Machine Learning (ML) Workshop

Dr Sara Soltaninejad
Fall 2021

Who am I?

- Sara Soltaninejad
- PhD in MRC, Computing Science Department, UofA, 2016-2020
  - PhD Thesis: Intelligent Parkinson's Disease Classification and Progress Monitoring Using Non-Invasive Techniques
  - Supervisors:
    - Dr Anup Basu
    - Dr Irene Cheng
- ML Developer AltaML, 2020-Now

ML Workshop Outline

Sep. 3

Part 1

- General introduction with ML

Sep. 10

Part 2

- ML workflow (Exploratory Data Analysis)

Sep. 17

Part 3

- ML workflow (Modelling)

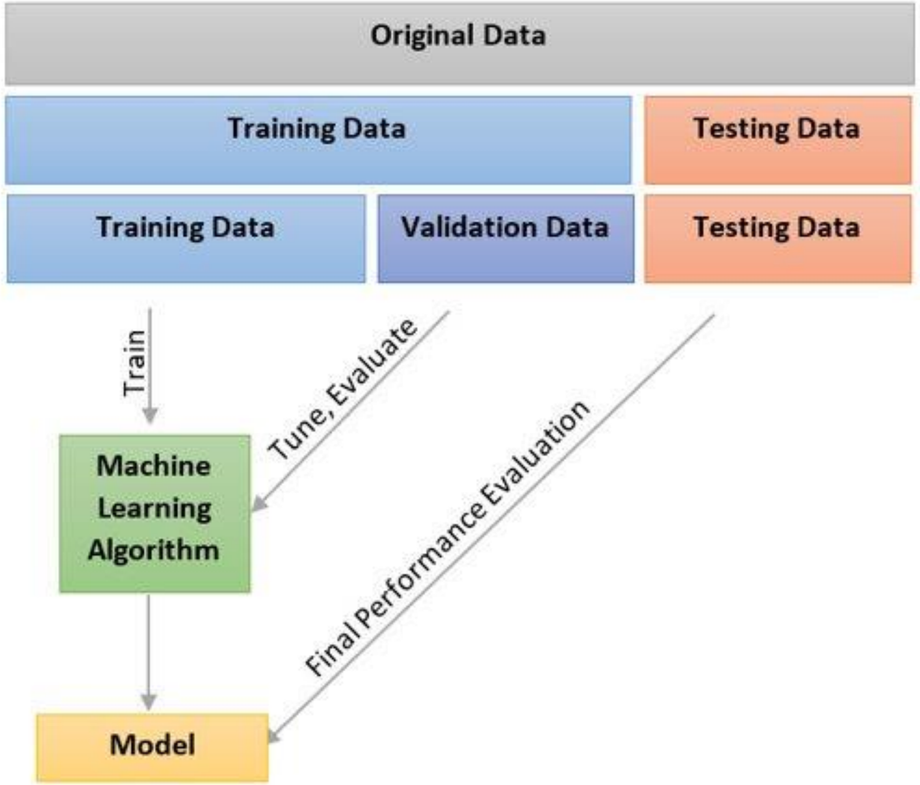
Sep. 24

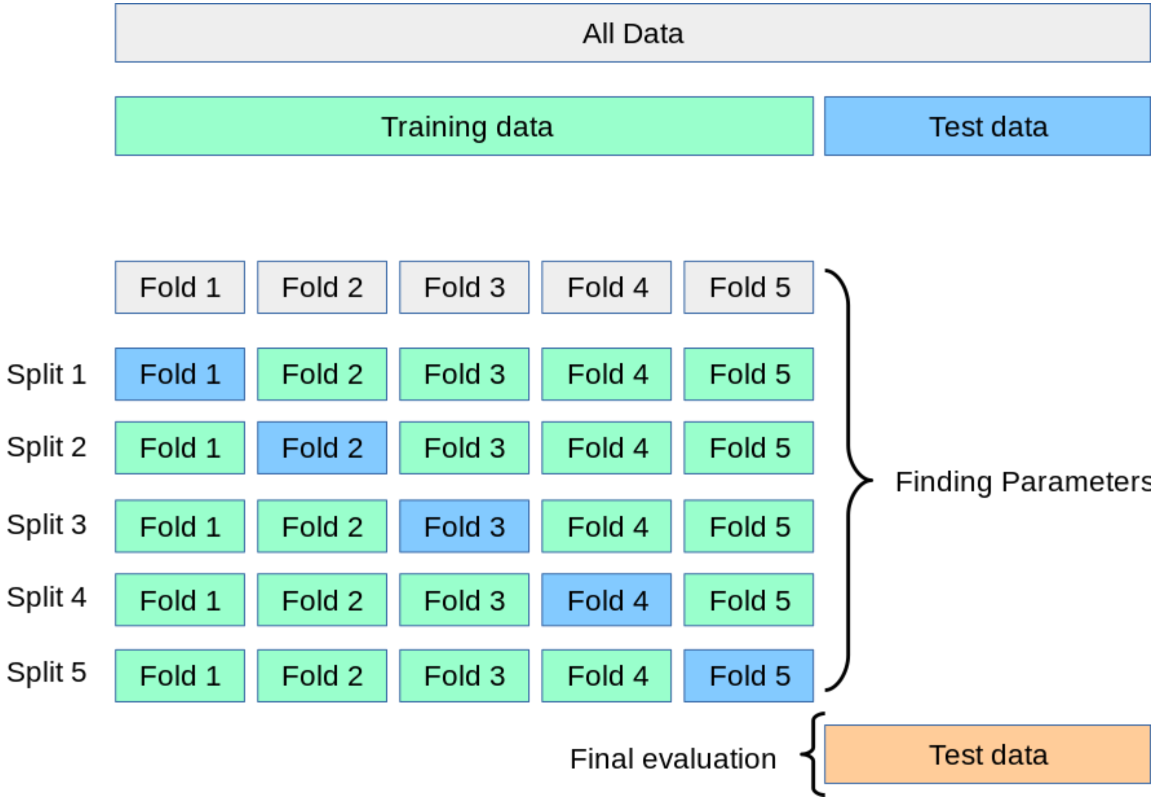
