Introduction: Artificial Neural Networks

SCO411 Neural Networks

Eric Araka Ph.D.

ICT Manager, May 2012 to April 2018, Express Communications Ltd

Associate Faculty, May 2013 – September 2016 Mount Kenya University

Tutorial Fellow, October 2016 to March 2021 Technical University of Kenya

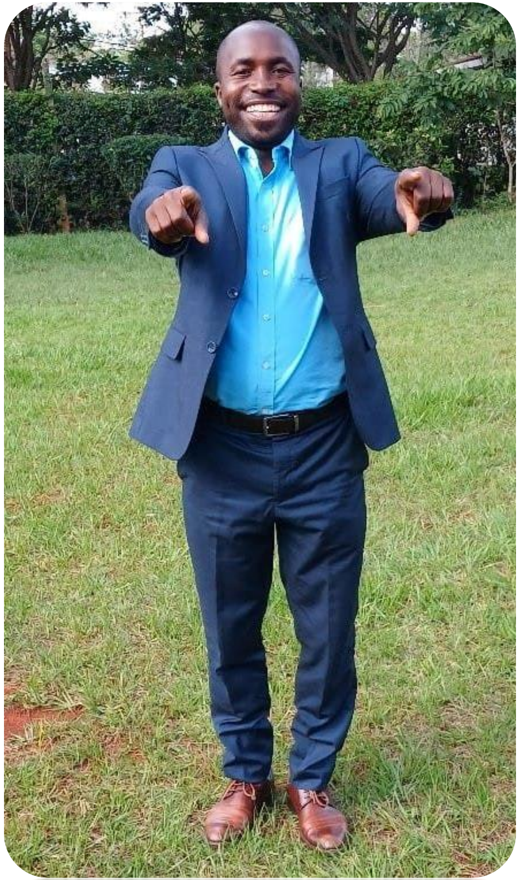
Tutorial Fellow, April 2021 to August 2023 Kenyatta University

Lecturer, September 2023 to Present, Kenyatta University

Bsc. Computer Science, August 2008- December 2012, Kenyatta University

Msc. Distributed Computing Technologies, May 2013 – September 2015 University of Nairobi

PhD Computer Science, May 2019 – July 2023 Kenyatta University



Dr. Eric Araka, PhD

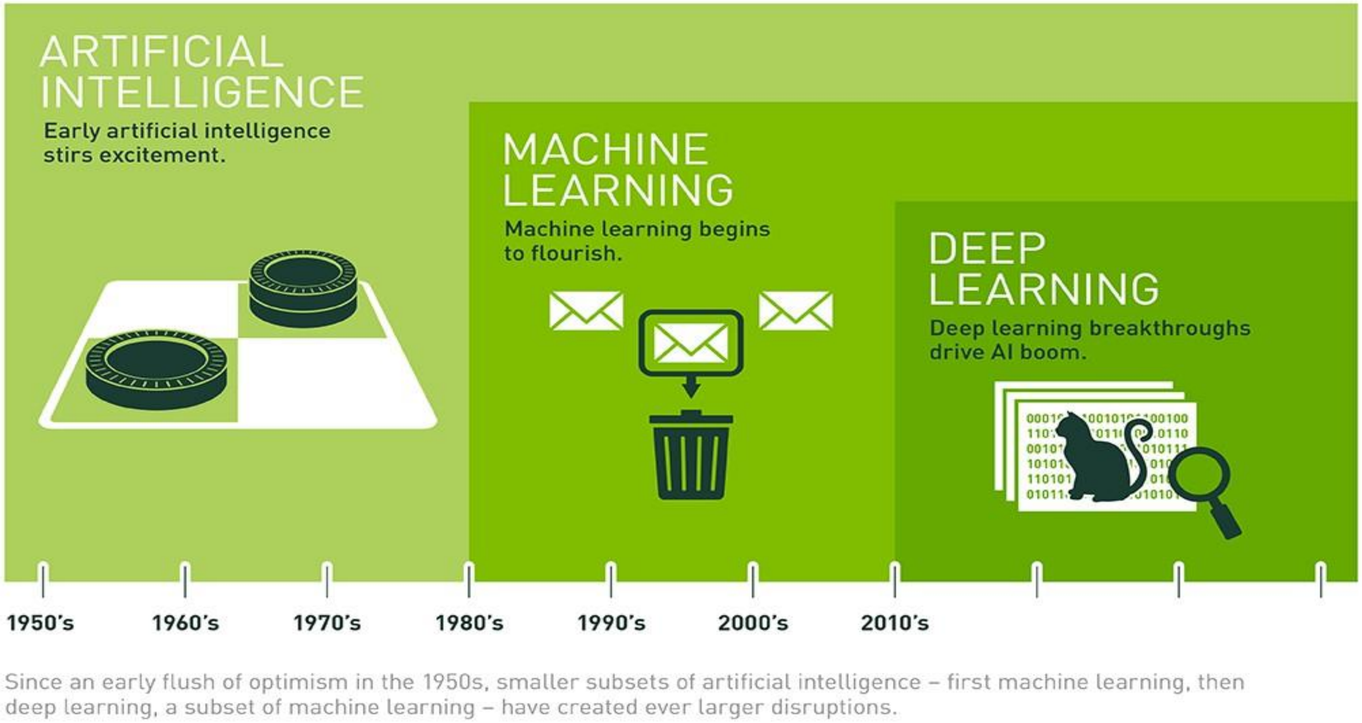
Introductions

Instructor

- Eric Araka
- Lecturer, School of Pure and Applied Sciences, Computing & Information Science Department
- CIS Building, Room 114
- Email: araka.eric@ku.ac.ke
- Tel: +254784815187 I prefer SMS/WhatsApp

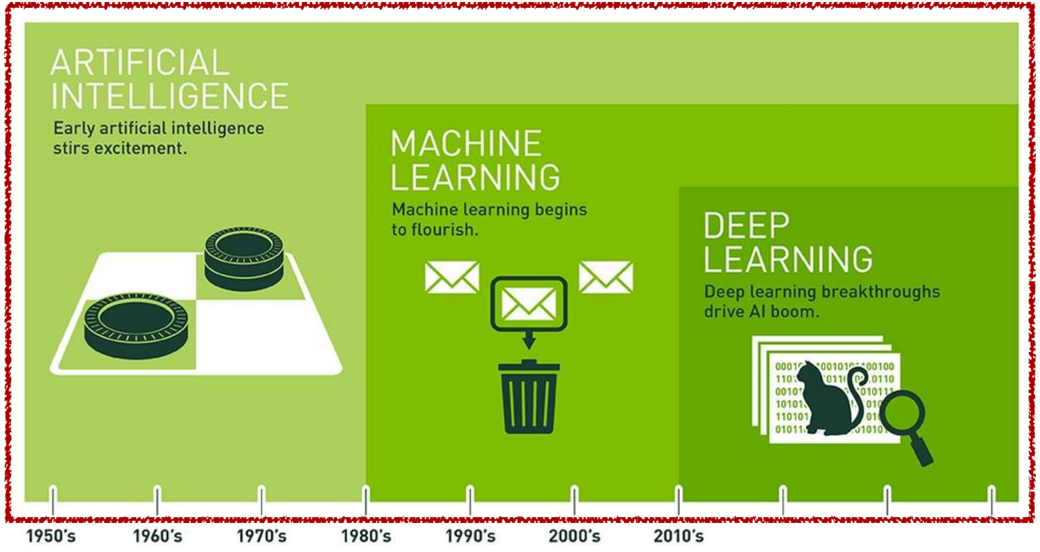
- Students?

Deep Learning, ML and AI



Source: http://bit.ly/2suufGJ

Deep Learning, ML and AI



Since an early flush of optimism in the 1950s, smaller subsets of artificial intelligence – first machine learning, then deep learning, a subset of machine learning – have created ever larger disruptions.

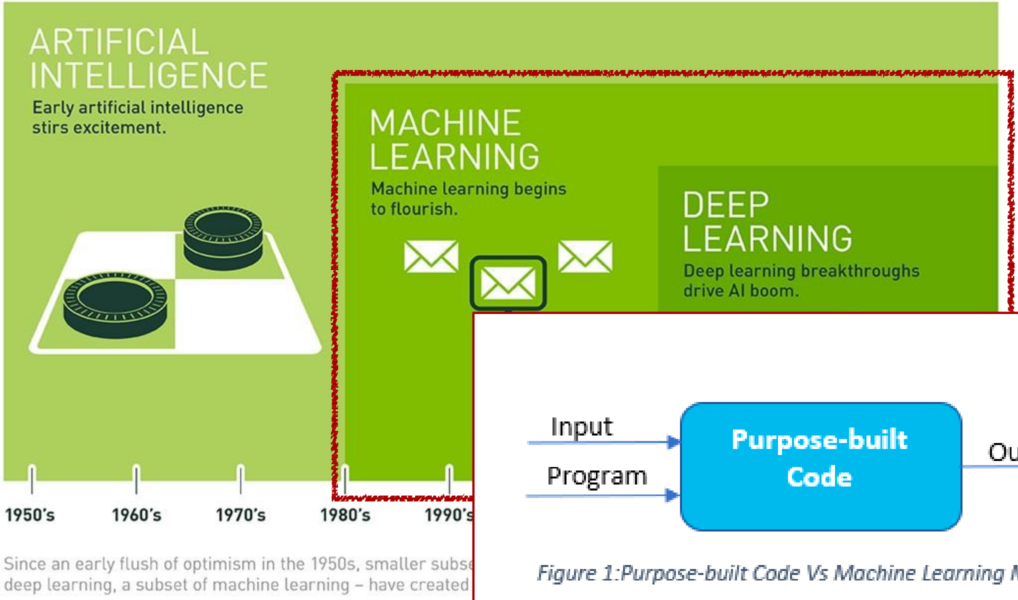
Artificial Intelligence

Artificial Intelligence is human intelligence exhibited by machines

[The world's best Dota 2 players just got destroyed by a killer AI from Elon Musk's startup]

[Elon Musk-funded Dota 2 bots spank top-tier humans, and they know how to trash talk]

Deep Learning, ML and AI



Machine Learning

Field of study that gives computers the ability to learn without being explicitly programmed.

