aws.amazon.com/deep-learning/

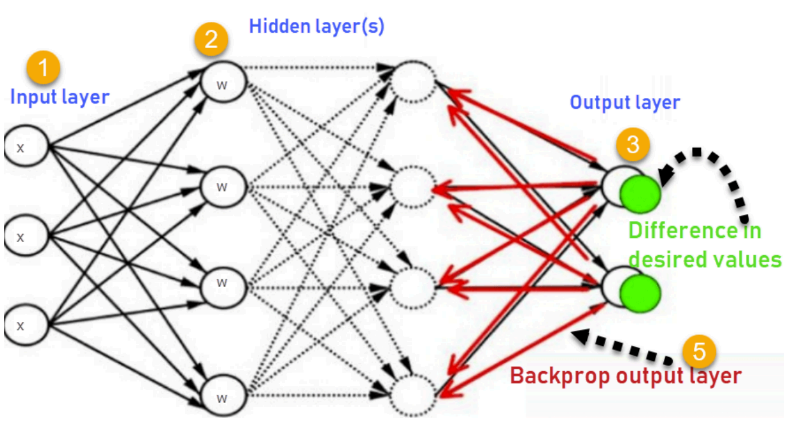
Neural Networks

A neural network is a learning system composed of interconnected nodes (aka neurons) that learn to perform tasks by processing data. Those networks are made up of layers of neurons. Each neuron in a layer [I] receives input from the neurons in layer [I-1]. Those neurons outputs a signal to the neurons in layer [I+1][16].

Moreover, the signals that are passed between neurons are weighted. Those weights are calibrated as the neural network learns more and more. An output of a neuron can be defined as y=x*W+b (x is the input, W is the weight and b is the bias). Bias is also a learnable parameter which is which stats the difference between the function's output and the intended data[17]. We can see a diagram of this description in the diagram below[18].

Overall, the learning process in a neural network is called backpropagation. By leveraging it the errors in the output (which are measured against the labels in the training data) of the neural network are propagated back through the network. Then weights of the neurons are calibrated accordingly based on the data. This process is repeated until the neural network is able to perform the task with a desired level of accuracy[19].

Lastly, we can say that the structure of a neural network depends on the number of hidden layers (determines the complexity of the model), the number of neurons in each layer (determines the amount of information the model can process) and the connections between those neurons (determines how the model learns). They are different types of neural networks like CNNs, RNNs, Deep neural networks[20]



[16] https://www.investopedia.com/terms/n/neuralnetwork.asp
[17] https://deepai.org/machine-learning-glossary-and-terms/weight-artificial-neural-network
[18] https://www.guru99.com/backpropogation-neural-network.html
[19] https://builtin.com/machine-learning/backpropagation-neural-network
[20] https://viso.ai/deep-learning/deep-neural-network-three-popular-types/

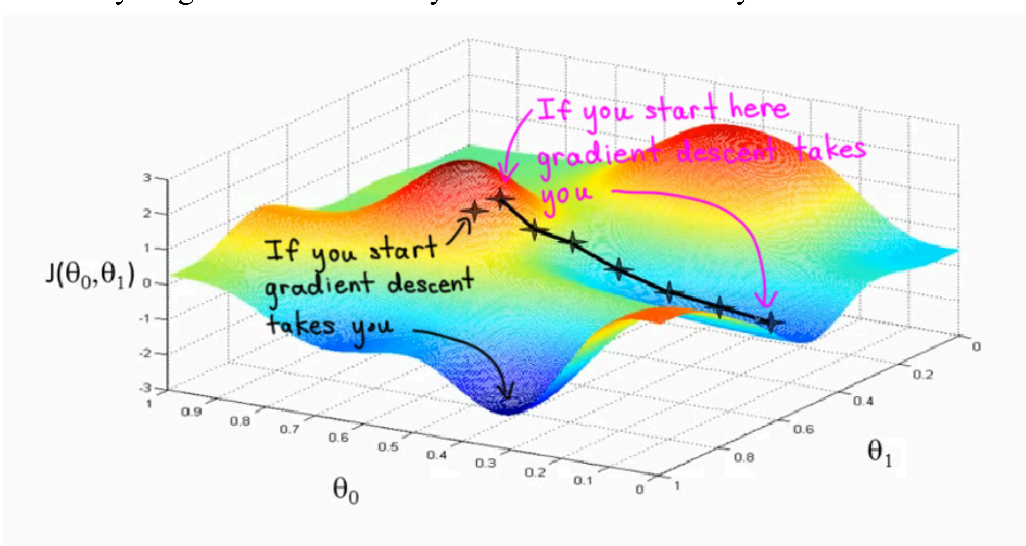
Gradient Descent (GD)