Part 4

- Deep Learning

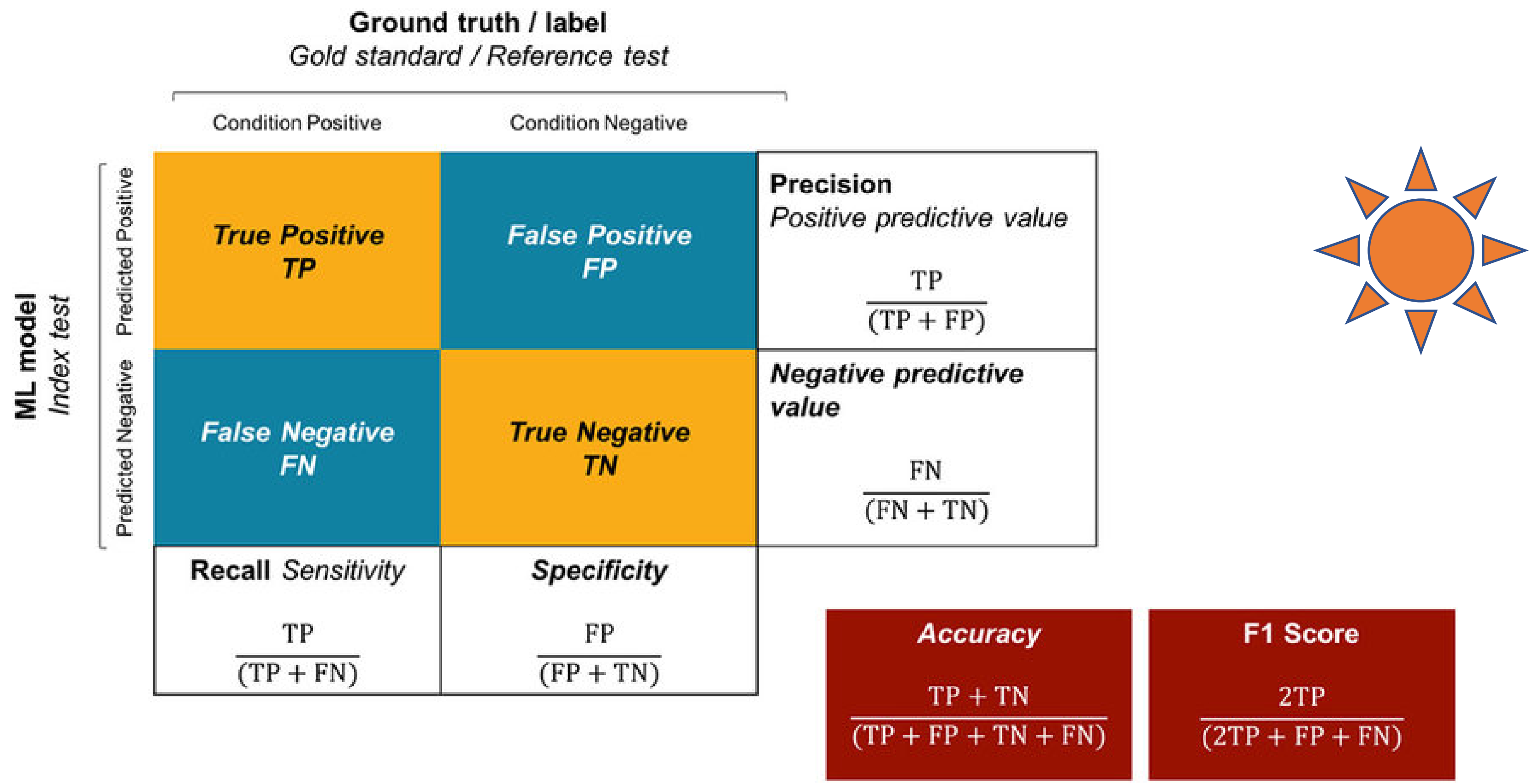
Left Over from the Previous Parts

Data Division





Evaluation

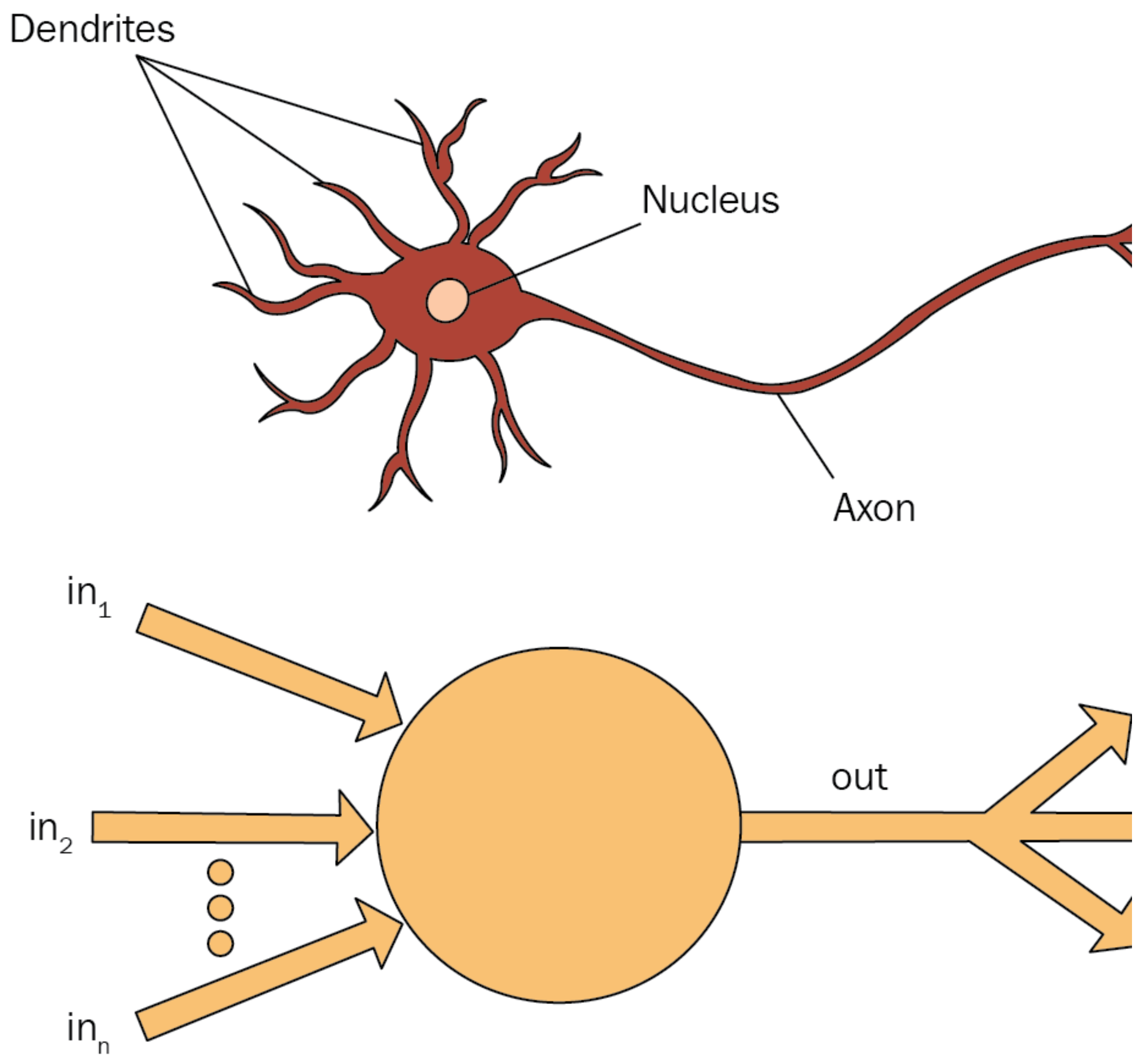


Ground truth / label
Gold standard / Reference test
Condition Positive
Condition Negative
ML model
Index test
Predicted Positive
Predicted Negative
True Positive TP
False Positive FP
False Negative FN
True Negative TN
Precision *Positive predictive value*
$\frac{TP}{(TP + FP)}$
Negative predictive value
$\frac{FN}{(FN + TN)}$
Recall *Sensitivity*
$\frac{TP}{(TP + FN)}$
Specificity
$\frac{FP}{(FP + TN)}$
Accuracy
$\frac{TP + TN}{(TP + FP + TN + FN)}$
