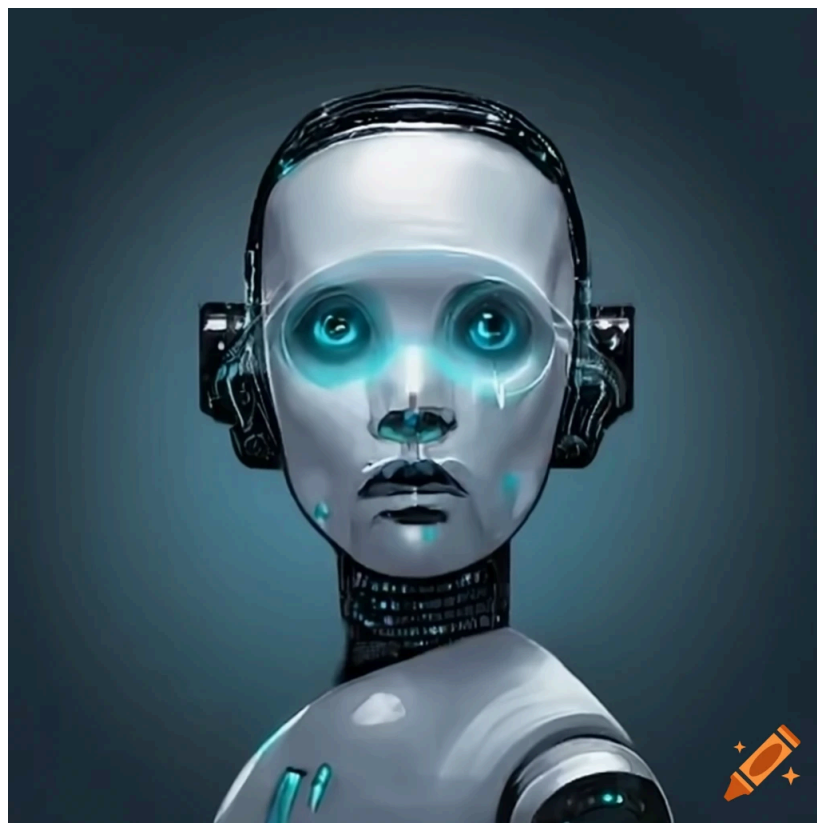


The Artificial Intelligence Journey

Version 2.0
February-2025

By Dr. Shlomi Boutnaru



Created using [Craiyon, AI Image Generator](#)

Table of Contents

Table of Contents.....	2
Introduction.....	5
Artificial Intelligence (AI).....	6
Machine Learning (ML).....	7
Supervised Learning.....	8
Unsupervised Learning.....	9
Semi-Supervised Learning.....	10
Reinforcement Learning.....	11
Deep Learning.....	12
Neural Networks.....	13
Loss Function.....	14
Gradient Descent (GD).....	15
Overfitting vs Underfitting.....	16
Activation Functions.....	17
Latent Space.....	18
PCA (Principal Component Analysis).....	19
Autoencoders.....	20
GenAI (Generative Artificial Intelligence).....	21
NLP (Natural Language Processing).....	22
LLM (Large Language Model).....	23
Base LLM (Base Large Language Model).....	24
Instruction Tuned LLM.....	25
LLM Temperature.....	26
SLM (Small Language Model).....	27
RAG (Retrieval-Augmented Generation).....	28
Context Window (aka Context Length).....	29
Llama (Large Language Model Meta AI).....	30

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>. Lastly, You can find my free eBooks at <https://TheLearningJourneyEbooks.com>.

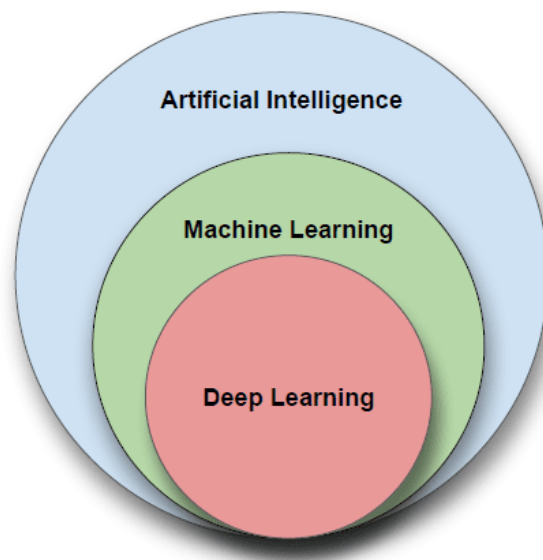
Lets GO!!!!!!

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 below¹.

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 AI³.

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 1956⁴.



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

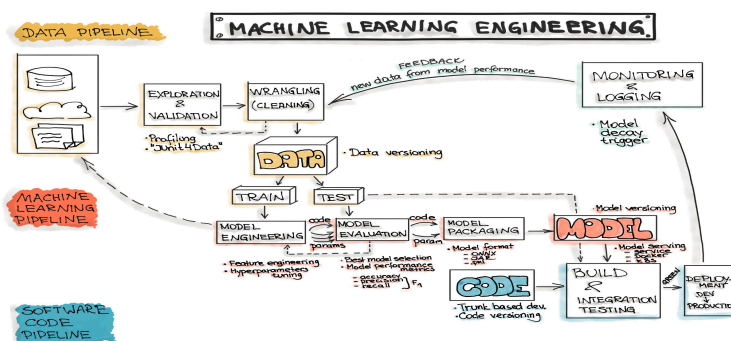
Machine Learning (ML)

Machine Learning is a subset/branch of artificial intelligence⁵ which leverages the use of statistical methods in order to train algorithms for making classifications and/or predictions. This provides the ability to uncover key insights in data mining projects. This can be done using different frameworks like PyTorch and TensorFlow⁶.

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 models⁷.

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 reward⁸. More on those types in future writeups. Lastly, there are different phases in the machine learning pipeline like: data collection, data cleaning, training a model, testing a model, deploying it and more - as shown in the diagram below⁹. 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/>

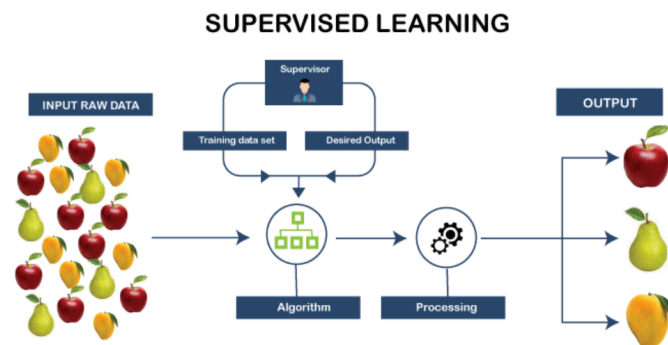
⁹ <https://ml-ops.org/content/end-to-end-ml-workflow>

Supervised Learning

“Supervised Learning” is one of the four types of machine learning¹⁰. In supervised learning we use labeled data for training AI¹¹ models in order to identify the underlying patterns and relationships between input features and outputs. By doing so we can create a model that predicts correct outputs on new real-world data¹².

Overall, we can use supervised learning for both classification and regression¹³. Classification is when the model tries to predict the correct label of the input data. Thus, the target variable is discrete¹⁴ - as shown in the diagram below¹⁵. Regression is when the goal of the model is to predict continuous numerical value based on one or more independent features. This is done by finding relationships between the input variables¹⁶.

Lastly, deep learning can be both supervised or unsupervised¹⁷. Examples of supervised learning algorithms are: "Naive Bayes"¹⁸, “Linear Regression”¹⁹, “Support Vector Machine”²⁰, “Random Forest”²¹ and “K-nearest Neighbor”²².