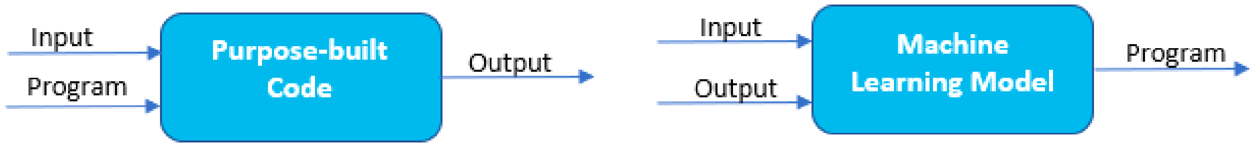


Figure 1:Purpose-built Code Vs Machine Learning Model

Source: https://bit.ly/2NnyrOz

Machine Learning

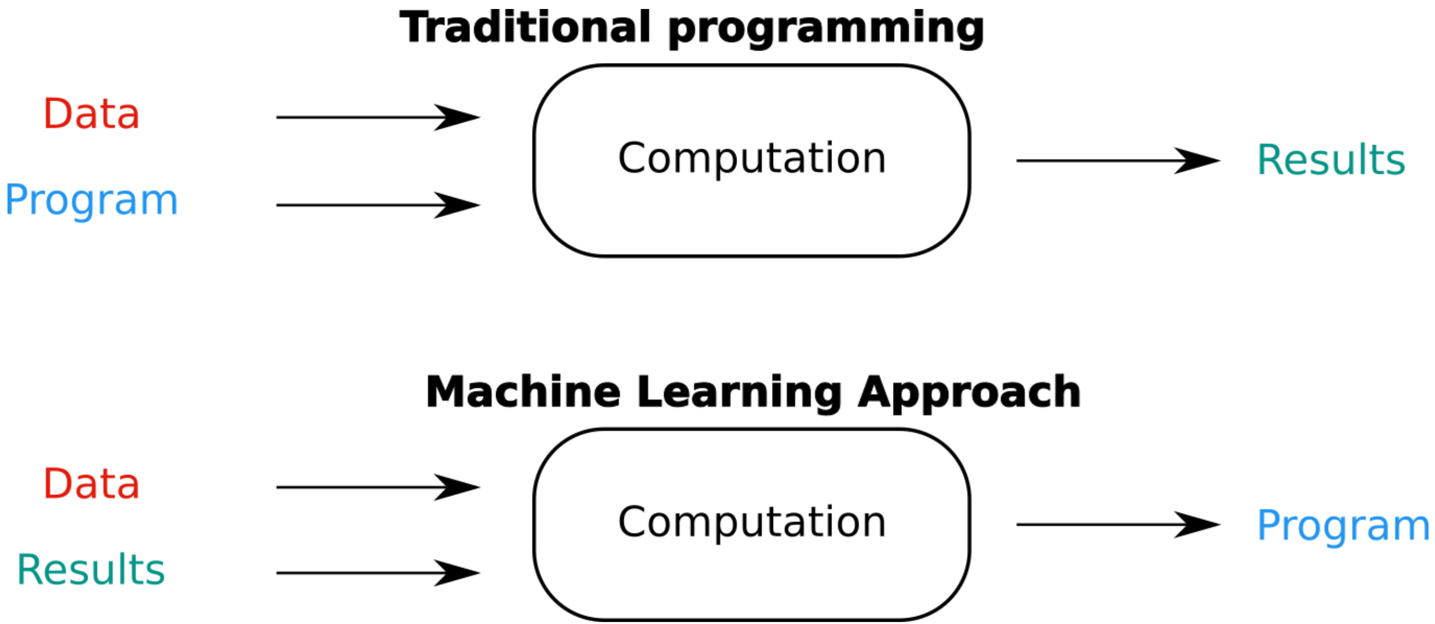


Image credits: https://recast.ai/blog/machine-learning-algorithms/

When Would We Use Machine Learning?

- When patterns exists in our data
  - Even if we don't know what they are
    - Or perhaps especially when we don't know what they are

- We can not pin down the functional relationships mathematically
  - Else we would just code up the algorithm

- When we have lots of (unlabeled) data
  - Labeled training sets harder to come by
  - Data is of high-dimension
    - High dimension "features"
    - For example, sensor data
  - Want to "discover" lower-dimension representations
    - Dimension reduction

- Aside: Machine Learning is heavily focused on implementability
  - Frequently using well know numerical optimization techniques
  - Lots of open source code available
    - See e.g., libsvm (Support Vector Machines): http://www.csie.ntu.edu.tw/~cjlin/libsvm/
    - Most of my code in python: http://scikit-learn.org/stable/ (many others)
    - Languages (e.g., octave: https://www.gnu.org/software/octave/)

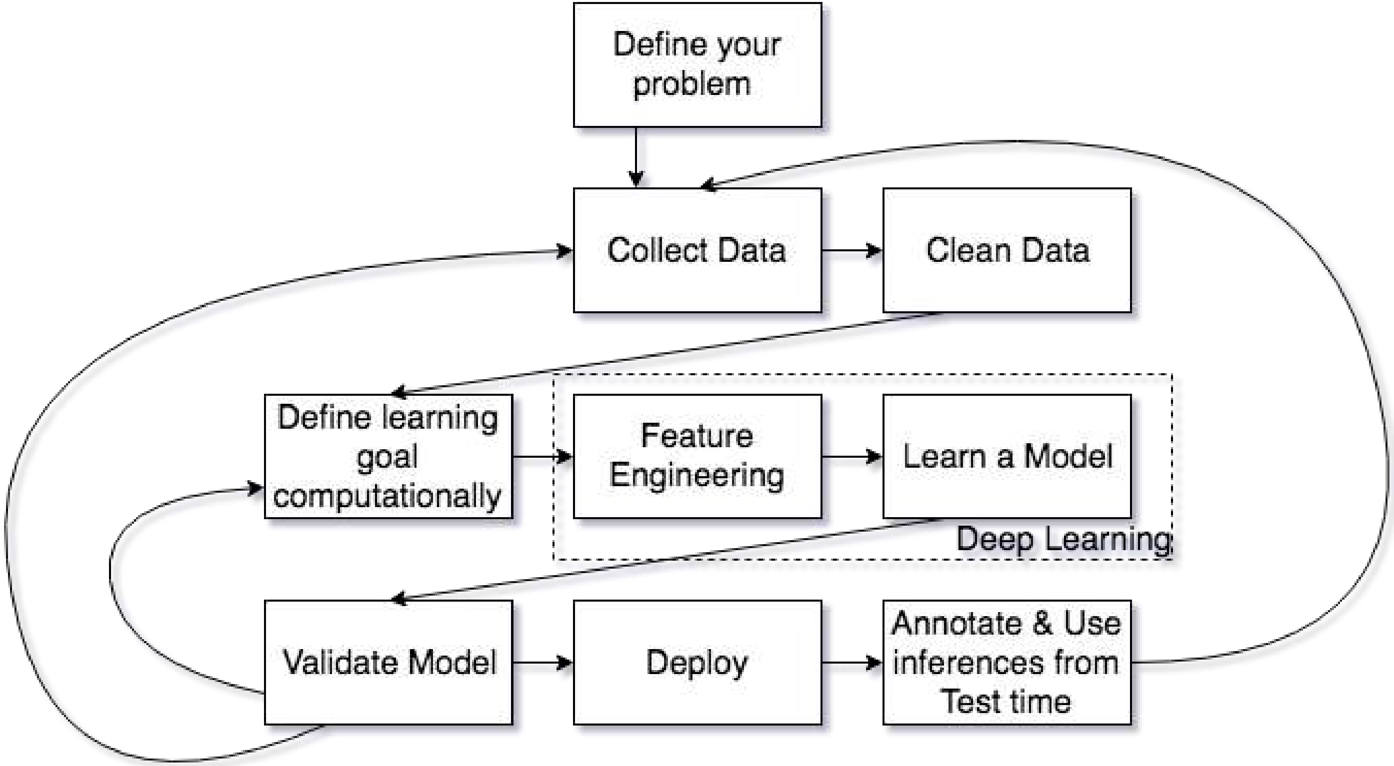
Examples of Machine Learning Problems

- Pattern Recognition
  - Facial identities or facial expressions
  - Handwritten or spoken words (e.g., Siri)
  - Medical images
  - Sensor Data/IoT
- Optimization
  - Many parameters have “hidden” relationships that can be the basis of optimization
- Pattern Generation
  - Generating images or motion sequences
- Anomaly Detection
  - Unusual patterns in the telemetry from physical and/or virtual plants (e.g., data centers)
  - Unusual sequences of credit card transactions
  - Unusual patterns of sensor data from a nuclear power plant
    - or unusual sound in your car engine or …
- Prediction
  - Future stock prices or currency exchange rates

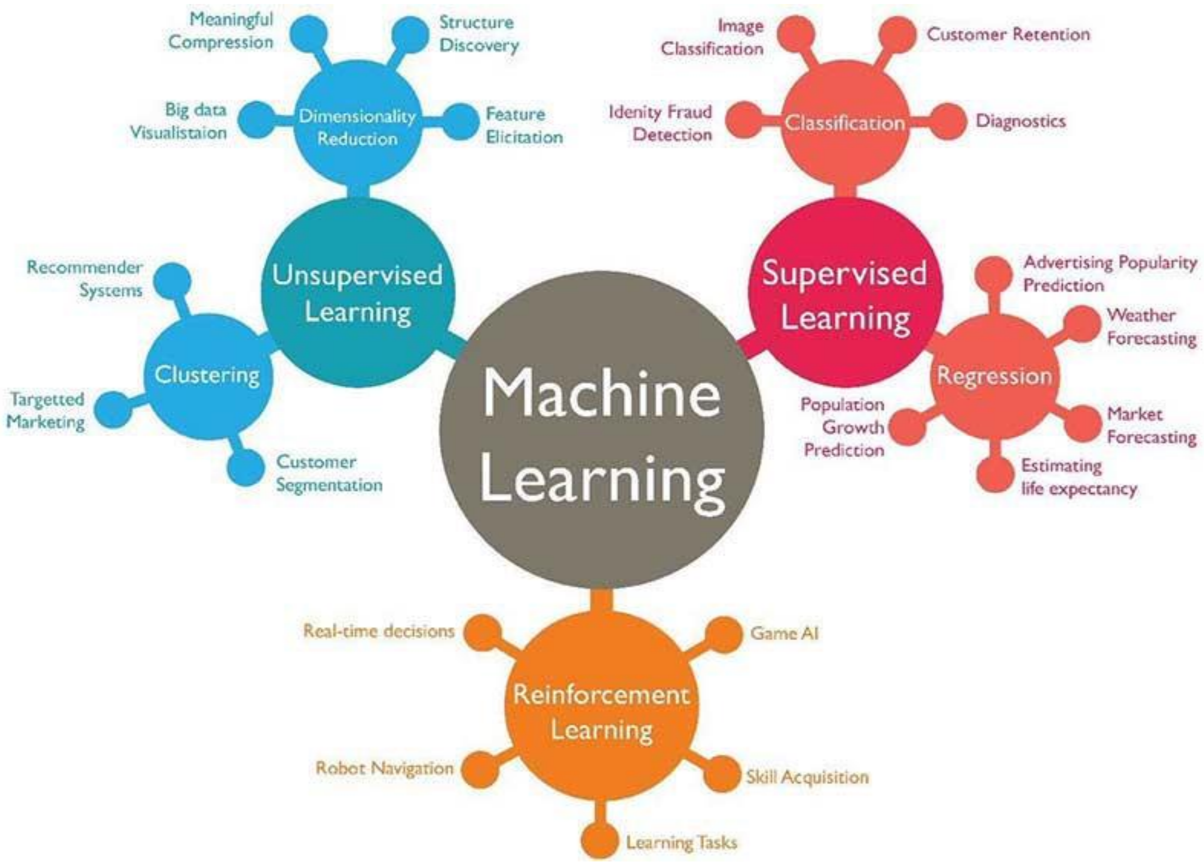
Machine Learning

Blends ideas from statistics, computer science, operations research, pattern recognition, information theory, control theory and many other disciplines to design algorithms that find low-level patterns in data, make predictions and help make decisions (at scale).