F1 Score
$\frac{2TP}{(2TP + FP + FN)}$

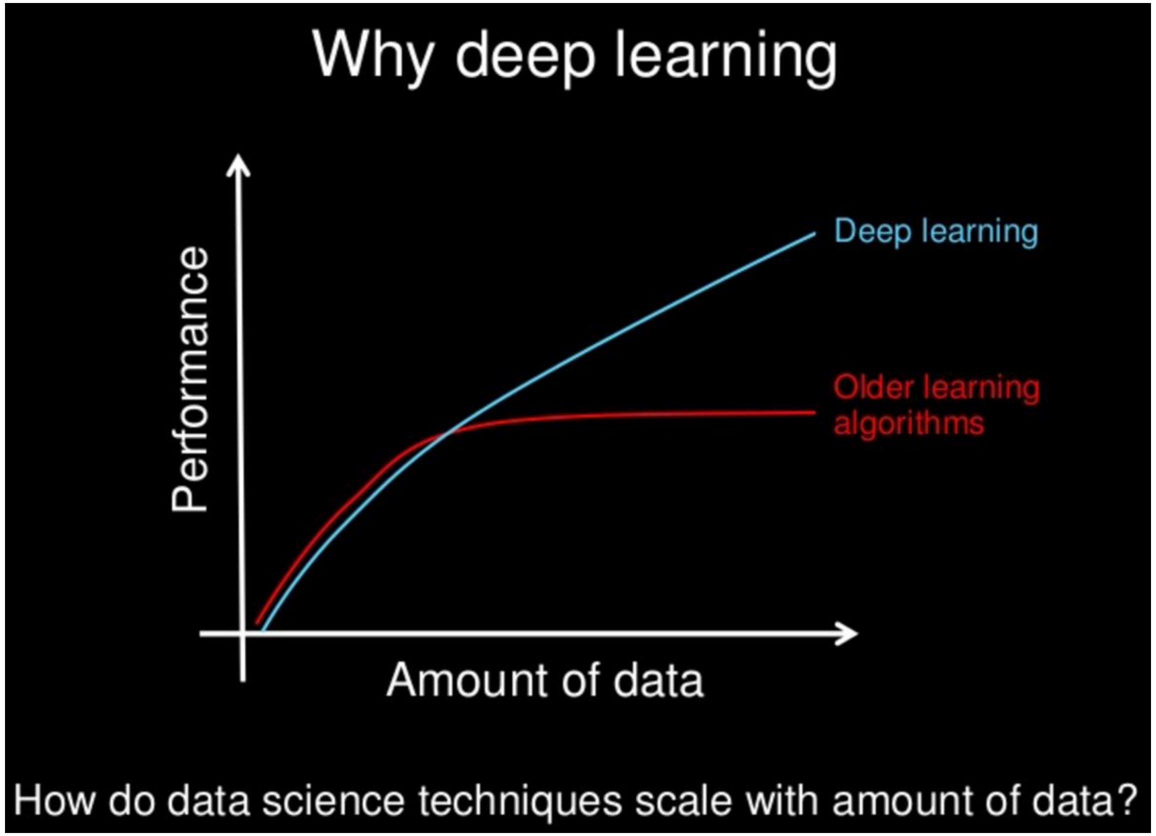
Part 4: Introduction to Deep Learning (DL)

What is Deep Learning

- Deep learning represents the very cutting edge of artificial intelligence (AI).
- Instead of teaching computers to process and learn from data (which is how machine learning works), with deep learning, the computer trains itself to process and learn from data. This is all possible thanks to layers of ANNs.
- The core of deep learning according to Andrew is that we now have fast enough computers and enough data to actually train large neural networks.