¹⁰ <https://medium.com/@boutnaru/introduction-to-machine-learning-9b488f02efb9>

¹¹ <https://medium.com/@boutnaru/introduction-to-ai-artificial-intelligence-8c71d4d25320>

¹² <https://www.ibm.com/think/topics/supervised-learning>

¹³ https://scikit-learn.org/stable/supervised_learning.html

¹⁴ <https://www.datacamp.com/blog/classification-machine-learning>

¹⁵ <https://mungfali.com/explore/Supervised-Learning-Logo>

¹⁶ <https://www.geeksforgeeks.org/regression-in-machine-learning/>

¹⁷ <https://levity.ai/blog/difference-machine-learning-deep-learning>

¹⁸ https://scikit-learn.org/stable/modules/naive_bayes.html

¹⁹ <https://www.analyticsvidhya.com/blog/2021/10/everything-you-need-to-know-about-linear-regression>

²⁰ <https://www.geeksforgeeks.org/support-vector-machine-algorithm/>

²¹ <https://www.analyticsvidhya.com/blog/2021/06/understanding-random-forest/>

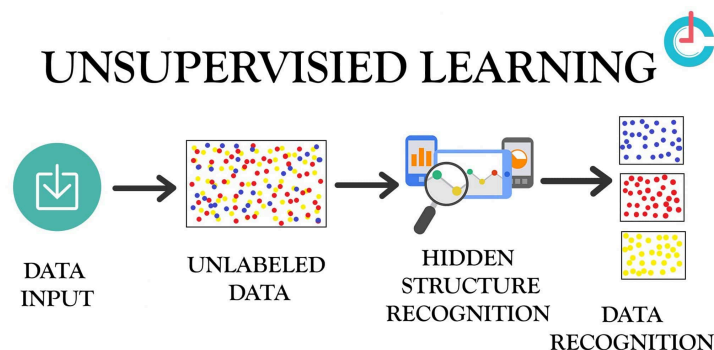
²² https://www.w3schools.com/python/python_ml_knn.asp

Unsupervised Learning

“Unsupervised Learning” is one of the four types of machine learning²³. Unsupervised learning allows learning from data without any human supervision. They are given as input unlabeled data in order to discover patterns and insights - as shown below²⁴. AI models based on unsupervised learning are used in different use cases such as: genetic research, NLP (Natural Language Processing), fraud detection, recommendation engines, anomaly detection and customer segmentation²⁵.

Overall, there are three types of algorithms usually used in case of unsupervised learning: clustering (groups unlabeled data into clusters based on their similarities), dimensionality reduction (reducing the number of features in a dataset while preserving as much information as possible) and association rule learning²⁶ - more on those in future writeups.

Lastly, we have different examples of unsupervised learning algorithms. Examples of clustering algorithms are: “K-means”, “Fuzzy K-means” and GMMs (Gaussian Mixture Models). In the field of association rules we can use algorithms like: “Apriori” and “FP-Growth”²⁷. PCA (Principal Component Analysis) is an example of dimensionality reduction algorithm²⁸.



²³ <https://medium.com/@boutnaru/introduction-to-machine-learning-9b488f02efb9>

²⁴ <https://mungfali.com/explore/Unsupervised-Learning-Block-Diagram>

²⁵ <https://cloud.google.com/discover/what-is-unsupervised-learning>

²⁶ <https://www.geeksforgeeks.org/unsupervised-learning/>

²⁷ <https://www.altexsoft.com/blog/unsupervised-machine-learning/>

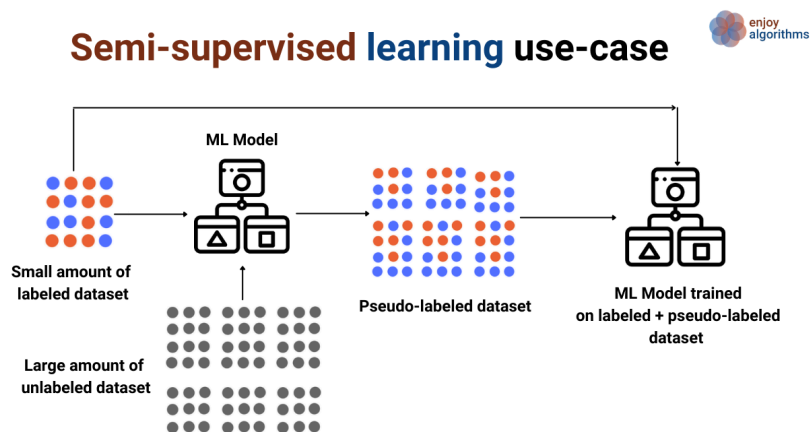
²⁸ <https://medium.com/@boutnaru/the-artificial-intelligence-journey-pca-principal-component-analysis-80b4d12b2cd2>

Semi-Supervised Learning

“Semi-Supervised Learning” is one of the four types of machine learning²⁹. In semi-supervised learning part of our training data is not labeled. Semi-supervised learning algorithms can perform well even if we have a small amount of labeled data together with a large amount of unlabeled data points³⁰ - as shown in the diagram below³¹.

Overall, there are different pros and cons in regards to semi-supervised learning. Among the pros are: improved model performance, effectiveness for unstructured data (like video and audio) and it is less expensive due to the fact it reduces the need for manual labeling. However, there are also cons such as: sensitivity to the data quality of the labeled data, limited transparency in regards with the reasons which lead to the predictions and less suited to complex/diverse data sets³².

Lastly, there are several approaches for semi-supervised learning: self-training, co-training, multi-view training and SSL using graph models - more on those in future writeups. There are different applications in which semi-supervised learning is leveraged like: document classification, image classification, speech recognition, face recognition, OCR (Optical Character Recognition) and handwritten text recognition³³.



²⁹ <https://medium.com/@boutnaru/introduction-to-machine-learning-9b488f02efb9>

³⁰ https://scikit-learn.org/stable/modules/semi_supervised.html

³¹ <https://www.enjoyalgorithms.com/blogs/supervised-unsupervised-and-semisupervised-learning/>

³² <https://www.oracle.com/ba/artificial-intelligence/machine-learning/semi-supervised-learning/>

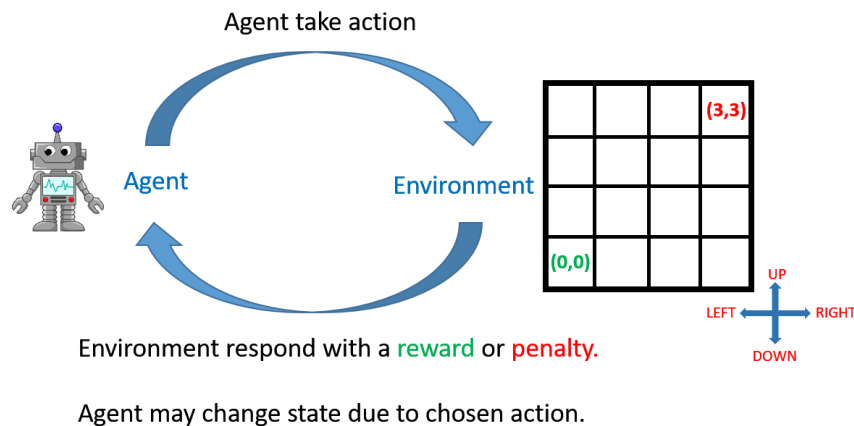
³³ <https://maddevs.io/blog/semi-supervised-learning-explained/>

Reinforcement Learning

“Reinforcement Learning” (RL) is one of the four types of machine learning³⁴. As part of reinforcement learning we mimic the “trial and error” learning experience human use for learning. Thus, actions which work towards the goal are reinforced while actions that detract from the goal are ignored. This means that reinforcement learning uses a “reward and punishment” technique as data is processed³⁵.

Overall, in reinforcement learning we have an “agent” that learns to make decisions by interacting with an environment (the problem space). The agent has a “state” that is the situation of the agent in the environment. Due to that each action affects the state. Based on the action of the agent a feedback is given to the agent (reward\penalty). Also, the agent has a policy which is a strategy it has to decide on actions at each state³⁶ - as shown below³⁷.

Lastly, there are two methods by which an agent can collect data for learning: “Online” and “Offline”. In the first (online) one the agent collects data directly from interacting with its surrounding environment. In the second (offline), agent does not have direct access to an environment, so it can learn using logged data from the relevant environment. By the way, we can use different approaches for RL like: dynamic programming, monte carlo, and temporal difference learning³⁸.