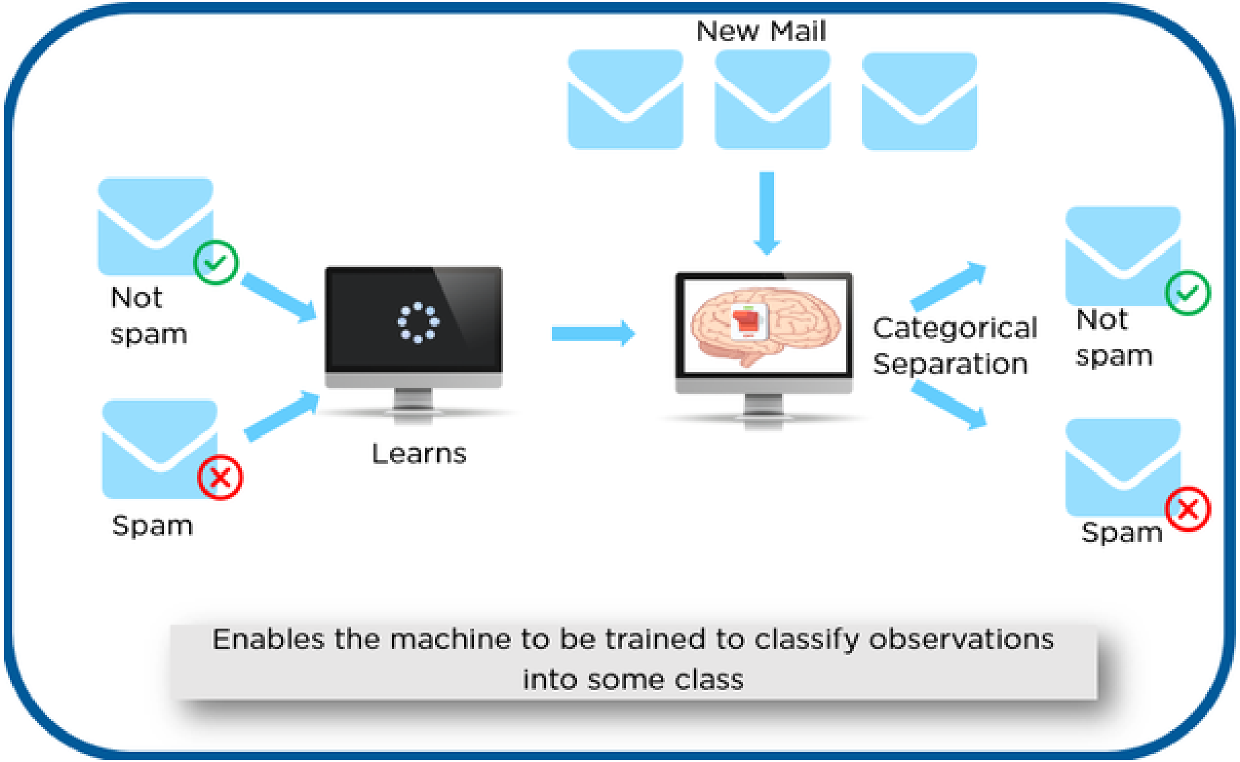
Typical Machine Learning Pipeline



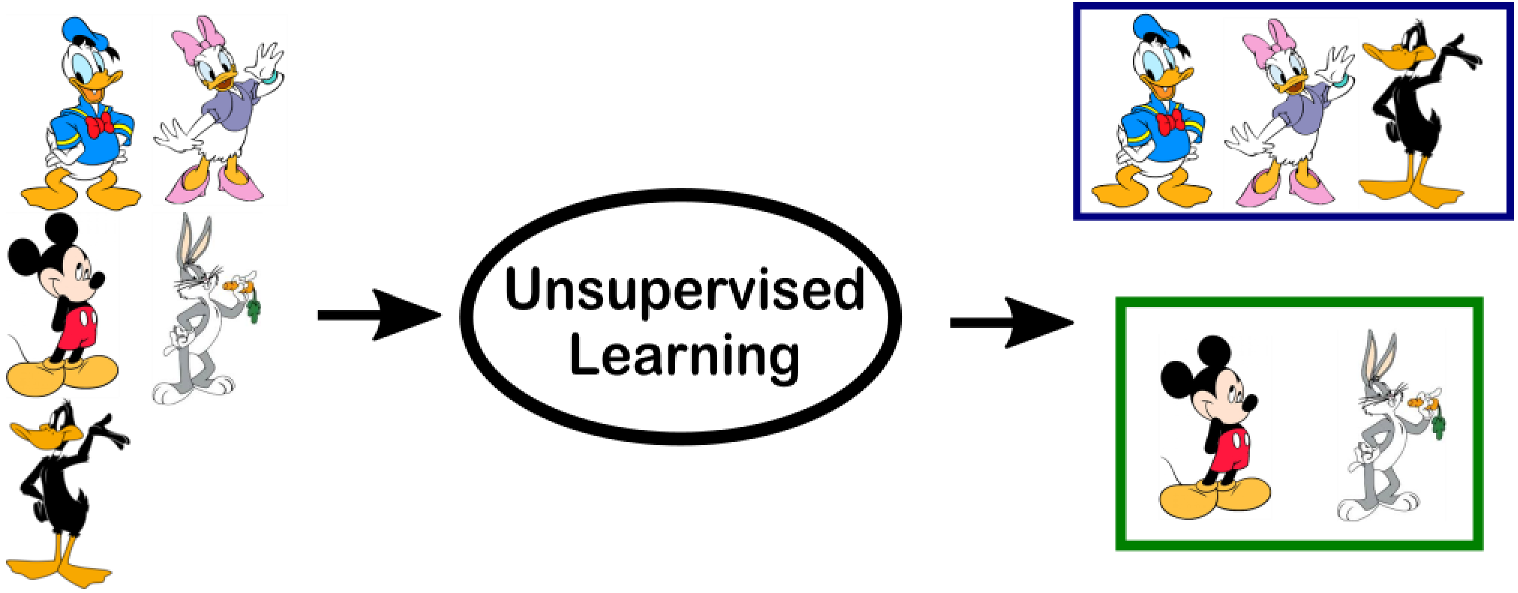
Types of Machine Learning



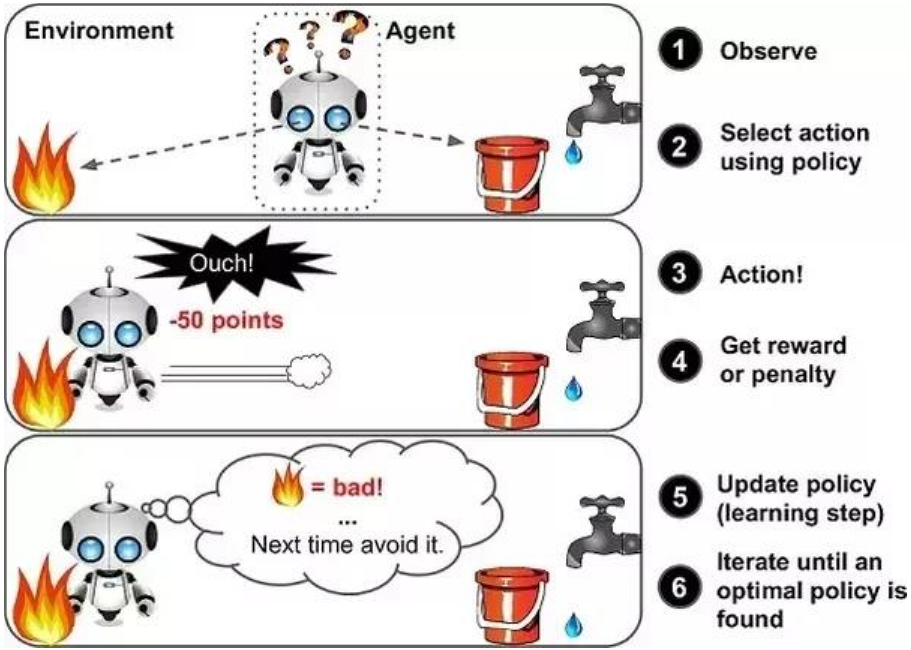
Example of Supervised Learning



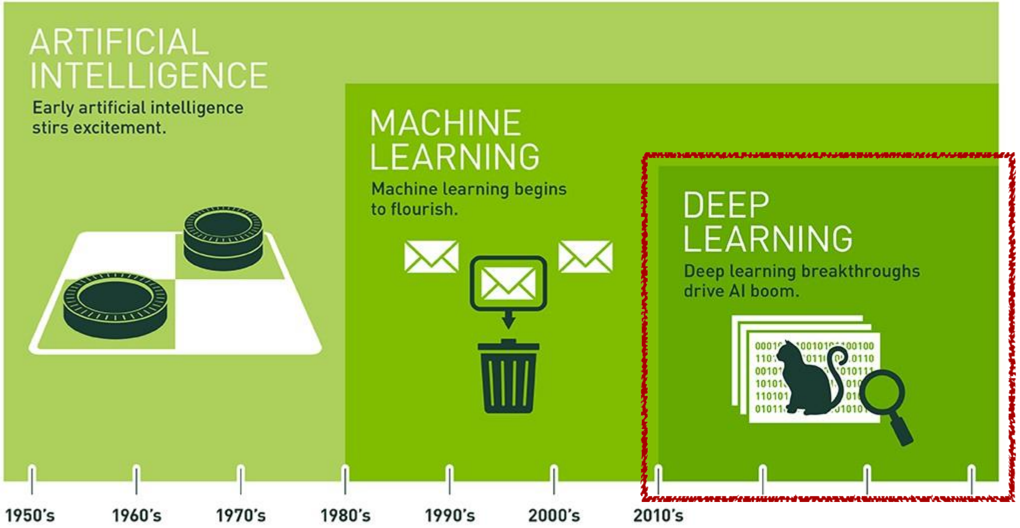
Example of Unsupervised Learning



Example of Reinforcement Learning



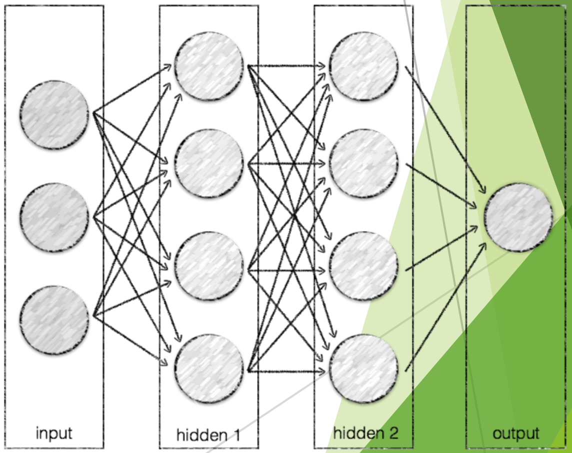
Deep Learning, ML and AI



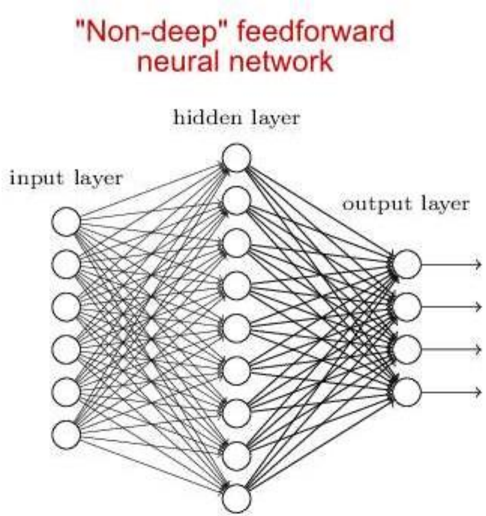
Since an early flush of optimism in the 1950s, smaller subsets of artificial intelligence – first machine learning, then deep learning, a subset of machine learning – have created ever larger disruptions.

Deep Learning

Using deep neural networks to implement machine learning



Few words on “Deep learning”



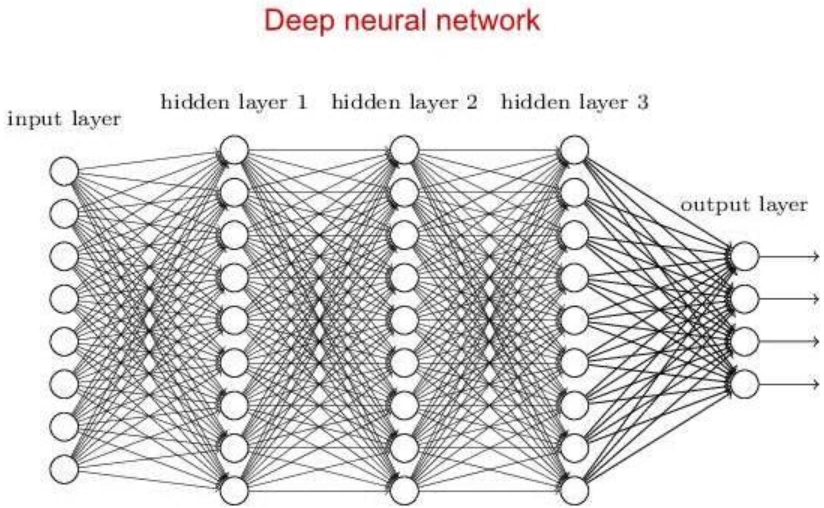


Image Credits: http://neuralnetworksanddeeplearning.com/chap5.html

ANN one of the Machine Learning Algorithms

- Machine learning is a paradigm that may refer to learning from past experience (which in this case is previous data) to improve future performance.
- The sole focus of this field is automatic learning methods. Learning refers to modification or improvement of algorithm based on past “experiences” automatically without any external assistance from human

Definition

Neural network is a machine learning technique which enables a computer to learn from the observational data. Neural network in computing is inspired by the way biological nervous system process information.

Biological neural networks consist of interconnected neurons with dendrites that receive inputs. Based on these inputs, they produce an output through an axon to another neuron.

Artificial neural networks (ANNs) or connectionist systems are computing systems vaguely inspired by the biological neural networks that constitute animal brains. Such systems "learn" to perform tasks by considering examples, generally without being programmed with any task-specific rules.

Neural Network

- A Neural Network is a massively parallel distributed processor made up of simple processing units that has a natural propensity for storing experiential knowledge and making it available for use.
- It resembles the brain in two respects:
  1. Knowledge is acquired by the network from its environment through a learning process.
  2. Interneuron connection strengths, known as synaptic weights, are used to store the acquired knowledge.
- The procedure used to perform the learning process is called a *learning algorithm*, the function of which is to modify the synaptic weights of the network in an orderly fashion to attain a desired design objective.