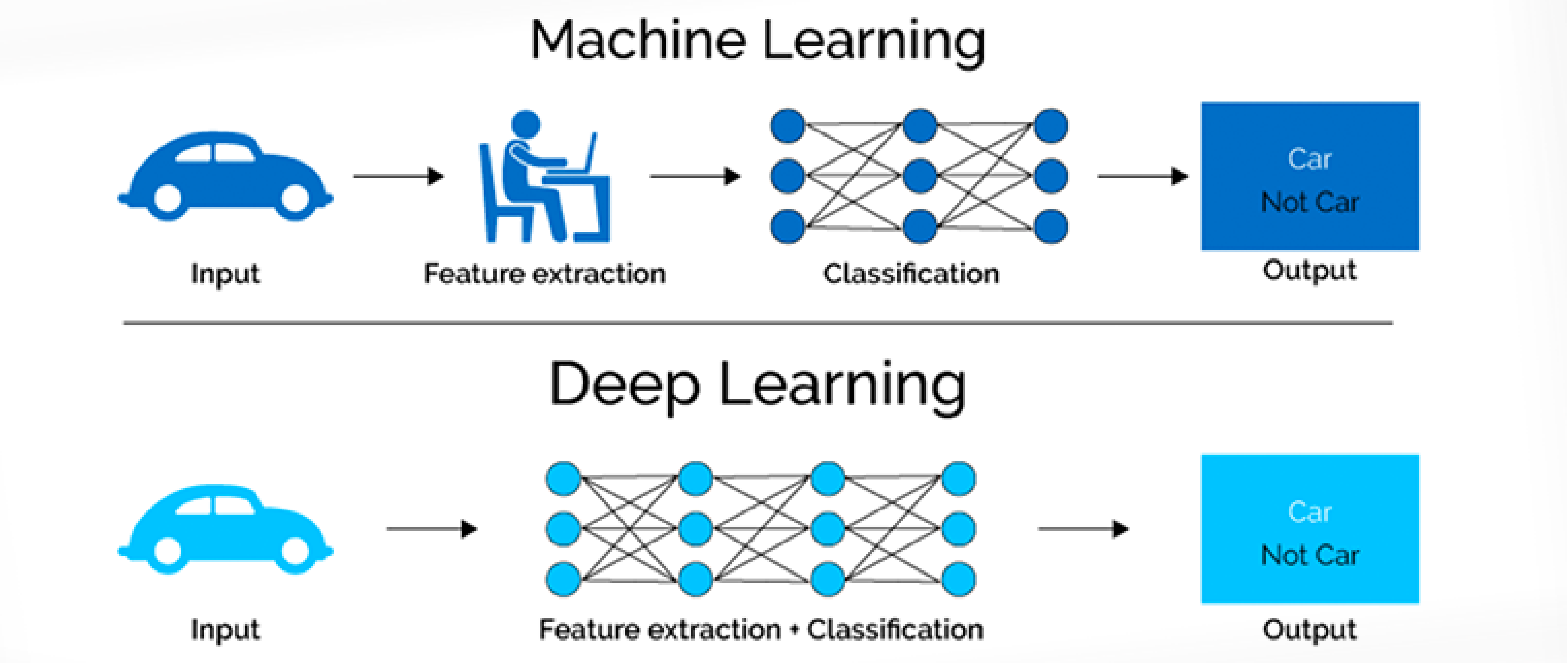


Deep Learning (DL) vs ML



	Deep Learning	Machine Learning
Data	Needs a big dataset	Performs well with a small to a medium dataset
Hardware requirements	Requires machines with GPU	Works with low-end machines
Engineering peculiarities	Needs to understand the basic functionality of the data	Understands the features and how they represent the data
Training time	Long	Short
Processing time	A few hours or weeks	A few seconds or hours
Number of algorithms	Few	Many
Data interpetation	Difficult	Some ML algorithms are easy to interpret, whereas some are hardly possible

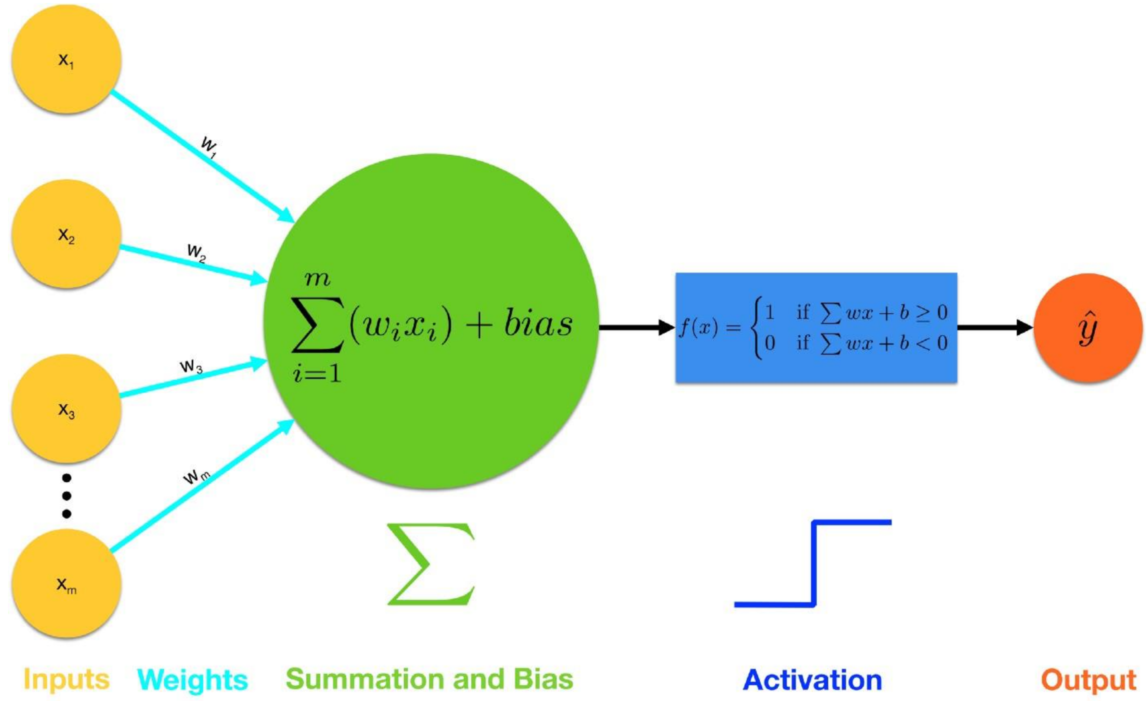
DL vs ML



Machine Learning
Car
Not Car
Input
Feature extraction
Classification
Output
Deep Learning
Car
Not Car
Input
Feature extraction + Classification
Output

Neural Network

- Neural networks are the workhorses of deep learning.
- Neural networks are a set of algorithms, modeled loosely after the human brain, that are designed to recognize patterns.
- A Neuron is the smallest unit of neural network which implements a mathematical function relevant to the network in context.
- A bias is an extra connection that is constant to the neuron on the given data.
- Activation function is a function that describes a rule for the neuron. It can be something like output ***1*** whenever the value is above a certain threshold and ***0*** otherwise.



NN: Training

For training a neural network the most important thing to remember is the Loss function.