³⁴ <https://medium.com/@boutnaru/introduction-to-machine-learning-9b488f02efb9>

³⁵ <https://aws.amazon.com/what-is/reinforcement-learning/>

³⁶ <https://saturncloud.io/glossary/reinforcement-learning-environments/>

³⁷ <https://morioh.com/a/eeaebec8872/a-simple-reinforcement-learning-environment-from-scratch>

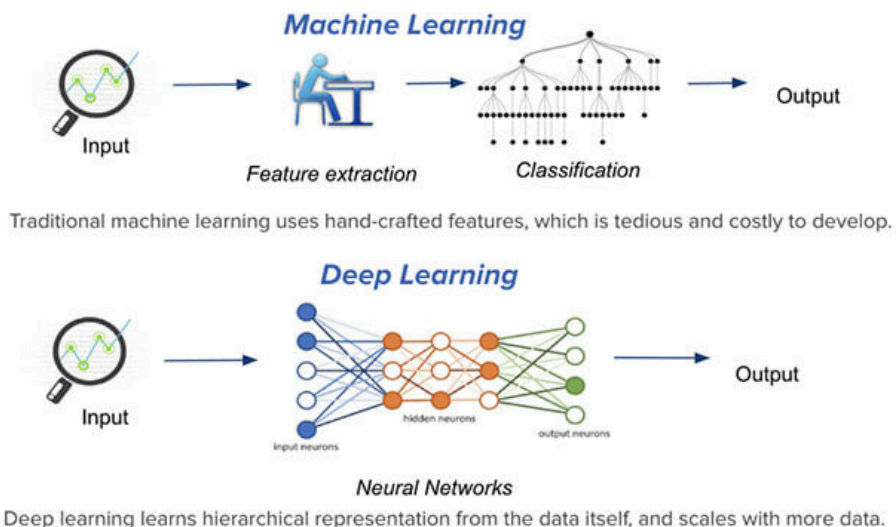
³⁸ <https://www.ibm.com/think/topics/reinforcement-learning>

Deep Learning

As stated in the “Introduction to Artificial Intelligence”³⁹, “Deep Learning” is a subset of “Machine Learning” - in the diagram below we can see a description on the difference between the two⁴⁰. 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 layers⁴¹. 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 extraction⁴².

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 layer⁴³. 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 more⁴⁴.



³⁹ <https://medium.com/@boutnaru/introduction-to-ai-artificial-intelligence-8c71d4d25320>

⁴⁰ <https://www.merkle.com/blog/dispelling-myths-deep-learning-vs-machine-learning>

⁴¹ <https://www.ibm.com/topics/deep-learning>

⁴² <https://aws.amazon.com/what-is/deep-learning/>

⁴³ <https://www.nature.com/articles/nature14539>

⁴⁴ <https://aws.amazon.com/deep-learning/>

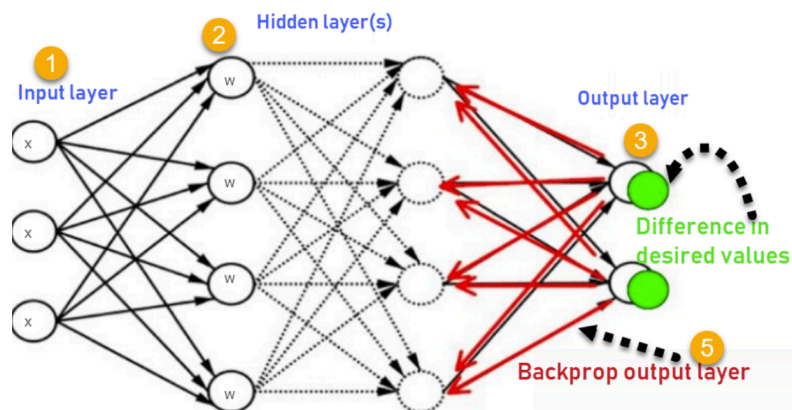
Neural Networks

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]⁴⁵.

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 data⁴⁶. We can see a diagram of this description in the diagram below⁴⁷.

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 accuracy⁴⁸.

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 networks⁴⁹



⁴⁵ <https://www.investopedia.com/terms/n/neuralnetwork.asp>

⁴⁶ <https://deepai.org/machine-learning-glossary-and-terms/weight-artificial-neural-network>

⁴⁷ <https://www.guru99.com/backpropagation-neural-network.html>

⁴⁸ <https://builtin.com/machine-learning/backpropagation-neural-network>

⁴⁹ <https://viso.ai/deep-learning/deep-neural-network-three-popular-types/>

Loss Function

Basically, loss function is a term used in math optimization/decision theory. It is sometimes called cost function, fitness function or even error function. The goal of this function is mapping an event/value(s) of a variable(s) to a real number, which represents the “cost” of the event. Thus, the loss function helps us in evaluating how well our algorithm is modeling the data set⁵⁰.

Moreover, by using a loss function we can define a learning problem as an optimization problem - as shown in the diagram below⁵¹. If we choose the correct loss function the learning phase can omit a good outcome⁵².

Lastly, there are different loss functions which are commonly used as part of deep learning algorithms - following are a couple of examples. With regression example loss functions are: MSE (Mean square error), MAE (Mean Absolute Error) and Hubber loss. With classification examples loss functions are: binary cross entropy and categorical cross-entropy. Autoencoders can use KL divergence. Object detection can leverage and word embedding can use triplet loss. GAN can use a loss function discriminator loss or minmax GAN loss⁵³.

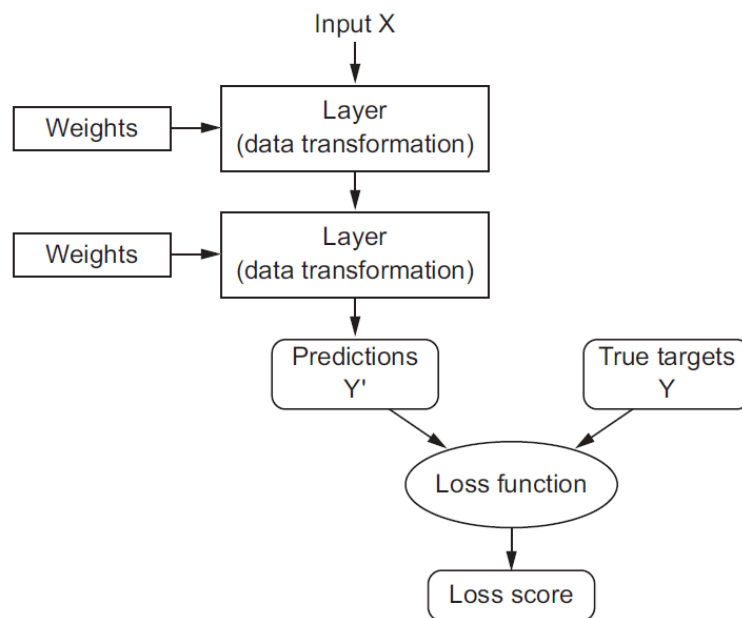


Figure 1.8 A loss function measures the quality of the network’s output.

⁵⁰ https://en.wikipedia.org/wiki/Loss_function

⁵¹ <https://medium.com/howtoai/playing-with-loss-functions-in-deep-learning-26faf29c85f>

⁵² <https://c3.ai/glossary/data-science/loss-function/>

⁵³ <https://www.analyticsvidhya.com/blog/2022/06/understanding-loss-function-in-deep-learning/>

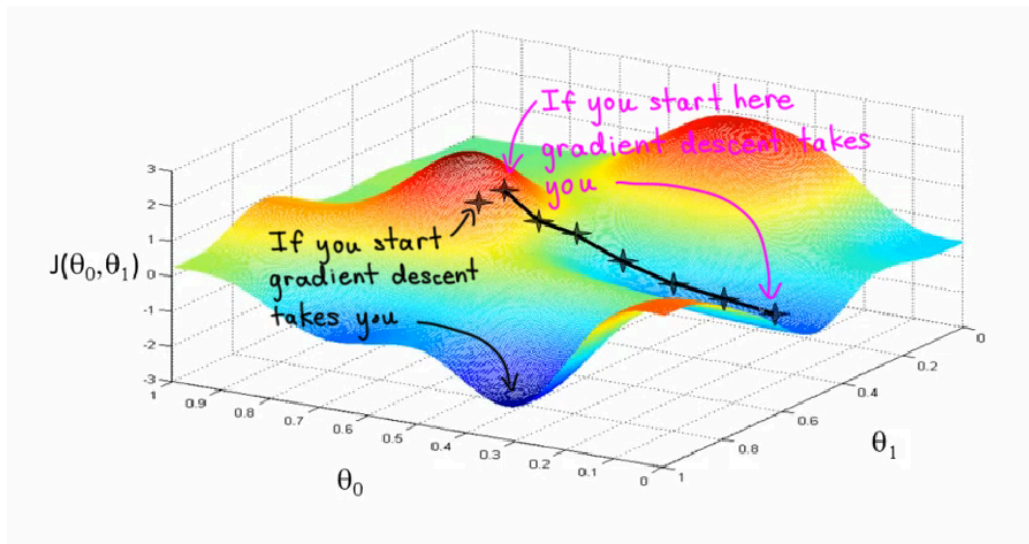
Gradient Descent (GD)

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 below⁵⁴. 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 convex⁵⁵.