Gradient descent (GD) is an optimization algorithm that is focused on locating the local minimum/maximum of a specific function. It is done in an iterative manner which starts from a random point and then takes steps in the direction of the negative gradient of the function - as shown in the diagram below[21]. Moreover, gradient descent does not work for every function. There are two requirements for a function in order for GD to work: the function needs to be differentiable and convex[22].

Differentiable means that for every point in the function domain there is a derivative. Not all functions are like that, think about f(x)=1/x[23]. Convex is relevant for continuous functions, it means that for any two distinct points on the function's graph the line that connects them is above the graph between those points[24].

Overall, we can think of a gradient as a vector that points in the direction of the steepest ascent of the function. If we take steps in the opposite direction of the vector we are going to move in the direction of the nearest minimum point of the function[25].

Thus, we can use gradient descent in order to minimize the errors between the actual results and the expected results while training machine learning models. There are multiple variants of gradient descent like: Batch gradient descent, stochastic gradient descent, and mini-batch gradient descent[26] - More on them and others in future writeups. By the way, gradient descent was discovered by Augustin-Louis Cauchy in the mid 18th century[27].



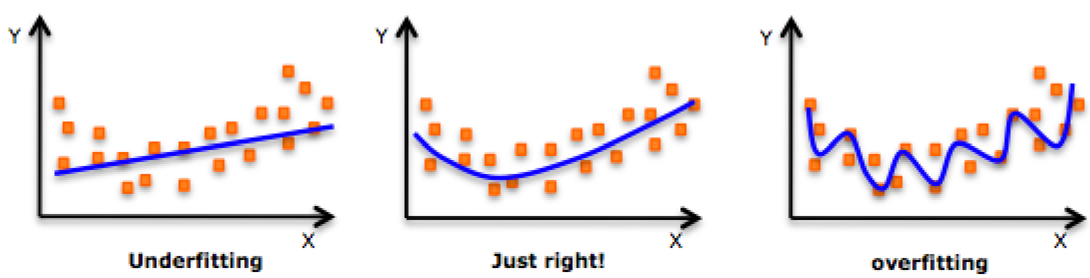
[21] https://regenerativetoday.com/machine-learning-gradient-descent-concept/
[22] https://towardsdatascience.com/gradient-descent-algorithm-a-deep-dive-cf04e8115f21
[23] https://math.mit.edu/~djk/calculus_beginners/chapter09/section03.html
[24] https://mathworld.wolfram.com/ConvexFunction.html
[25] https://www.uio.no/studier/emner/matnat/ifi/IN3050/v23/groups/gradient-descentascent.pdf
[26] https://www.javatpoint.com/gradient-descent-in-machine-learning
[27] https://www.deeplearning.ai/the-batch/gradient-descent-its-all-downhill/

Overfitting vs Underfitting

Overfitting is a case in which a machine learning model learns the training data too well and due to that it is unable to generalize on new data (not contained in the training data). So we can say that the statistical model performs great on the training data but it can't perform accurately against unseen data. We need to remember that generalization of a machine learning model to new data is what makes it relevant for classifying data/predicting results[28]. There are two main reasons for that: the training data is not representative of the real world and/or the model is too complex.

Underfitting is the case in which a machine learning model can't capture the relationship between the input and output variables. It generates a high error rate on real data (it can also happen with the training data). It can happen in case the size of the training dataset used is not enough, the model is too simple, the training data is not cleaned and more[29].