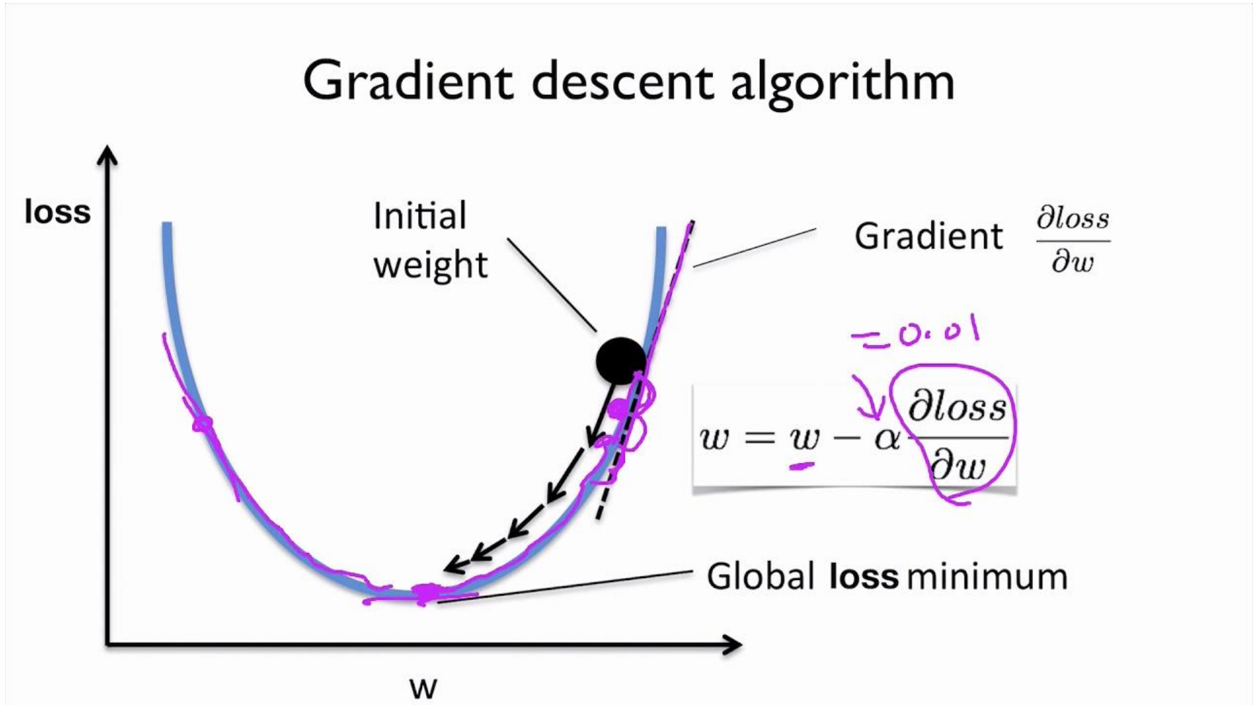
A Loss function evaluates the loss of the model, i.e how bad the model is performing.

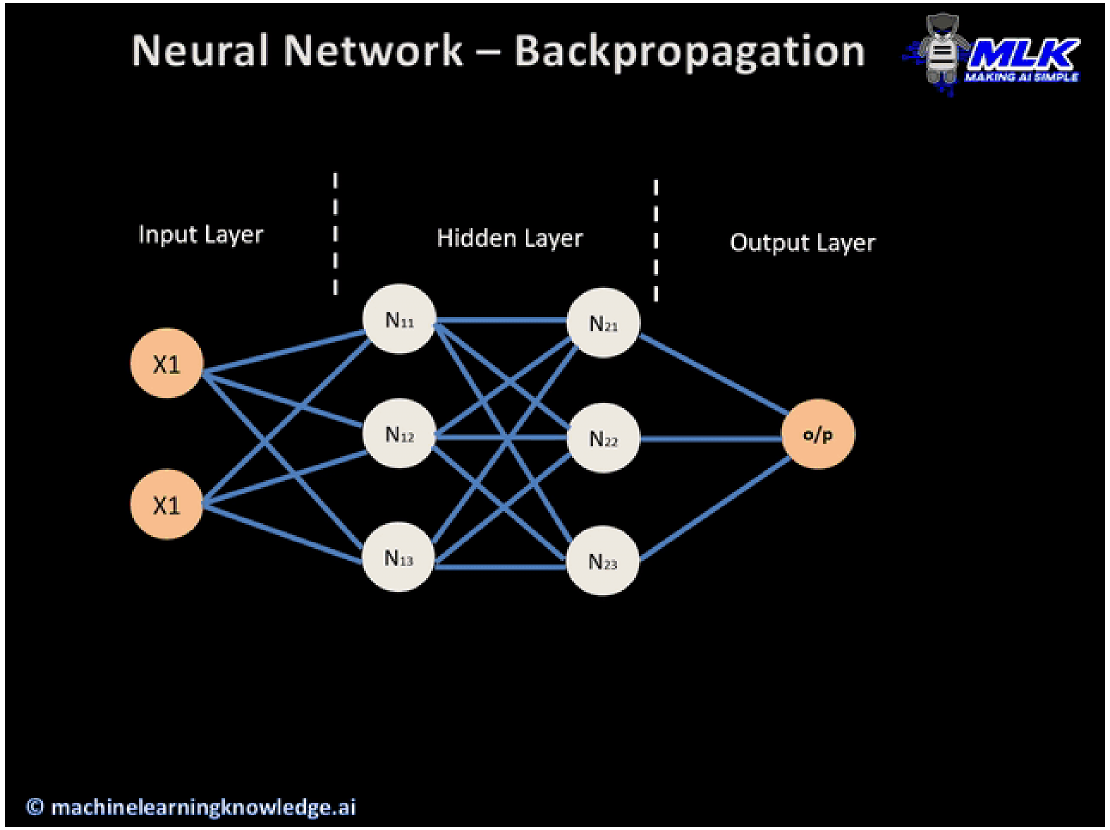
Training goal is to minimize the Loss value in each iteration for the model by modifying the weights of the model which is done by **Backpropagation** through time and using a Gradient descent optimization algorithm.

NN: Training

Back Propagation Through Time (BPTT): BPTT is a continuous loop where after each iteration the algorithm reverses back to modify the weights of its connections, and this modification is done by Gradient Descent Algorithm.

Network Training **Gradient Descent**: This algorithm tries to minimize the loss function by finding the local minima. It takes small steps in the direction of the negative gradient to hit a local minimum.