- The modification of synaptic weights provides the traditional method for the design of neural networks.

Birds inspired us to fly, burdock plants inspired velcro, and nature has inspired many other inventions. It seems only logical, then, to look at the brain's architecture for inspiration on how to build an intelligent machine.

This is the key idea that inspired *artificial neural networks* (ANNs). ANNs are at the very core of Deep Learning.

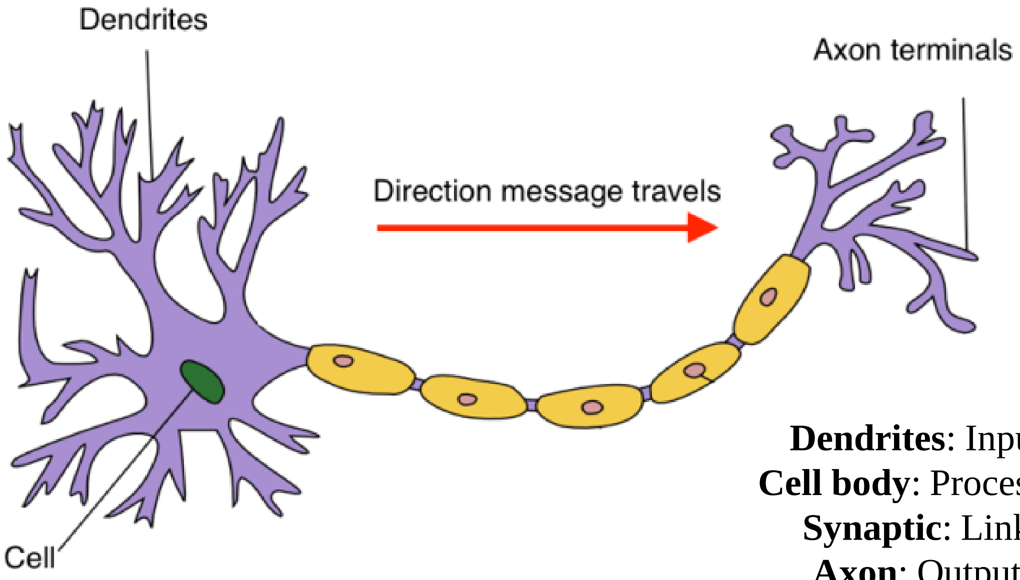
They are versatile, powerful, and scalable, making them ideal to tackle large and highly complex Machine Learning tasks, such as classifying billions of images (e.g., **Google Images**), powering speech recognition services (e.g., **Apple's Siri**), recommending the best videos to watch to hundreds of millions of users every day (e.g., **YouTube**), or learning to beat the world champion at the game of *Go* by examining millions of past games and then playing against itself (**DeepMind's AlphaGo**).

In this lesson, we will introduce artificial neural networks, starting with a quick tour of the very first ANN architectures. Then we will present *Multi-Layer Perceptrons* (MLPs) and implement one using TensorFlow to tackle the MNIST digit classification problem.

Reason for ANN having profound impact

1. There is now **a huge quantity of data available** to train neural networks, and ANNs frequently outperform other ML techniques on very large and complex problems.
2. The **tremendous increase in computing power** since the 1990s now makes it possible to train large neural networks in a reasonable amount of time. This is in part due to Moore's Law, but also thanks to the gaming industry, which has produced powerful GPU cards by the millions.
3. **The training algorithms have been improved.** To be fair they are only slightly different from the ones used in the 1990s, but these relatively small tweaks have a huge positive impact.
4. ANNs seem to have entered a virtuous circle of funding and progress. Amazing products based on ANNs regularly make the headline news, which pulls more and more attention and funding toward them, resulting in more and more progress, and even more amazing products.

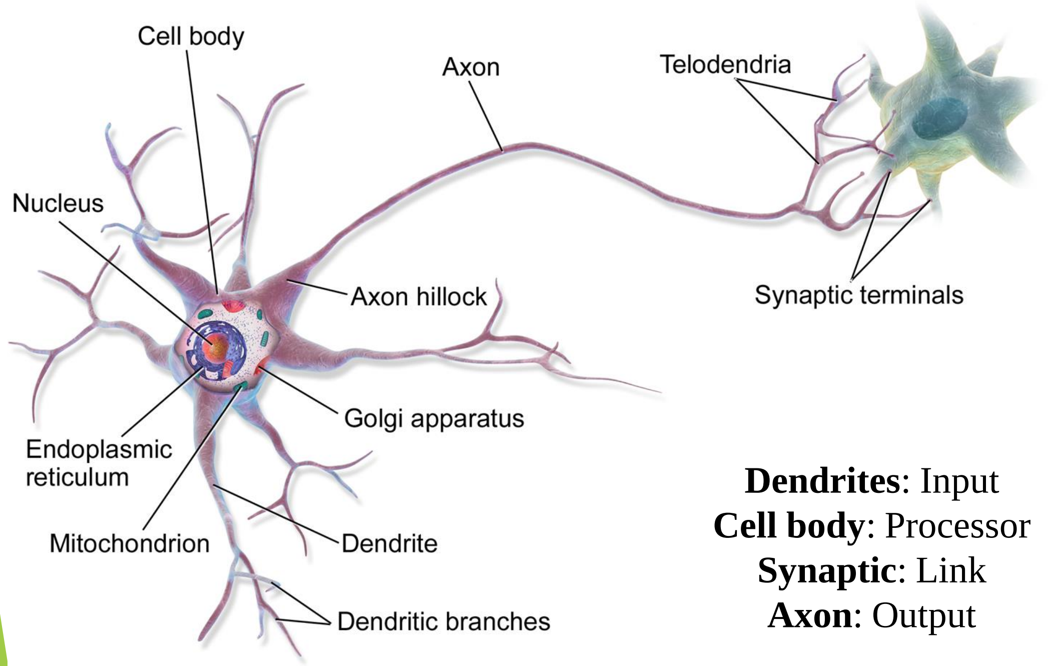
Structure of a Neuron



Dendrites: Input
Cell body: Processor
Synaptic: Link
Axon: Output

The term “neural network” is derived from the work of a neuroscientist, Warren S. McCulloch and Walter Pitts, a logician, who developed the first conceptual model of an artificial neural network. In their work, they describe the concept of a neuron, a single cell living in a network of cells that receives inputs, processes those inputs, and generates an output.