Differentiable means that for every point in the function domain there is a derivative. Not all functions are like that, think about $f(x)=1/x$ ⁵⁶. 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 points⁵⁷.

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 function⁵⁸.

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 descent⁵⁹ - More on them and others in future writeups. By the way, gradient descent was discovered by Augustin-Louis Cauchy in the mid 18th century⁶⁰.



⁵⁴ <https://regenerativetoday.com/machine-learning-gradient-descent-concept/>

⁵⁵ <https://towardsdatascience.com/gradient-descent-algorithm-a-deep-dive-cf04e8115f21>

⁵⁶ https://math.mit.edu/~djg/calculus_beginners/chapter09/section03.html

⁵⁷ <https://mathworld.wolfram.com/ConvexFunction.html>

⁵⁸ <https://www.uio.no/studier/emner/matnat/ifi/IN3050/v23/groups/gradient-descentascent.pdf>

⁵⁹ <https://www.javatpoint.com/gradient-descent-in-machine-learning>

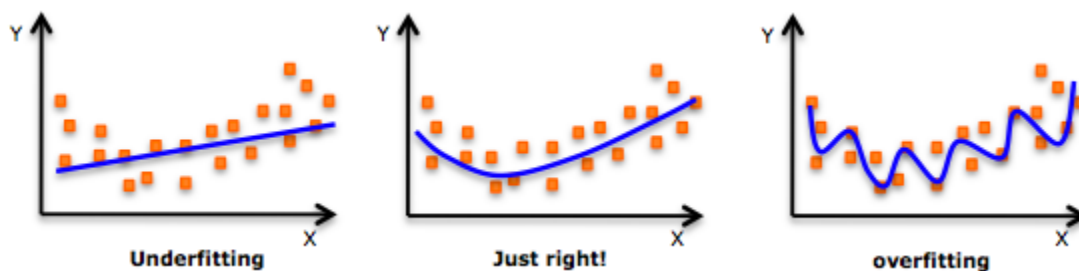
⁶⁰ <https://www.deeplearning.ai/the-batch/gradient-descent-its-all-downhill/>

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 results⁶¹. 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 more⁶².

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 below⁶³. 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.



⁶¹ <https://www.ibm.com/topics/overfitting>

⁶² <https://www.geeksforgeeks.org/underfitting-and-overfitting-in-machine-learning/>

⁶³ <https://www.analyticsvidhya.com/blog/2015/02/avoid-over-fitting-regularization/>

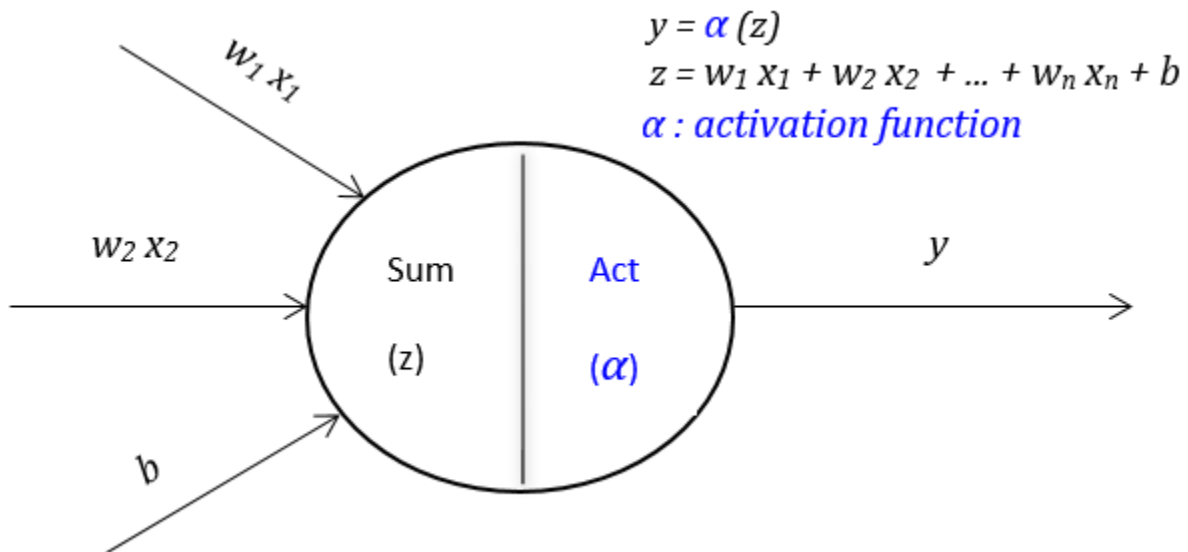
Activation Functions

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”⁶⁴. 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 models⁶⁵.

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 .



⁶⁴ <https://www.analyticsvidhya.com/blog/2021/04/activation-functions-and-their-derivatives-a-quick-complete-guide/>

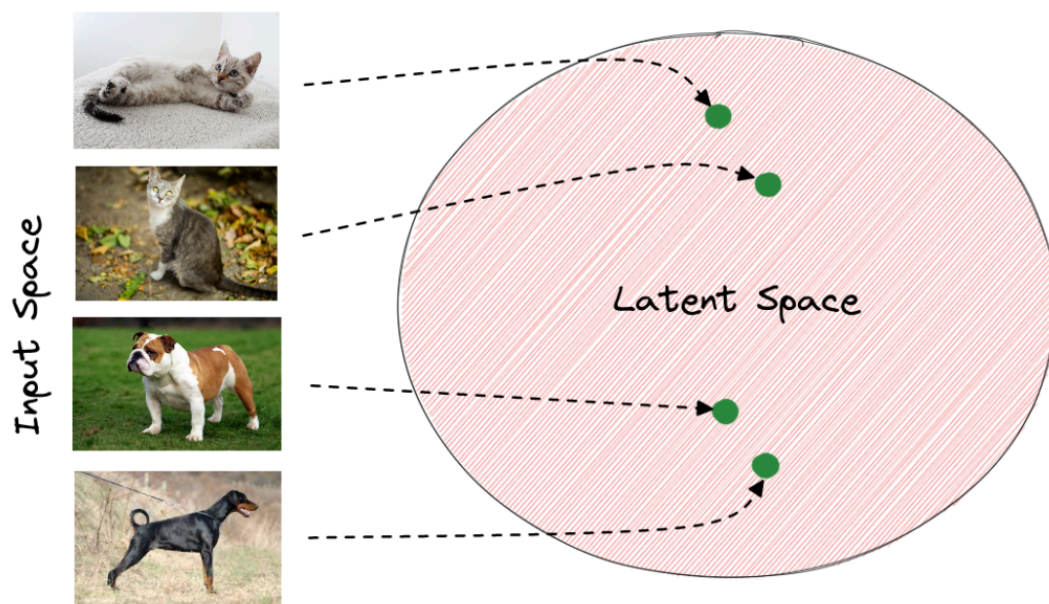
⁶⁵ <https://www.exxactcorp.com/blog/Deep-Learning/activation-functions-and-optimizers-for-deep-learning-models>

Latent Space

A latent space provides a lower-dimensional representation designed to capture the underlying structure and relationships within that data. It is used in different areas such as neuroscience, data analysis and deep learning. By the way, it is also known as embedding space or as a latent feature space. We can say that by using a latent space the input data is organized and mapped in a way where similar things are positioned closer together - as shown in the diagram below⁶⁶.

Thus, using a latent space we can perform dimension reduction and identify similarity and relationship between different inputs. We can construct a latent space using different linear/nonlinear dimensionality reduction techniques, such as Principal Component Analysis (PCA), t-Distributed Stochastic Neighbor Embedding (t-SNE), and autoencoders⁶⁷.

Lastly, latent space is used in clustering⁶⁸, dimensionality reduction⁶⁹, GAN⁷⁰, feature extraction⁷¹, image recognition⁷², anomaly detection⁷³ and more. .