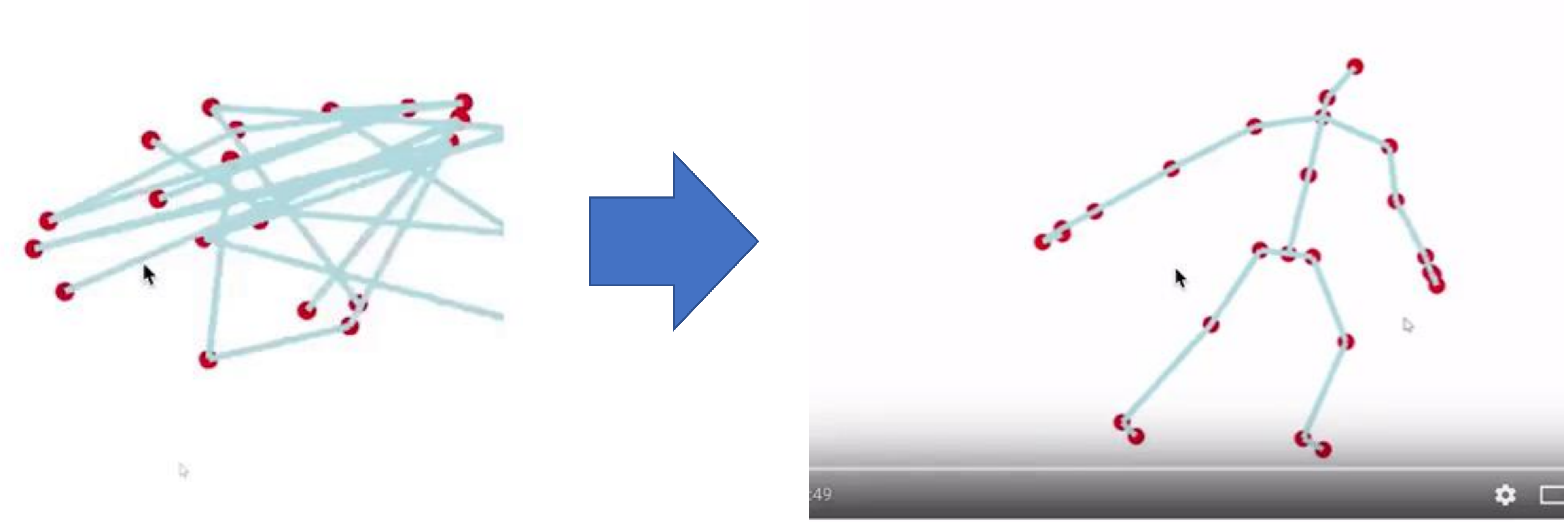


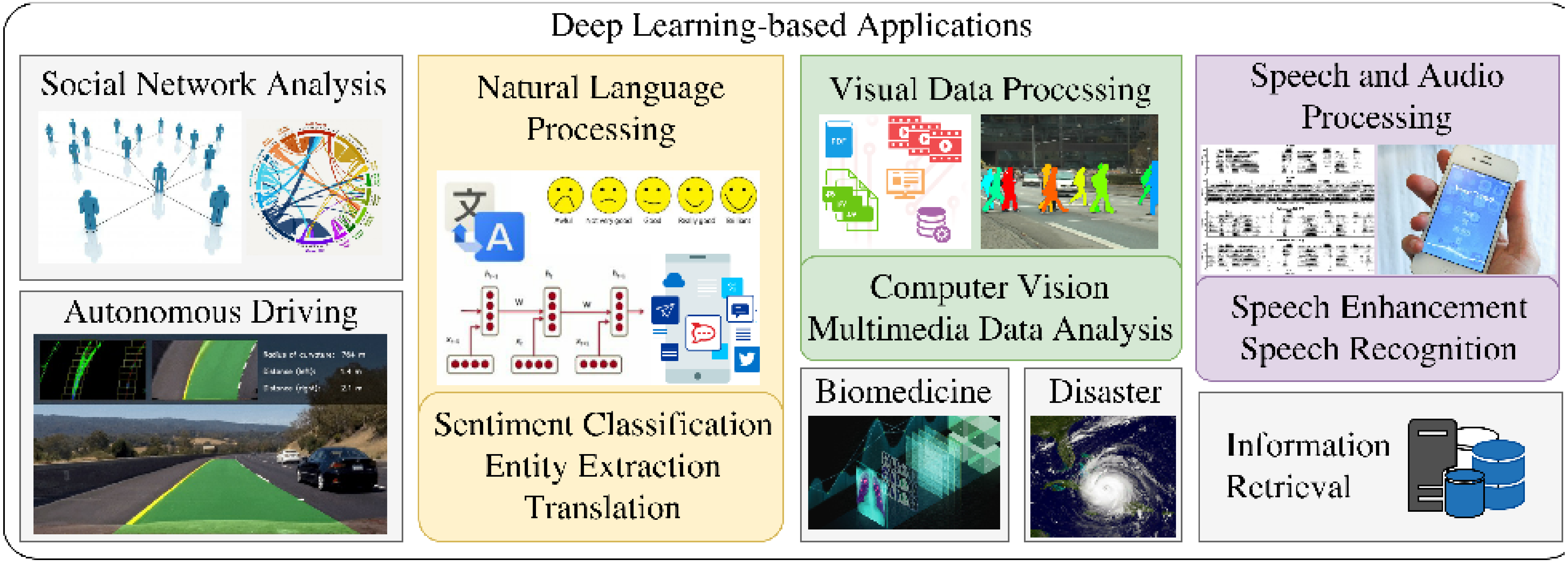
DL Workflow

- Each layer represents a deeper level of knowledge, i.e., the hierarchy of knowledge. A neural network with four layers will learn more complex feature than with that with two layers.
- learning occurs in two phases.
  - The first phase consists of applying a nonlinear transformation of the input and create a statistical model as output.
  - The second phase aims at improving the model with a mathematical method known as derivative.

DL Example

the model is trying to learn how to dance.