Structure of a Neuron



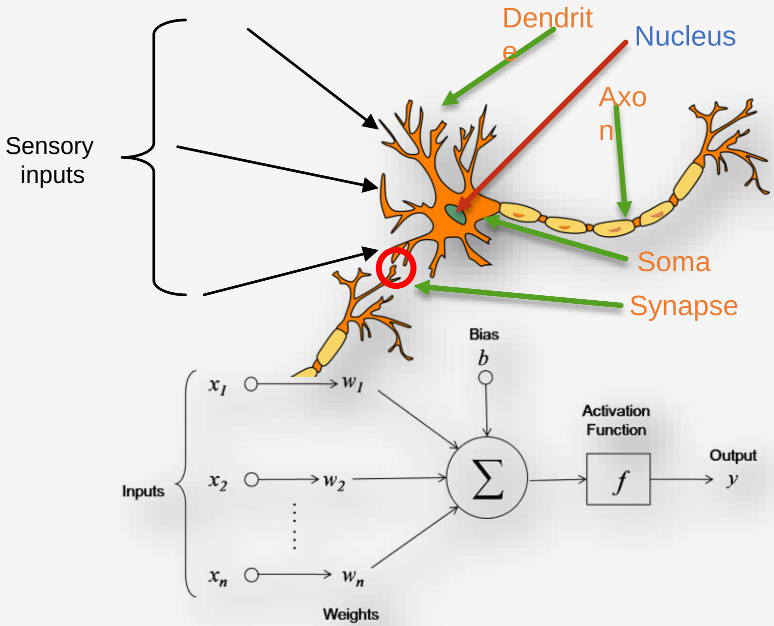
Dendrites: Input
Cell body: Processor
Synaptic: Link
Axon: Output

Biological neurons receive short electrical impulses called *signals* from other neurons via these synapses. When a neuron receives a sufficient number of signals from other neurons within a few milliseconds, it fires its own signals.

Thus, individual biological neurons seem to behave in a rather simple way, but they are organized in a vast network of billions of neurons, each neuron typically connected to thousands of other neurons. Highly complex computations can be performed by a vast network of fairly simple neurons, much like a complex anthill can emerge from the combined efforts of simple ants.

Neural Networks

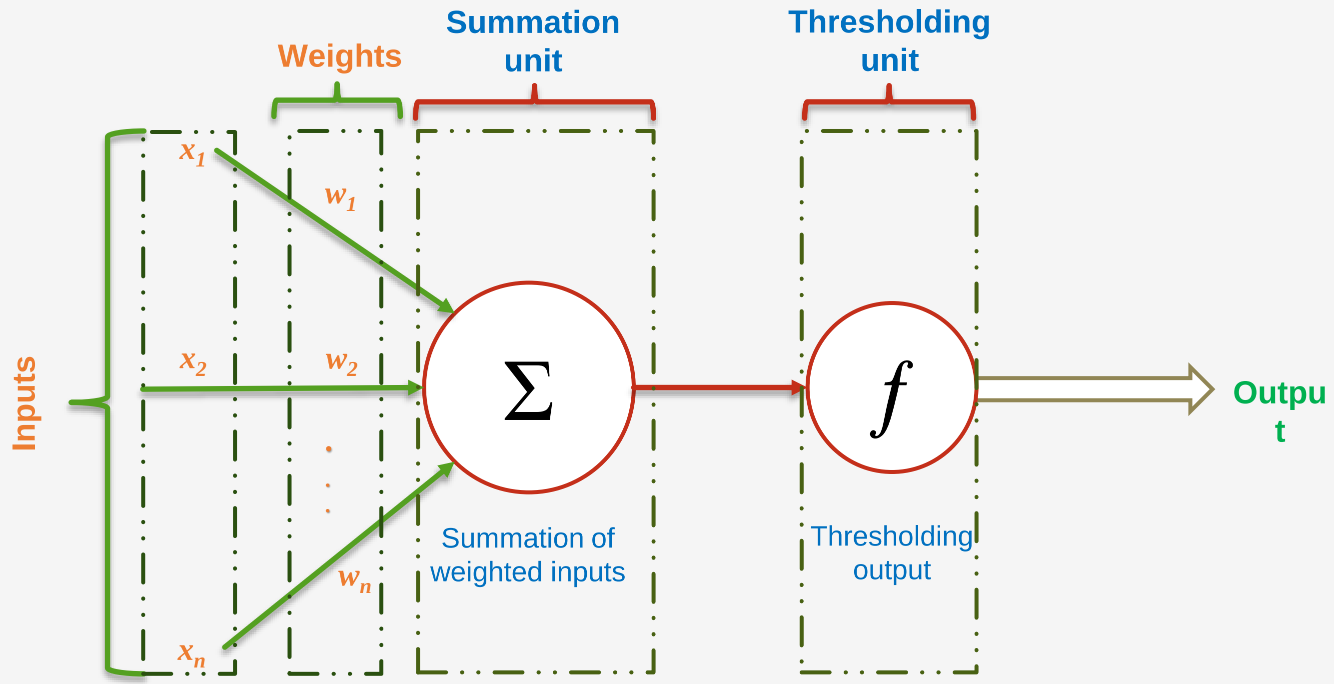
- Simplified models of the biological nervous system
- Processing elements called neurons – inspired by the brain
- Parallel distributed processing
- Characteristics:
  - mapping capabilities or pattern association
  - robustness
  - fault tolerance
  - parallel and high speed information processing
  - nonlinearity
  - adaptivity

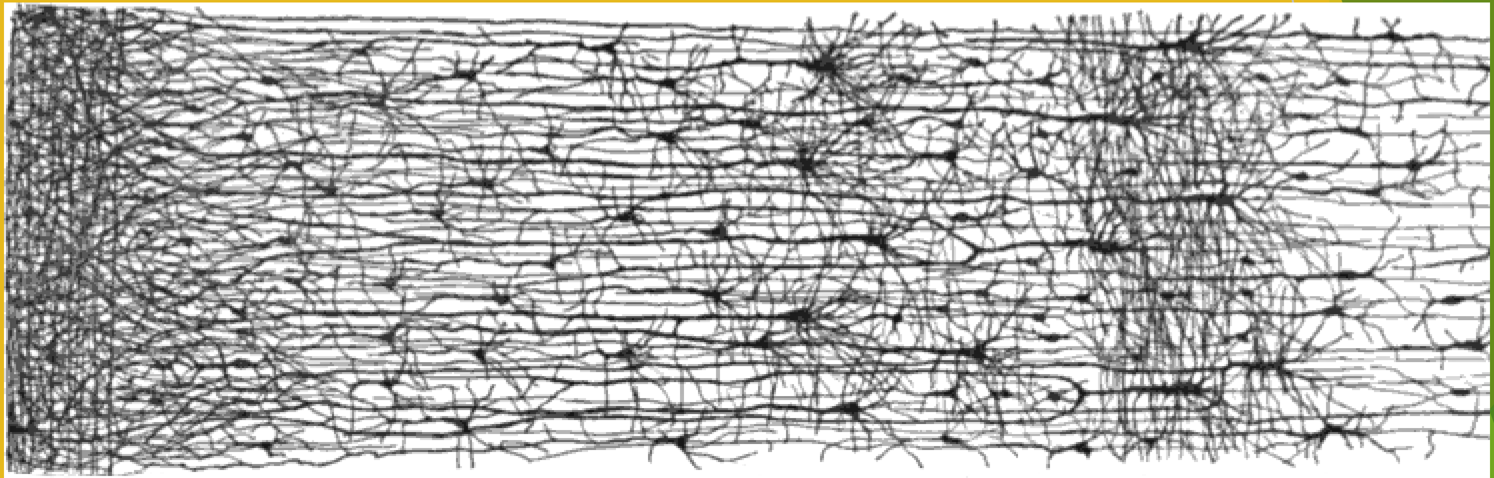


Terminology Relationship

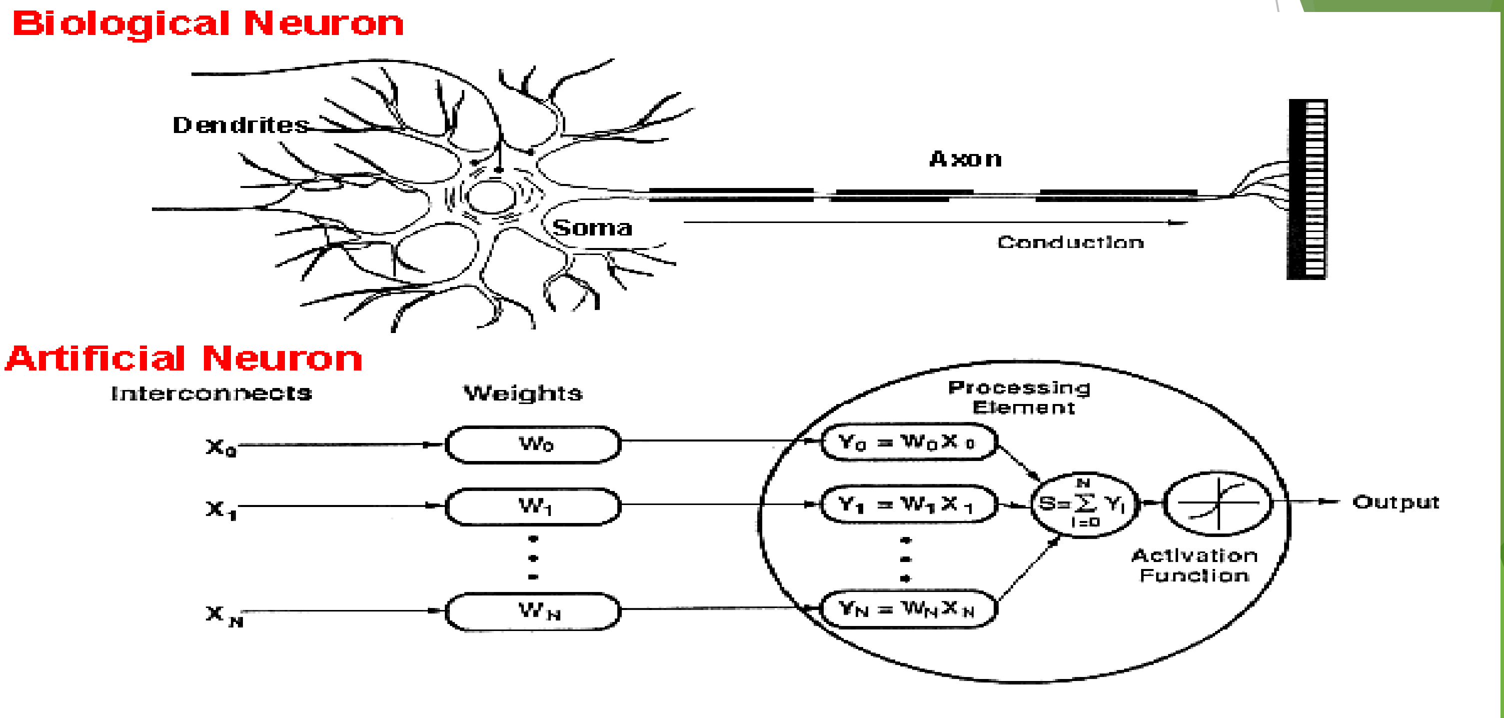
Biological Neuron	Artificial Neuron
Cell	Neuron
Dendrites	Weights or Interconnections
Soma	Net input
Axon	Output

Simple Model of Artificial Neuron





How do ANNs work?