⁶⁶ <https://www.baeldung.com/cs/dl-latent-space>

⁶⁷ <https://saturncloud.io/glossary/latent-space/>

⁶⁸ <https://ojs.aaai.org/index.php/AAAI/article/view/4385>

⁶⁹ <https://cdn.aaai.org/AAAI/2008/AAAI08-108.pdf>

⁷⁰ <https://proceedings.mlr.press/v119/voynov20a.html>

⁷¹ <https://www.mdpi.com/2076-3425/12/10/1348>

⁷² <https://ieeexplore.ieee.org/abstract/document/8545506/>

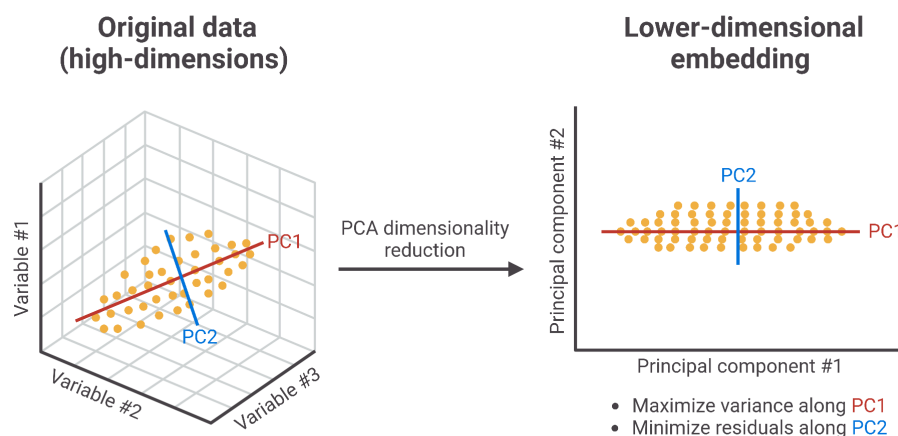
⁷³ <https://link.springer.com/article/10.1007/s10994-022-06153-4>

PCA (Principal Component Analysis)

The goal of PCA (Principal Component Analysis) is to perform dimensionality reduction - as shown in the diagram below⁷⁴. By using PCA we can simplify a large data set into a smaller set while still maintaining significant patterns and trends⁷⁵.

Thus, PCA is usually used as a data preprocessing phase with machine learning algorithms. By doing so we reduce the model complexity, because the addition of each new feature negatively impacts model performance (“curse of dimensionality”). The idea is to project a high-dimensional dataset into a smaller feature space⁷⁶.

Lastly, as opposed to LDA (Linear Discriminant Analysis) we can use PCA both for supervised learning tasks and unsupervised learning⁷⁷. PCA is based on different concepts from linear algebra such as: eigenvalues, eigenvectors and covariance matrix.



⁷⁴ <https://www.biorender.com/template/principal-component-analysis-pca-transformation>

⁷⁵ <https://builtin.com/data-science/step-step-explanation-principal-component-analysis>

⁷⁶ <https://www.ibm.com/topics/principal-component-analysis>

⁷⁷ <https://www.ibm.com/topics/linear-discriminant-analysis>

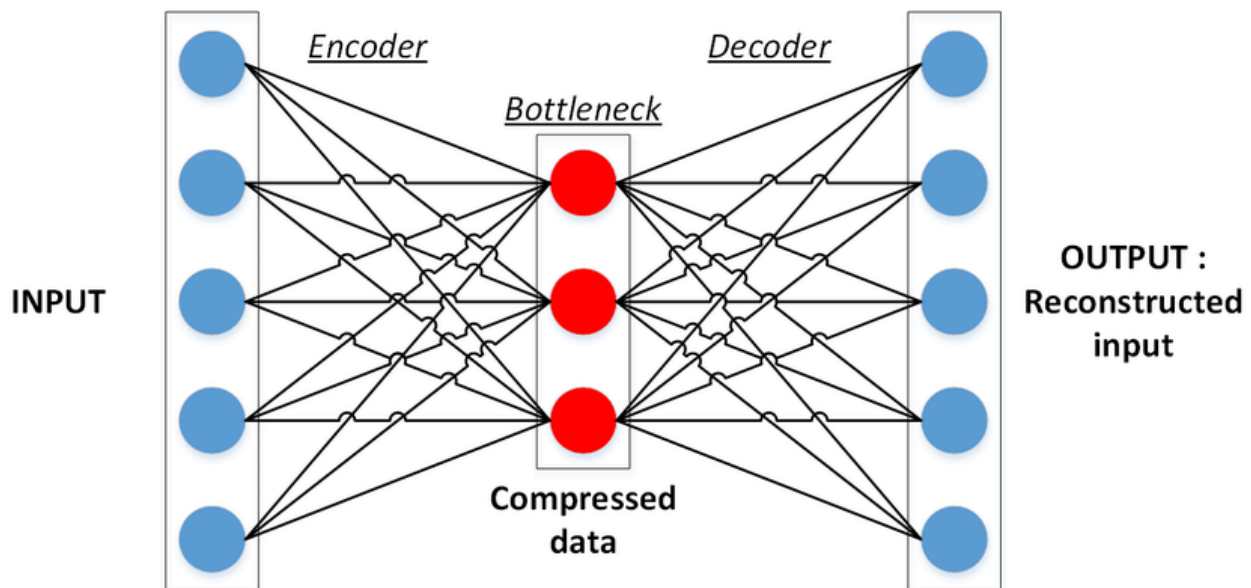
Autoencoders

In general, autoencoders are a specific type of a neural network⁷⁸ which can be used for dimension reduction by learning an efficient coding of unlabeled data. We can say it is trained to compress the input data (to a lower dimension) and then reconstructing the data in order to match it to the original input as closely as possible⁷⁹.

Thus, the architecture of an autoencoder is composed of three layers: encoder, code (aka the bottleneck) and decoder - as shown in the diagram below⁸⁰. The encoder layer compresses the input data into a latent-space representation. The code layer represents the compressed input fed to the decoder layer. The decoder layer reconstructs the input data from the latent space representation⁸¹.

Moreover, the goal is to train autoencoders to minimize the reconstruction error. This is done by using a loss/fitness/error function⁸² an example is MSE (mean squared error) between the input data and the reconstructed data. The weights of the neural network are adjusted during the training phase to minimize the loss function⁸³ mostly using backpropagation.

Lastly, there are different types of autoencoders such as: sparse autoencoders, contractive autoencoders, undercomplete autoencoders, denoising autoencoders and variational autoencoders⁸⁴.



⁷⁸ <https://medium.com/@boutnaru/introduction-to-neural-networks-f65bf17afd2c>

⁷⁹ <https://deepai.org/machine-learning-glossary-and-terms/autoencoder>

⁸⁰ <https://stackabuse.com/autoencoders-for-image-reconstruction-in-python-and-keras/>

⁸¹ <https://www.simplilearn.com/tutorials/deep-learning-tutorial/what-are-autoencoders-in-deep-learning>

⁸² <https://medium.com/@boutnaru/the-artificial-intelligence-journey-loss-function-3534db6c0d69>

⁸³ <https://www.v7labs.com/blog/autoencoders-guide>

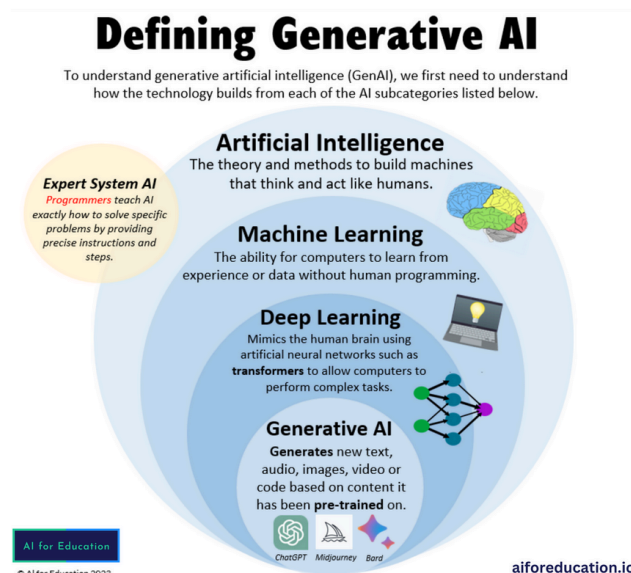
⁸⁴ <https://www.v7labs.com/blog/autoencoders-guide>

GenAI (Generative Artificial Intelligence)

GenAI (Generative Artificial Intelligence) in the field of artificial intelligence⁸⁵. It uses “Deep Learning”⁸⁶ models which receive as input raw data (think about all wikipedia as an example and much more) and “learn” how to generate statistically probable outputs when prompted⁸⁷ - as also explained in the diagram below⁸⁸. Well known examples of GenAI applications are: “ChatGPT”, “DALL-E” and “Google Gemini” which was previously called “Bard”⁸⁹.