DL Applications

DL Models

At a high-level, neural networks are either encoders, decoders, or a combination of both:

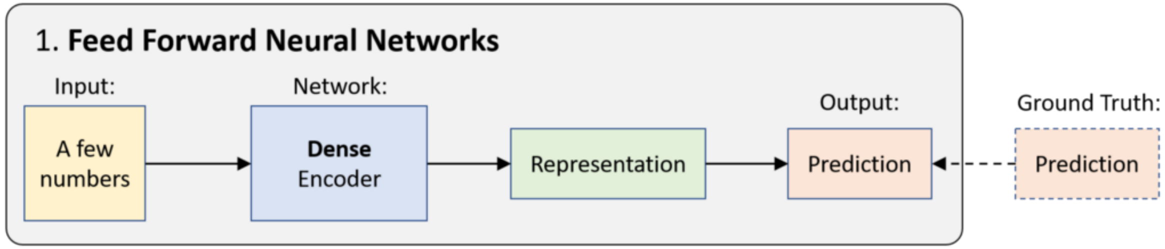
Encoders find patterns in raw data to form compact, useful representations.

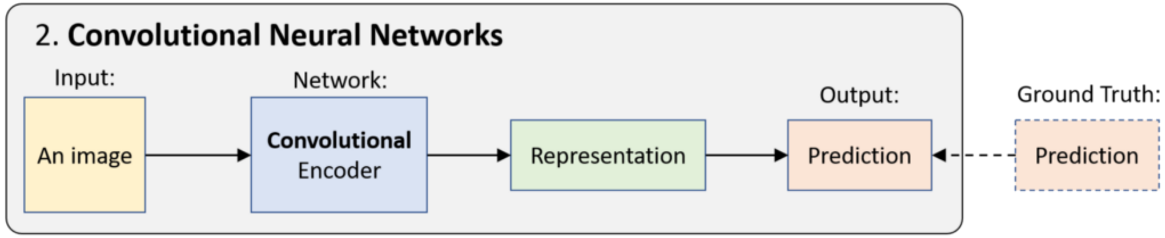
Decoders generate high-resolution data from those representations.

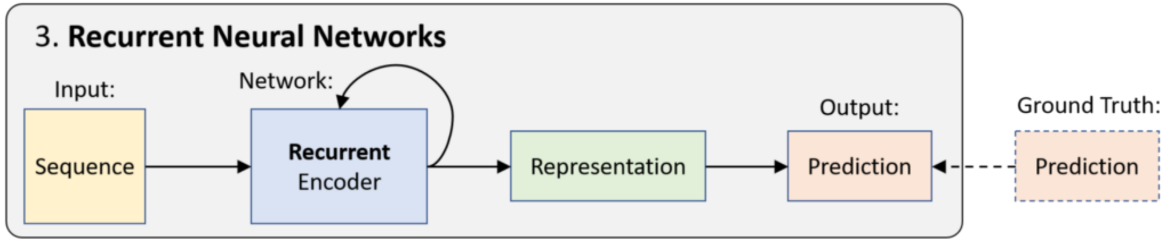
The rest is clever methods that help us deal effectively with visual information, language, audio (#1–6) and even act in a world based on this information and occasional rewards (#7).