An artificial neuron is an imitation of a human neuron

Briefly, **a neural network** is defined as a computing system that consist of a number of simple but highly interconnected elements or nodes, called '**neurons**', which are organized in layers which process information using dynamic state responses to external inputs. This algorithm is extremely useful in finding patterns that are too complex for being manually extracted and taught to recognize to the machine. In the context of this structure, patterns are introduced to the neural network by the ***input layer*** that has one neuron for each component present in the input data and is communicated to one or more ***hidden layers*** present in the network; called 'hidden' only due to the fact that they do not constitute the input or output layer. It is in the hidden layers where all the processing actually happens through a system of connections characterized by ***weights and biases*** *(commonly referred as W and b)****:*** the input is received, the neuron calculate a weighted sum adding also the bias and according to the result and a pre-set **activation function** (most common one is sigmoid, *σ,* even though it almost not used anymore and there are better ones like ReLu*)*, it decides whether it should be 'fired' or activated. Afterwards, the neuron transmit the information downstream to other connected neurons in a process called '*forward pass*'. At the end of this process, the last hidden layer is linked to the ***output layer*** which has one neuron for each possible desired output.

Characteristics of Artificial Neural Networks

- A large number of very simple processing neuron-like processing elements
- A large number of weighted connections between the elements
- Distributed representation of knowledge over the connections Knowledge is acquired by network through a learning process

The Architecture of an Artificial Neural Network

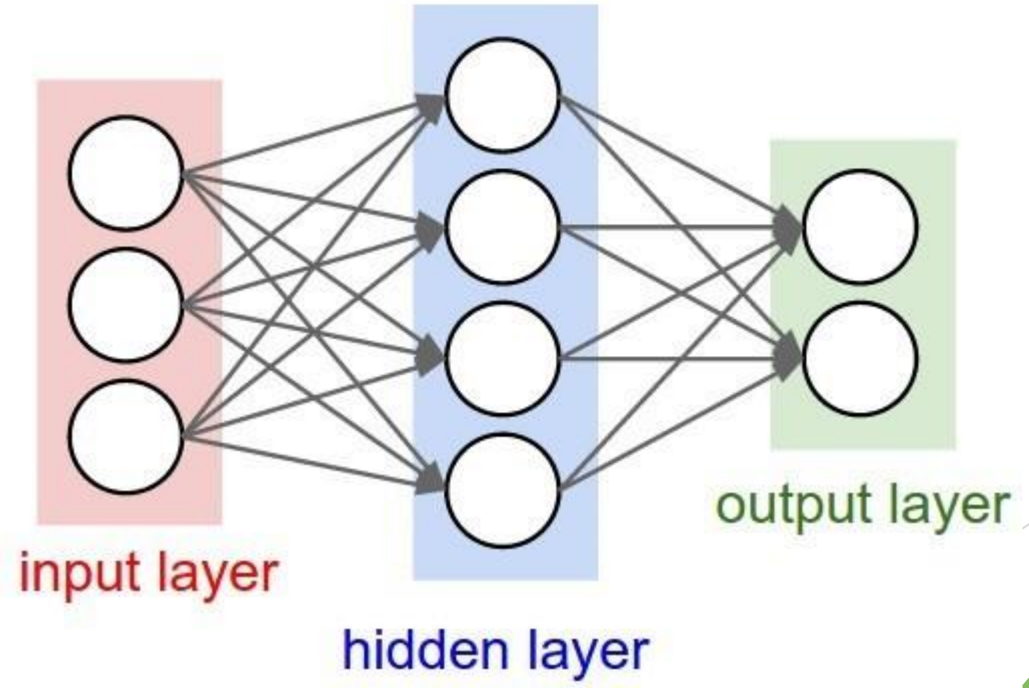
ANN is a set of connected neurons organized in layers:

input layer: brings the initial data into the system for further processing by subsequent layers of artificial neurons.

hidden layer: a layer in between input layers and output layers, where artificial neurons take in a set of weighted inputs and produce an output through an activation function.

output layer: the last layer of neurons that produces given outputs for the program.

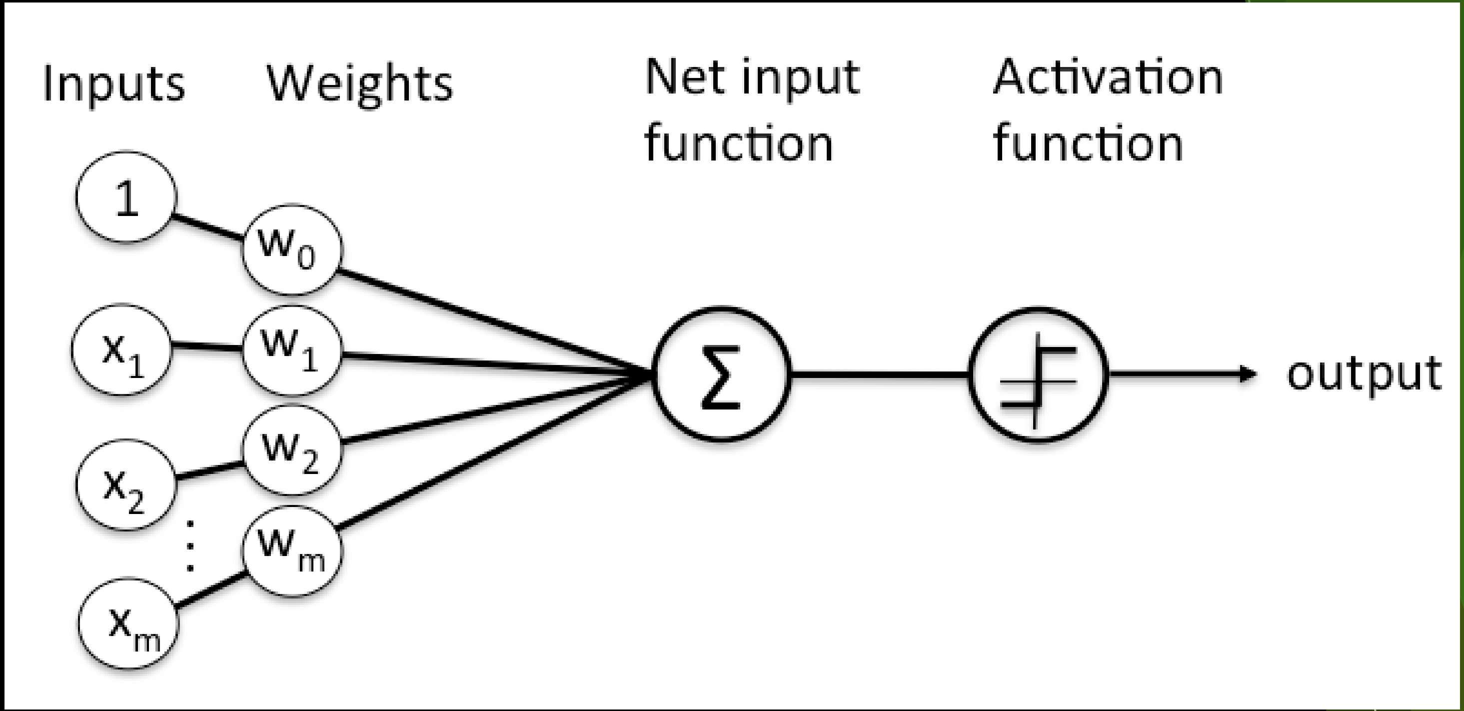
The Architecture of an Artificial Neural Network



The layers are made of *nodes*. A node is just a place where computation happens, loosely patterned on a neuron in the human brain, which fires when it encounters sufficient stimuli.

A node combines input from the data with a set of coefficients, or weights, that either amplify or dampen that input, thereby assigning significance to inputs with regard to the task the algorithm is trying to learn; e.g. which input is most helpful is classifying data without error?

These input-weight products are summed and then the sum is passed through a node’s so-called activation function, to determine whether and to what extent that signal should progress further through the network to affect the ultimate outcome, say, an act of classification. If the signals passes through, the neuron has been “activated.”



[redacted]ough the net. Each [redacted]ayer receiving your

[redacted]**eature**, and each feature [redacted]weighted sum of inputs [redacted]the “firing rate” of a b[redacted]ormula.

How then does Neurons or learning work?

So if that’s how a neuron *works*, let’s look at how it *learns*. In simple terms, **training** a neuron refers to iteratively updating the weights associated with each of its inputs so that it can progressively approximate the underlying relationship in the dataset it’s been given. Once properly trained, a neuron can be used to do things like correctly sort entirely new samples—say, images of cats and dogs—into separate buckets, just like people can. In machine learning terminology, this is known as **classification**.