Overall, there are multiple types of generative AI models like diffusion models. Examples of those are VAEs (Variational Autoencoders) and GANs (Generative Adversarial Networks). Also, there are different architectures for GenAI models, probably the most used one is a “Transformer Network”⁹⁰.

Lastly, generative AI is able to create new content like text, images, video and speech based on the patterns of data learned by the models powering it. Another type of models that can be used for generative AI: LLMs (Large Language Models), SLMs (Small Language Models), RAG (Retrieval-Augmented Generation) - they can also be combined with those mentioned above. Also, by using “AI Agents” we can autonomously perform different tasks by using generative AI⁹¹.



⁸⁵ <https://medium.com/@boutnaru/introduction-to-ai-artificial-intelligence-8c71d4d25320>

⁸⁶ <https://medium.com/@boutnaru/introduction-to-deep-learning-433680bfafef>

⁸⁷ <https://research.ibm.com/blog/what-is-generative-AI>

⁸⁸ <https://www.aiforeducation.io/ai-resources/generative-ai-explainer>

⁸⁹ <https://www.coursera.org/articles/generative-ai-applications>

⁹⁰ <https://www.nvidia.com/en-eu/glossary/generative-ai/>

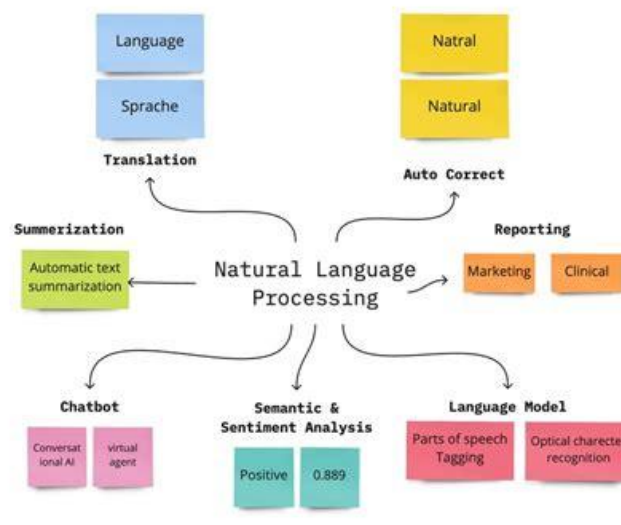
⁹¹ <https://generativeai.net/>

NLP (Natural Language Processing)

NLP (Natural Language Processing) is a realm of AI⁹² which is used for helping computer systems analyze/understand/respond to humans using speech and written text. NLP is a part of “computational linguistics” which merges between AI, computer science and linguistics. Its goal is studying the computational aspects of human language⁹³.

Overall, there are different approaches for NLP. “Rules Based NLP” the earlier form, we can think about that as if-then decision trees based on programming rules an example is the original version of “Moviefone”. “Statistical NLP” which maps language elements (words and/or grammatical rules) to a vector representation. Then that language can be modeled by using statistical methods (like regression or Markov models). ”Deep Learning NLP” leverages volumes of unstructured data (voice and text) in order to be more accurate. Examples of such models are: “seq2seq” (Sequence-to-Sequence), “Transformer Models” and “Autoregressive Models”⁹⁴.

Lastly, we could use NLP for different use case such as machine translation, spam detection, sentiment analysis, text simplifications and more - as also described in the diagram below⁹⁵. Real world examples for those are: “Google Translate”, “Grammarly”, “Alexa” and “OK Google”⁹⁶.



⁹² <https://medium.com/@boutnaru/introduction-to-ai-artificial-intelligence-8c71d4d25320>

⁹³ <https://www.elastic.co/what-is/natural-language-processing>

⁹⁴ <https://www.ibm.com/topics/natural-language-processing>

⁹⁵ https://www.researchgate.net/figure/Natural-Language-Processing_fig1_364842359

⁹⁶ <https://k21academy.com/datascience-blog/machine-learning/natural-language-processing/>

LLM (Large Language Model)

LLM (Large Language Model) is an AI⁹⁷ application which can be leveraged (among other things) to recognize and generate text. Those models are trained on huge data sets (so they have enough examples to recognize/interpret human language) using a type of neural network called a “transformer model”⁹⁸.

Overall, the goal of an LLM is to understand the relationships between characters, words and sentences. We can use LLMs to generate/translate text and perform different NLP⁹⁹ tasks. Thus, we can leverage LLMs from many use cases like: chatbots, document summarization, AI-powered contact center and more¹⁰⁰. We can compare between various LLMs based on attributes like size, code license, use and more - as shown in table below¹⁰¹.

Lastly, from all the above we can say the LLMs are a subcategory of AI. They are focused on understanding, predicting, and generating human-like text. Because of that, LLMs are a fundamental piece of GenAI¹⁰². Also, large language models come with different challenges such as: privacy concerns, technical complexity, business dependency and continuity, ethical concerns regarding bias and fairness, misinterpretation of data and more¹⁰³.

Comparing different LLMs									
Model	Size	Use	Training code available	Inference code available	Finetuning code available	Code license	Weights license	Instruction-tuned / foundation model	Backbone
Bloomz	176B	Restricted applications	✗	✓	✗	Responsible AI (OpenRail)	Responsible AI (OpenRail)	Instruction-tuned	Bloom
Chat GPT (gpt-3.5-turbo)	175B	Paid API	✗	✗	✗	Public Web API	Public Web API	Instruction-tuned	GPT3
Dolly-V2	1B	Commercial	✗	✗	✗	Apache License 2.0	Apache License 2.0	Instruction-tuned	Pythia
Lit-LLaMA	7B	Non-commercial research	✓	✓	✓	Apache License 2.0	Non-commercial research	Foundation model	—
Lit-LLaMA + Alpaca	7B	Non-commercial research	✓	✓	✓	Apache License 2.0	Non-commercial research	Instruction-tuned	LLaMA

⁹⁷ <https://medium.com/@boutnaru/introduction-to-ai-artificial-intelligence-8c71d4d25320>

⁹⁸ <https://www.cloudflare.com/learning/ai/what-is-large-language-model/>

⁹⁹ <https://medium.com/@boutnaru/the-artificial-intelligence-journey-nlp-natural-language-processing-41ae7d1d4428>

¹⁰⁰ <https://cloud.google.com/ai/llms?hl=en>

¹⁰¹ <https://muhtasham.github.io/blog/posts/llm-bootcamp/>

¹⁰² <https://medium.com/@boutnaru/the-artificial-intelligence-journey-genai-generative-artificial-intelligence-29d1228e905e>

¹⁰³ <https://www.sap.com/resources/what-is-large-language-model>

Base LLM (Base Large Language Model)

A “Base LLM” (Base Large Language Model) is an LLM¹⁰⁴ in the field of “Generative AI”¹⁰⁵. They represent the fundamental model which had been obtained due the training on large volumes of text from the Internet. Examples of such models are: “GPT-3”, “GPT-4” and “BERT”¹⁰⁶.

Overall, “Base LLMs” are great for understanding and predicting language patterns but they don’t work well in following instructions provided by prompts (as opposed to “Instruction-tuned LLMs” - more on those in future writeups). Due to those reasons “Base LLMs” are mostly relevant for applications in the following areas: general knowledge responses, translations, summarization and more¹⁰⁷. By the way, “Base LLMs” are sometimes called pre-trained LLMs¹⁰⁸.

Lastly, we can summarize that “Base LLMs” are designed to predict the next word based on their training data¹⁰⁹ - as shown below¹¹⁰. For better understanding let us think about the next example. If we prompt “What is the capital of France?” we could get a response like “What is France's largest city?” or “What is France's population?” from the “Base LLM”¹¹¹.

Eric: Hi Tom! How are
Eric: Hi Tom! How are you?
Eric: Hi Tom! How are you? Tom:
Eric: Hi Tom! How are you? Tom: Sorry,
Eric: Hi Tom! How are you? Tom: Sorry, who
Eric: Hi Tom! How are you? Tom: Sorry, who are
Eric: Hi Tom! How are you? Tom: Sorry, who are you
Eric: Hi Tom! How are you? Tom: Sorry, who are you again?