Lastly, as we can see both overfitting and underfitting are statistical problems we want to avoid when creating machine learning models. We can see an illustration of the different problems in the diagram below[30]. In the diagram the data points are shown as orange squares, while the model is represented as the blue line. In the next writeups we are going to talk about how to avoid overfitting and underfitting.



[28] https://www.ibm.com/topics/overfitting
[29] https://www.geeksforgeeks.org/underfitting-and-overfitting-in-machine-learning/
[30] https://www.analyticsvidhya.com/blog/2015/02/avoid-over-fitting-regularization/

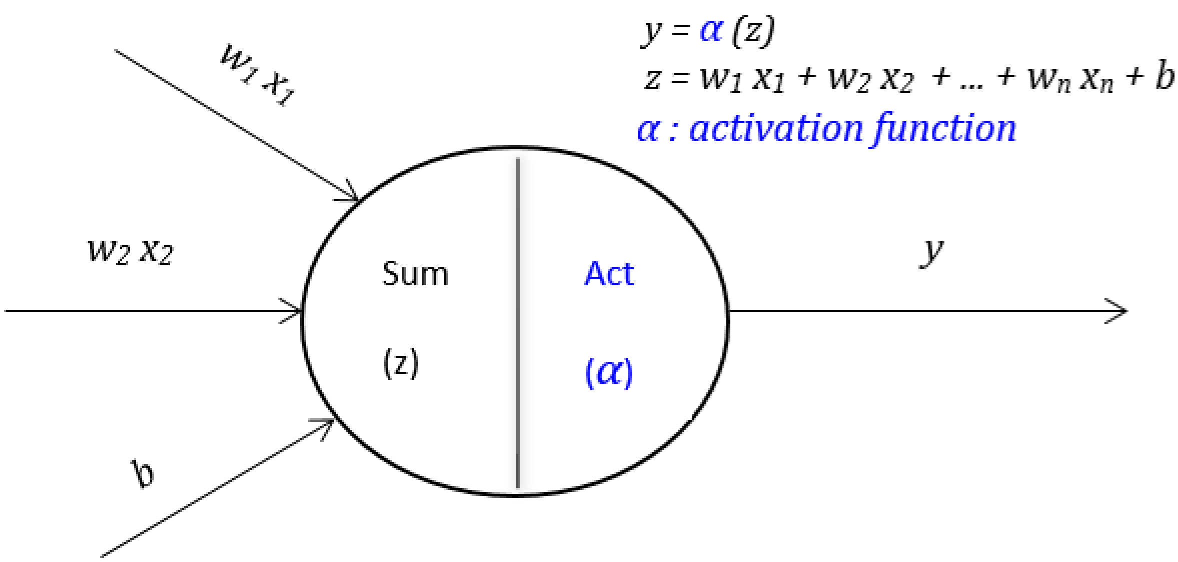
Activation Functions

An activation function is used by neural networks in order to compute the weighted sum of the input and biases - as shown in the diagram below. In some books those functions are called "Transfer Functions"[31]. We call them "Activation Functions" because they cause a specific neuron to be activated or not.

Moreover, activation functions can be linear or nonlinear. If we think about it, classical machine learning algorithms (like regression and SVM) were created based on linear models. However, not all real-life problems have a nature of non-linearity. Due to that we can get non-optimal results in case we use just linear based models[32].

Based on that understanding and the fact an activation function can be linear/non-linear we can use them to cope with the non-linearity of real-life problems. We just apply a non-linear function on the output of a specific layer before it is being propagated to the input of the next layer.

Lastly, there are different types of non-linear active functions that we can use like: Sigmoid, TanH, ReLU, Parametric ReLU and ELU .



[31] https://www.analyticsvidhya.com/blog/2021/04/activation-functions-and-their-derivatives-a-quick-complete-guide/
[32] https://www.exxactcorp.com/blog/Deep-Learning/activation-functions-and-optimizers-for-deep-learning-models