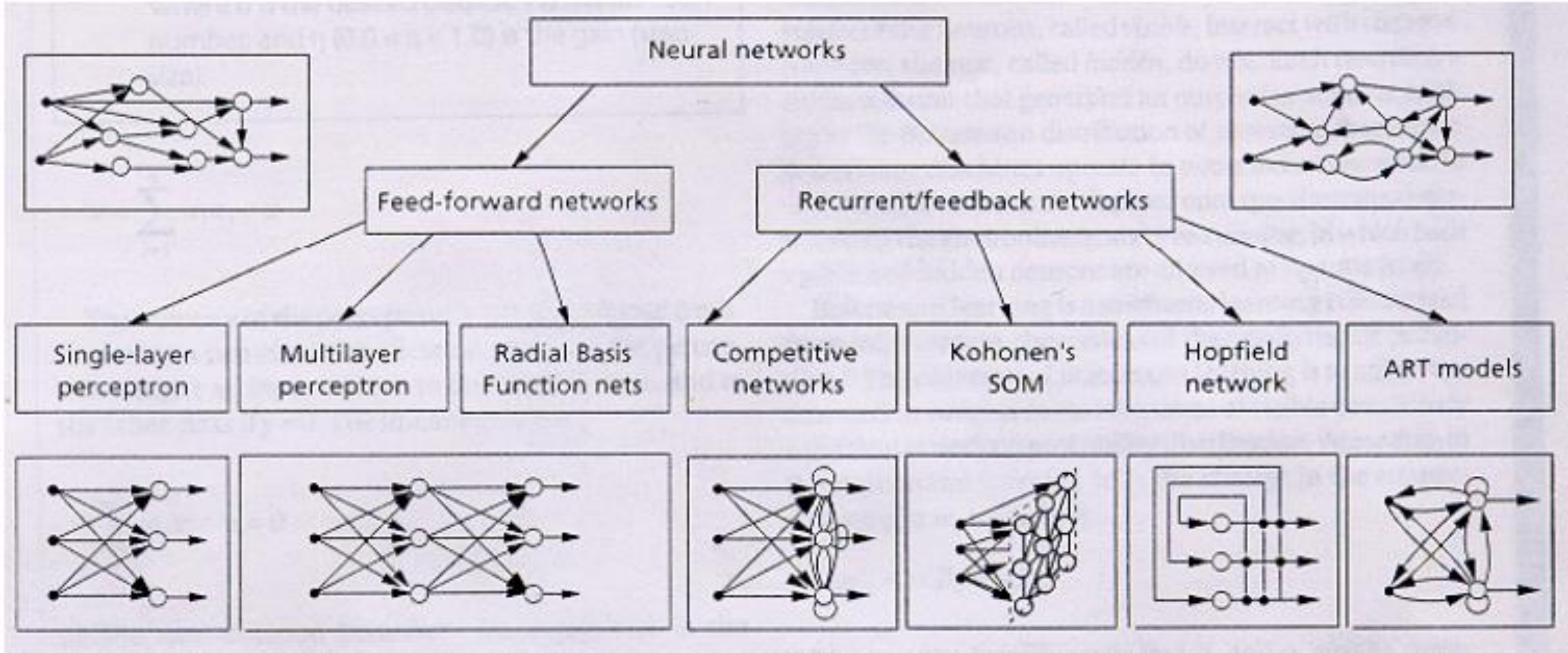
Why ANN

Neural networks are typically used to derive meaning from complex and non-linear data, detect and extract patterns which cannot be noticed by the human brain.

Some of the applications of neural network used these days are

- Pattern/ Image or object recognition
- Times series forecasting/ Classification
- Signal processing
- In self-driving cars to manage control
- Anomaly detection

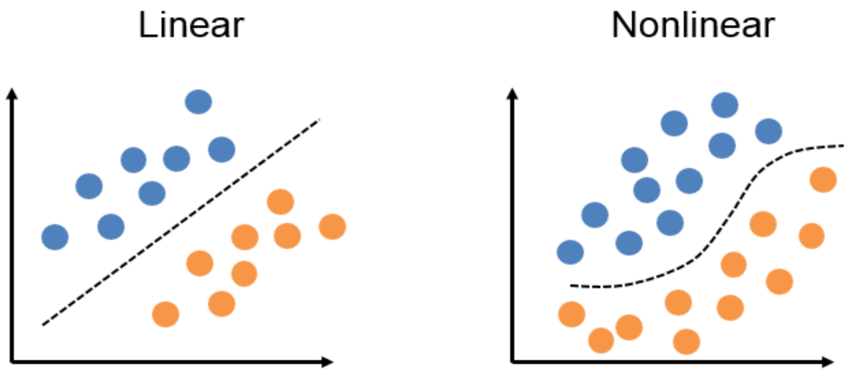
These applications fall into different types of neural networks such as convolutional neural network, recurrent neural networks, and feed-forward neural networks. The first one is more used in image recognition as it uses a mathematical process known as convolution to analyze images in non-literal ways.


Neural networks
Feed-forward networks
Recurrent/feedback networks
Single-layer perceptron
Multilayer perceptron
Radial Basis Function nets
Competitive networks
Kohonen's SOM
Hopfield network
ART models

Types of ANNs:

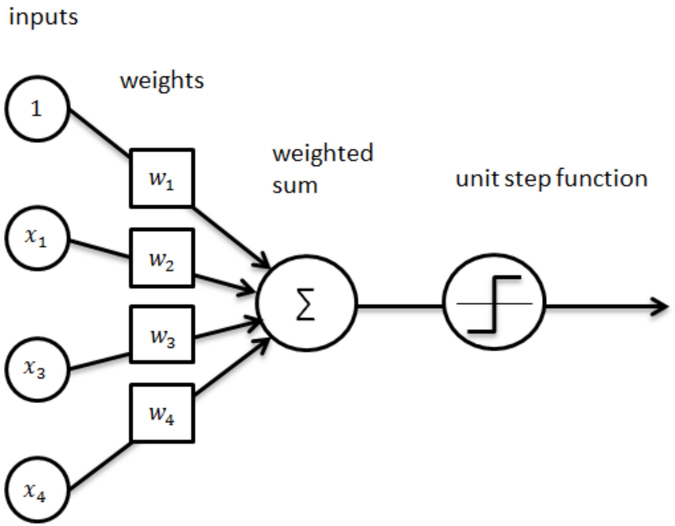
Perceptron:

The simplest and oldest model of an ANN, the Perceptron is a linear classifier used for binary predictions. This means that in order for it to work, the data must be linearly separable.



Types of ANNs:

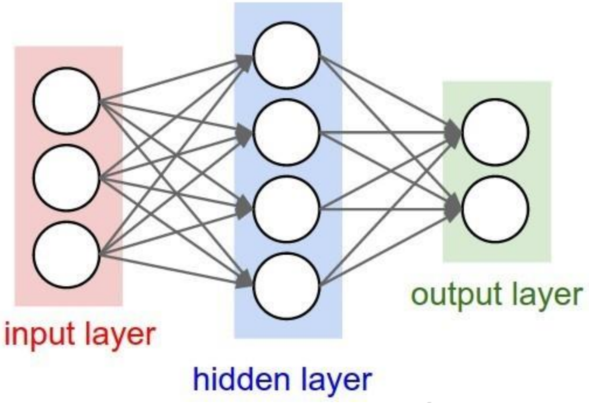
Perceptron: **Its Architecture:**



Multi-layer ANN:

More sophisticated than the perceptron, a Multi-layer ANN (e.g.: Convolutional Neural Network, Recurrent Neural Network etc ...) is capable of solving more complex classification and regression tasks thanks to its hidden layer(s).

Its Architecture:



How Do Neural Networks Differ From Conventional Computing?

To better understand artificial neural computing it is important to know first how a conventional 'serial' computer and it's software process information. A serial computer has a central processor that can address an array of memory locations where data and instructions are stored. Computations are made by the processor reading an instruction as well as any data the instruction requires from memory addresses, the instruction is then executed and the results are saved in a specified memory location as required. In a serial system (and a standard parallel one as well) the computational steps are deterministic, sequential and logical, and the state of a given variable can be tracked from one operation to another.

In comparison, ANNs are not sequential or necessarily deterministic. There are no complex central processors, rather there are many simple ones which generally do nothing more than take the weighted sum of their inputs from other processors. ANNs do not execute programed instructions; they respond in parallel (either simulated or actual) to the pattern of inputs presented to it. There are also no separate memory addresses for storing data. Instead, information is contained in the overall activation 'state' of the network. 'Knowledge' is thus represented by the network itself, which is quite literally more than the sum of its individual components.

What Applications Should Neural Networks Be Used For?

Neural networks are universal approximators, and they work best if the system you are using them to model has a high tolerance to error. One would therefore not be advised to use a neural network to balance one's cheque book! However they work very well for:

- capturing associations or discovering regularities within a set of patterns;
- where the volume, number of variables or diversity of the data is very great;
- the relationships between variables are vaguely understood; or,
- the relationships are difficult to describe adequately with conventional approaches.

What Are Their Limitations?

- Backpropagational neural networks (and many other types of networks) are in a sense the ultimate 'black boxes'. Apart from defining the general archetecture of a network and perhaps initially seeding it with a random numbers, the user has no other role than to feed it input and watch it train and await the output. In fact, it has been said that with backpropagation, "you almost don't know what you're doing". Some software freely available software packages (NevProp, bp, Mactivation) do allow the user to sample the networks 'progress' at regular time intervals, but the learning itself progresses on its own. The final product of this activity is a trained network that provides no equations or coefficients defining a relationship (as in regression) beyond it's own internal mathematics. The network 'IS' the final equation of the relationship.
- Backpropagational networks also tend to be slower to train than other types of networks and sometimes require thousands of epochs. If run on a truly parallel computer system this issue is not really a problem, but if the BPNN is being simulated on a standard serial machine (i.e. a single SPARC, Mac or PC) training can take some time. This is because the machines CPU must compute the function of each node and connection separately, which can be problematic in very large networks with a large amount of data. However, the speed of most current machines is such that this is typically not much of an issue.

What Are Their Advantages Over Conventional Techniques?

Depending on the nature of the application and the strength of the internal data patterns you can generally expect a network to train quite well. This applies to problems where the relationships may be quite dynamic or non-linear.

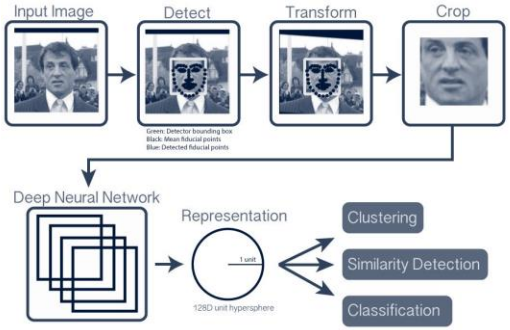
ANNs provide an analytical alternative to conventional techniques which are often limited by strict assumptions of normality, linearity, variable independence etc. Because an ANN can capture many kinds of relationships it allows the user to quickly and relatively easily model phenomena which otherwise may have been very difficult or imposible to explain otherwise.

Benefits of Neural Networks

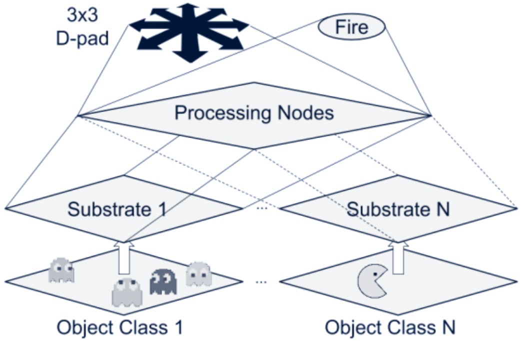
- **Nonlinearity:** An artificial neuron can be linear or nonlinear.
- **Input-Output Mapping:** A popular paradigm of learning, called *learning with a teacher*, or *supervised learning*, involves modification of the synaptic weights of a neural network by applying a set of labelled *training examples*, or *task examples*.
- **Adaptivity:** Neural networks have a built-in capability to *adapt* their synaptic weights to changes in the surrounding environment.
- **Evidential Response:** In the context of pattern classification, a neural network can be designed to provide information not only about which particular pattern to *select*, but also about the *confidence* in the decision made.
- **Fault Tolerance:** A neural network, implemented in hardware form, has the potential to be inherently *fault tolerant*, or capable of robust computation, in the sense that its performance degrades gracefully under adverse operating conditions.