Several iterations of an LLM predicting the next word of a sentence

¹⁰⁴ <https://medium.com/@boutnaru/the-artificial-intelligence-llm-large-language-model-f3e1a3fb15d6>

¹⁰⁵ <https://medium.com/@boutnaru/the-artificial-intelligence-journey-genai-generative-artificial-intelligence-29d1228e905e>

¹⁰⁶ <https://oliviermills.com/articles/understanding-ai-models-base-language-learning-models-vs-instruction-tuned-language-learning-models>

¹⁰⁷ <https://toloka.ai/blog/base-llm-vs-instruction-tuned-llm/>

¹⁰⁸ <https://whymlabs.ai/learning-center/introduction-to-llms/training-and-fine-tuning-large-language-models>

¹⁰⁹ <https://roadmap.sh/prompt-engineering/basic-llm/llm-types>

¹¹⁰ <https://www.gravity-testing.com/blog/mastering-llm-agent-the-testers-essential-guide/>

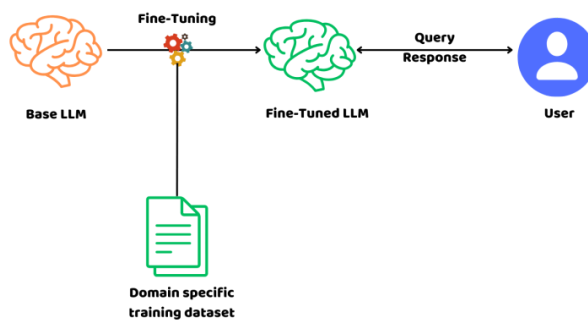
¹¹¹ <https://learnius.com/llms/2+LLMs+and+Transformers/base+LLM>

Instruction Tuned LLM

“Instruction Tuned LLM” is a LLM¹¹² built on top of “Base LLMs”¹¹³. However, instead of just trying re predictive (autocomplete) text they try to follow the given instructions using the data that they have been trained on. Thus, we can start with a “Base LLM” and expend the training using a large dataset covering sample “Instructions” and how the model should react as a result of those instructions. Then we fine-tune the process - as shown in the diagram below¹¹⁴. The fine-tuned model can leverage “RLHF” (“Reinforcement Learning with Human Feedback”). By using RLHF the model learns from human feedback and improves its performance¹¹⁵.

Overall, instruction tuned LLMs provide different benefits like: improved instruction following, better handling complex requests, task specialization, responsive to tune and style, better understanding of user intent and more. Examples of applications are: business and productivity tools, customer support and virtual assistance and creative writing and content ideas¹¹⁶.

Lastly, there are different open source human created instruction datasets like: Flan, OpenAssistant and Dolly. There are also LLM generated datasets (due to the prohibitive amount of cost and labor required to manually generate instructions) such as: Self-Instruct, Evol-Instruct and OpenOrca¹¹⁷.



Fine-Tuning Process

¹¹² <https://medium.com/@boutnaru/the-artificial-intelligence-llm-large-language-model-f3e1a3fb15d6>

¹¹³ <https://medium.com/@boutnaru/the-artificial-intelligence-journey-base-llm-base-large-language-model-726423106b14>

¹¹⁴ <https://crucialbits.com/blog/llm-customizations-prompt-engineering-rag-fine-tuning/>

¹¹⁵ <https://roadmap.sh/prompt-engineering/basic-llm/llm-types>

¹¹⁶ <https://toloka.ai/blog/base-llm-vs-instruction-tuned-llm/>

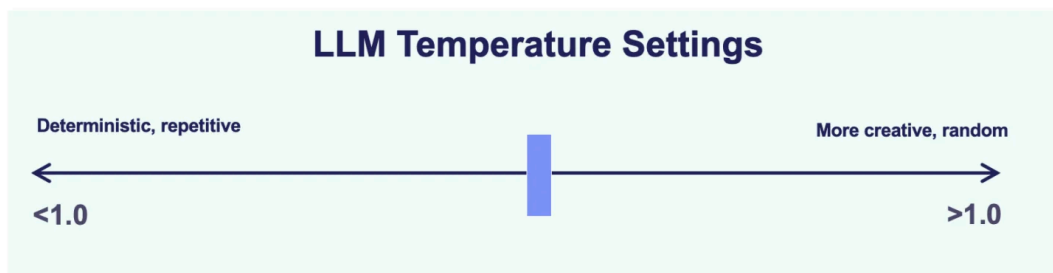
¹¹⁷ <https://www.ibm.com/think/topics/instruction-tuning>

LLM Temperature

A temperature is a number which controls the randomness of an LLM¹¹⁸. When using APIs for leveraging LLMs the temperature value has a specific range like from 0-2. We can think about it as adjusting how much the model is “explorative”\”conservative when answering¹¹⁹.

Overall, LLMs iteratively generate tokens, which basically represent words\parts of words. Foreach optional token there is an assigned likelihood, the goal of the temperature is to dictate how to weigh those likelihoods¹²⁰.

Lastly, by default temperature can have a value of 1 (like in OpenAI’s ChatGPT), which balances between randomness and determinism. If we lower the value we get completions which are less random. Thus, if the temperature value approaches zero the model is more deterministic and repetitive¹²¹. The value range of temperature is divided to: low temperature (<1.0), high temperature (>1.0) and 1.0 - as demonstrated in the diagram below¹²².



¹¹⁸ <https://medium.com/@boutnaru/the-artificial-intelligence-llm-large-language-model-f3e1a3fb15d6>

¹¹⁹ <https://towardsdatascience.com/a-comprehensive-guide-to-llm-temperature/>

¹²⁰ <https://www.vellum.ai/llm-parameters/temperature>

¹²¹ <https://www.hopsworks.ai/dictionary/llm-temperature>

¹²² <https://www.iguazio.com/glossary/llm-temperature/>



SLM (Small Language Model)

SLM (Small Language Model) is an AI¹²³ model which can be used (among others) to process/understand/generate natural language content. Their main goal is speed and realtime performance. They are more compact than LLMs¹²⁴. Also, they can run on smartphones, tablets and even smartwatches¹²⁵.

Probably the most significant difference between LLMs and SLMs is the size of the model. For example “GPT-4” which is a LLM has about 1.76 trillion parameters while an SLM like “Mistral 7B” can have only 7 billion model parameters. Moreover, an SLM is usually trained on data from a specific domain as opposed to LLMs - a table comparing “LLM vs LSM” is shown below¹²⁶. Because the model is smaller SLMs can run on local computers and generate data within acceptable time¹²⁷.

Lastly, a great example for using SLM usage is “Microsoft Recall”¹²⁸. By the way, examples of popular SLMs are: “DistilBERT”, “Gemma” and “Phi”¹²⁹.

LLM vs SLM

Aspect	LLM (Large Language Models)	SLM (Small Language Models)
Advantages 	<ul style="list-style-type: none">• Deep language understanding• Versatility• Contextual relevance	<ul style="list-style-type: none">• Efficient resource utilization• Suitable for smaller datasets• Faster training and inference
Drawbacks 	<ul style="list-style-type: none">• Computationally intensive• Data requirements• Potential for bias	<ul style="list-style-type: none">• Limited language understanding• Contextual limitations• Task-specific training

¹²³ <https://medium.com/@boutnaru/introduction-to-ai-artificial-intelligence-8c71d4d25320>

¹²⁴ <https://medium.com/@boutnaru/the-artificial-intelligence-llm-large-language-model-f3e1a3fb15d6>

¹²⁵ <https://www.datacamp.com/blog/top-small-language-models>

¹²⁶ <https://www.thesunflowerlab.com/llms-vs-slms/>

¹²⁷ https://www.splunk.com/en_us/blog/learn/language-models-slm-vs-llm.html

¹²⁸ <https://medium.com/@boutnaru/the-windows-concept-journey-windows-copilot-runtime-5f4c824d6f7b>

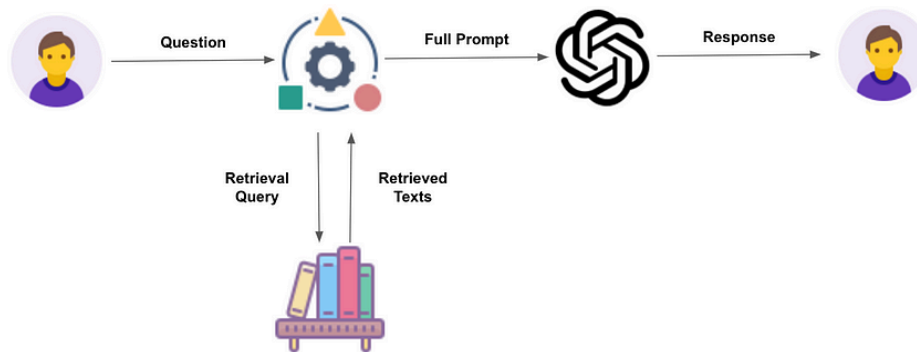
¹²⁹ <https://www.ibm.com/think/topics/small-language-models>

RAG (Retrieval-Augmented Generation)

RAG (Retrieval-Augmented Generation) is an AI¹³⁰ framework which combines the capabilities of GenAI¹³¹ and traditional retrieval systems like search\ databases. By doing so we can get better results which are more accurate\up-to-date\relevant for specific needs¹³². By leveraging a RAG we don't have to regularly train the model on new data and update its parameters¹³³.