Applications of neural networks

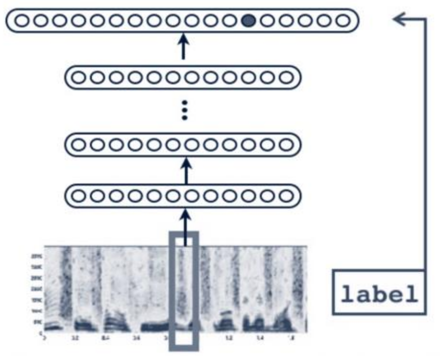
Object Recognition: Facial



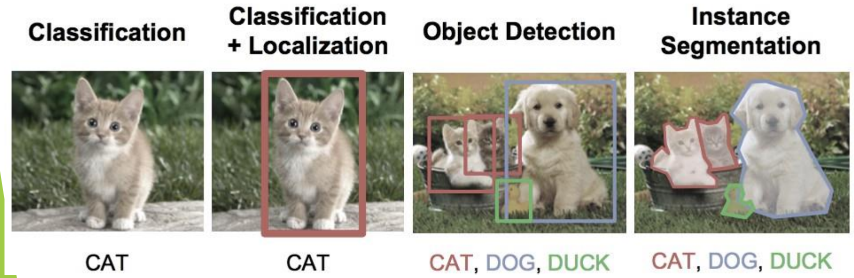
Game Playing



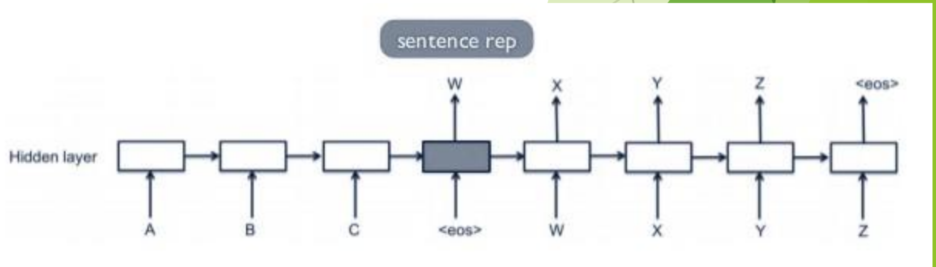
Object Recognition: Speech



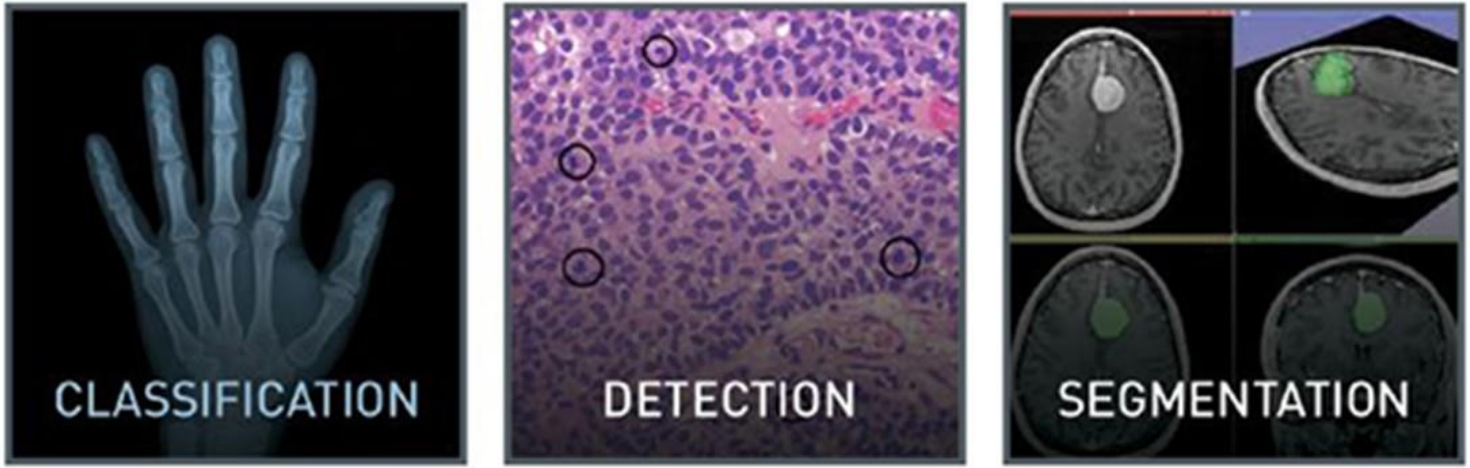
Object Classification



Language Translation



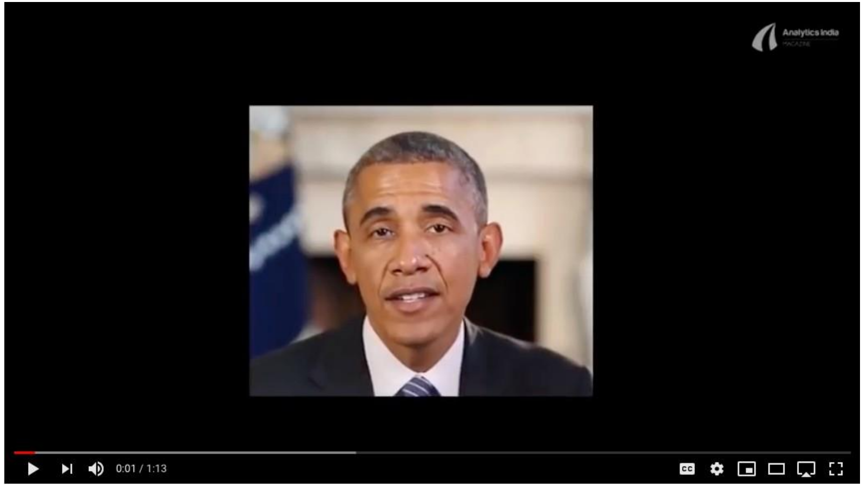
Healthcare



Source: http://bit.ly/2BXF5sR

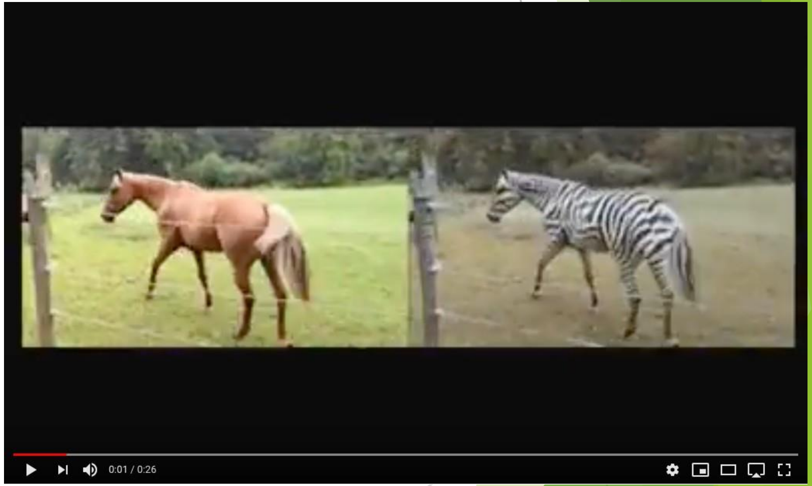
Faking videos

AI-generated "real fake" video of Barack Obama



Source: https://youtu.be/dkoi7sZvWiU

Turning a horse video into a zebra video in real time using GANs

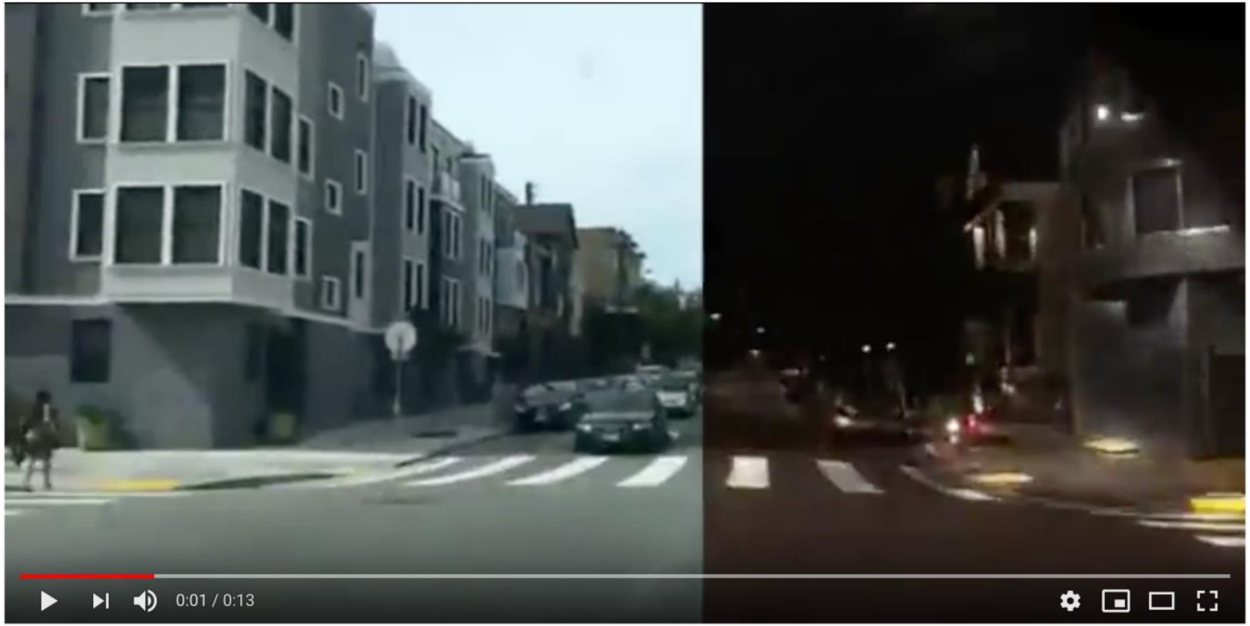


Source: https://youtu.be/JzgOfISLNjk

Self-driving cars

Source: https://youtu.be/URmxzxYlmtg?t=6m30s

Turning the day into night



Source: https://youtu.be/N7KbfWodXJE

Applications

- Natural language Processing
- Optical Character Recognition
- Speech recognition
- Neural Machine Translation
- Video Classification
- Emotion Recognition
- Face Recognition
- Object Detection
- Image Classification