Overall, RAGs try to cope with the known challenges of LLMs¹³⁴ such as: presenting false information when it does not have the answer, presenting out-of-date information, creating responses from non-authoritative sources and creating inaccurate responses due to terminology confusions. RAGs operate by first retrieving data from different data sources. Second, augment the LLM prompt using the retrieved data which. Thirud, send the full prompt to to an LLM which sends the response for the user question¹³⁵ - as shown below¹³⁶.

Lastly, there are different real-world use-case that can highly benefit from RAGs like: virtual assistants, content creation, medical diagnostics, clinical trial design optimization, code generation, sales automation, financial planning, customer support and knowledge management¹³⁷. We can also use RAGs to build by trust by providing the model sources which it can cite. This allows users to check any claim given by the model as a response¹³⁸.



¹³⁰ <https://medium.com/@boutnaru/introduction-to-ai-artificial-intelligence-8c71d4d25320>

¹³¹ <https://medium.com/@boutnaru/the-artificial-intelligence-journey-genai-generative-artificial-intelligence-29d1228e905e>

¹³² <https://cloud.google.com/use-cases/retrieval-augmented-generation>

¹³³ <https://www.codiste.com/what-is-a-retrieval-augmented-generation>

¹³⁴ <https://medium.com/@boutnaru/the-artificial-intelligence-llm-large-language-model-f3e1a3fb15d6>

¹³⁵ <https://aws.amazon.com/what-is/retrieval-augmented-generation/>

¹³⁶ <https://nbkcomputer.com/what-is-retrieval-augmented-generation-rag-embedding-model-vector/>

¹³⁷ <https://www.signitysolutions.com/blog/real-world-examples-of-retrieval-augmented-generation>

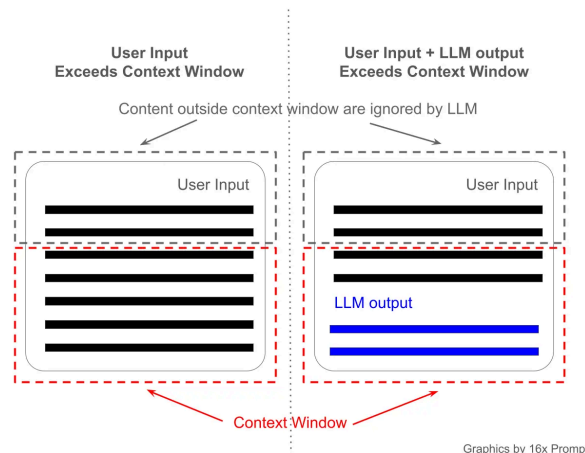
¹³⁸ <https://blogs.nvidia.com/blog/what-is-retrieval-augmented-generation/>

Context Window (aka Context Length)

Context Window (aka Context Length) is an attribute of an AI model. It is the amount of text (in tokens) that the model can consider\remember at any point of time. We can think about the context window like the “working memory” of an AI model. Thus, it determines how long of a conversation it can carry out without forgetting details from earlier in the exchange. Also determines the maximum size of documents or code samples that it can process at once¹³⁹ - as shown in the diagram below¹⁴⁰.

Overall, context windows are important because they help AI models recall information during a session. For example Gemini could process up to 32K tokens, while “Gemini 1.5 Pro” has a context window of up to 1 million tokens and as of April 2025 Google also tested up to 10 million token in their research¹⁴¹. For better understanding: “4K tokens is about 3K words and about 6 pages”, “32K tokens are about 24 words and about 48 pages”, “128K tokens are about 96K words and 192 pages” and “200K tokens are about 150K words and 300 pages”¹⁴². Based on those numbers we can conclude that “1M tokens are about 750K words and about 1500 pages”.

Lastly, it is important to know that although a large context window has benefits like forming better connections between words and improved contextual information it also has drawbacks (the “context window paradox”). This is because it can cause: information overload (reducing focus and slowing performance), getting lost in data (prioritize edges and missing key middle info), understanding relationship becomes harder and poor management due to noise¹⁴³.



¹³⁹ <https://www.ibm.com/think/topics/context-window>

¹⁴⁰ <https://prompt.16x.engineer/blog/claude-daily-usage-limit-quota>

¹⁴¹ <https://blog.google/technology/ai/long-context-window-ai-models/>

¹⁴² <https://lifearchitect.ai/gemini/>

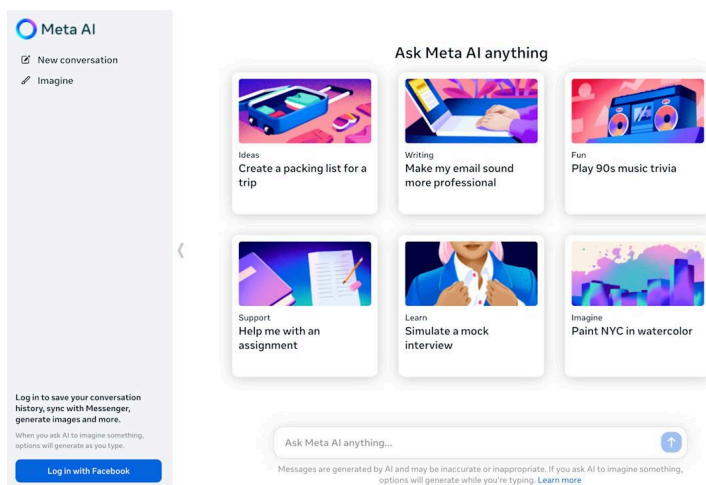
¹⁴³ <https://datasciencedojo.com/blog/the-llm-context-window-paradox/>

Llama (Large Language Model Meta AI)

Llama (Large Language Model Meta AI) is a family of LLMs¹⁴⁴ that has been released by “Meta” since February 2023. The models are trained on different parameter sizes between 1B-405B. By the way, Meta has added a virtual assistant feature to Facebook\WhatsApp based on the “Llama 3” model¹⁴⁵. Beside the parameters sizes the models also differ in the context window size¹⁴⁶ for example: “Llama 3” has a 8K context window while Llama 3.1\Llama 3.2\Llama 3.3 has a context window of 128K.

Overall, Llama are open source models which we can fine-tune, distill and deploy anywhere. We can deploy a 1B/3B based model on end devices for different use-cases like summarizing. Also, we can use the 11B/90B models for multimodal interactions like transforming an existing image into something new¹⁴⁷.

Lastly, meta releases as part of “Llama” pre-trained models and instruction tuned models¹⁴⁸. By the way, we can play with the “Meta AI” using a dedicated web application¹⁴⁹ - as shown in the screenshot below¹⁵⁰.



¹⁴⁴ <https://medium.com/@boutnaru/the-artificial-intelligence-llm-large-language-model-f3e1a3fb15d6>

¹⁴⁵ [https://en.wikipedia.org/wiki/Llama_\(language_model\)](https://en.wikipedia.org/wiki/Llama_(language_model))

¹⁴⁶ <https://medium.com/@boutnaru/the-artificial-intelligence-journey-context-window-aka-context-length-9ead45714938>

¹⁴⁷ <https://www.llama.com/>

¹⁴⁸ <https://medium.com/@boutnaru/the-artificial-intelligence-journey-instruction-tuned-llm-f08585fded43>

¹⁴⁹ <https://meta.ai/>

¹⁵⁰ <https://zapier.com/blog/llama-meta/>