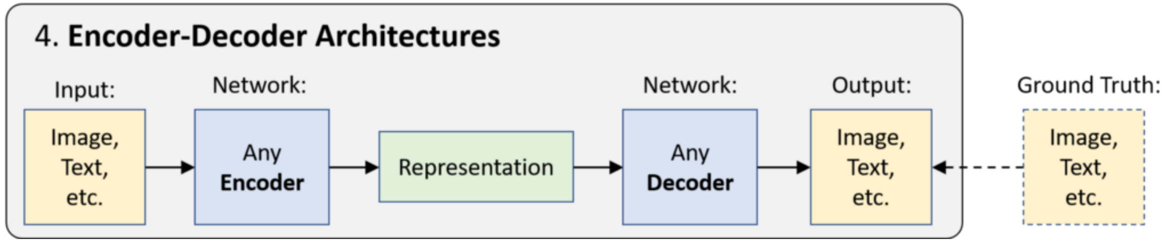
DL Models

Supervised Learning

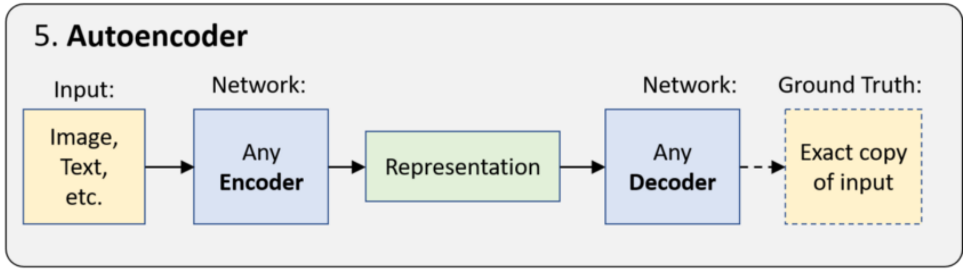


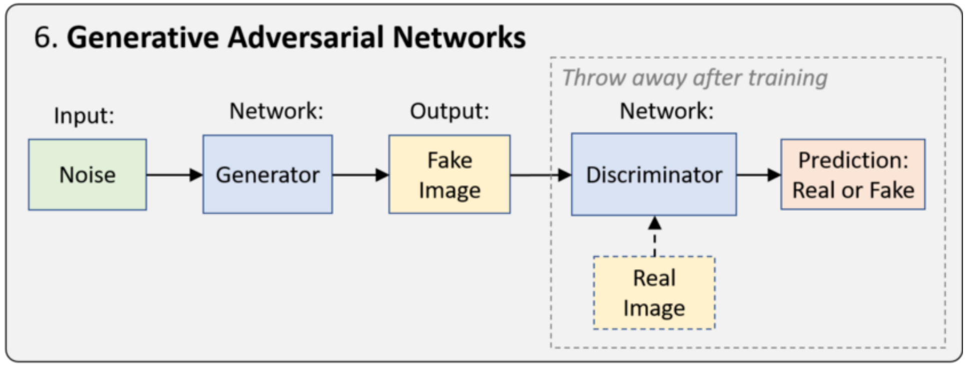




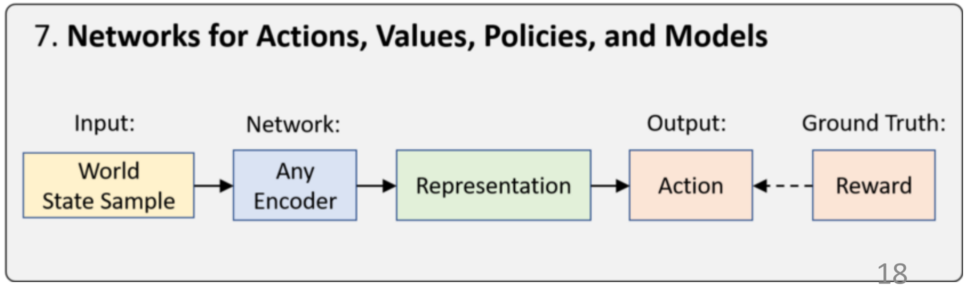


Unsupervised Learning





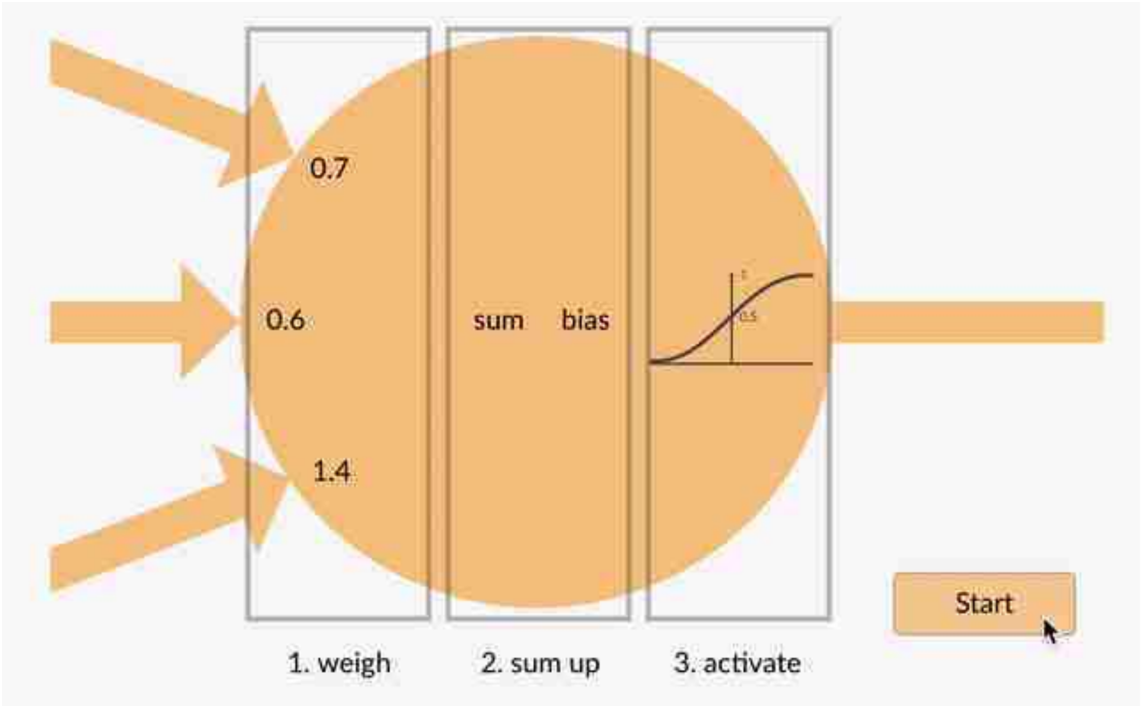
Reinforcement Learning

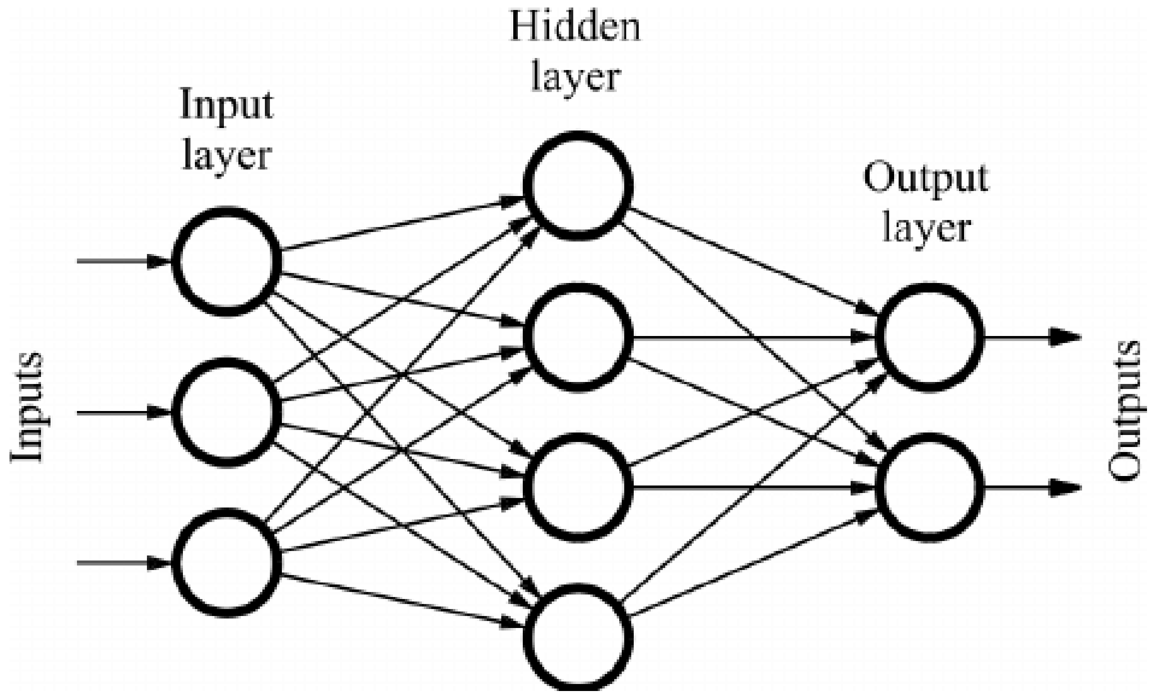


FFNN

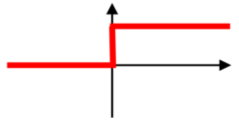
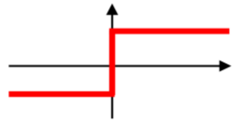
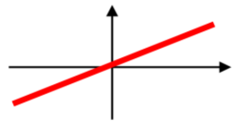
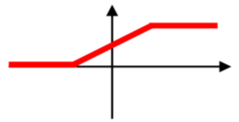
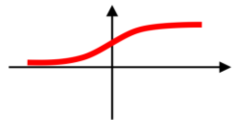
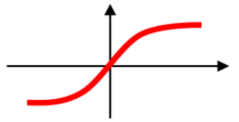
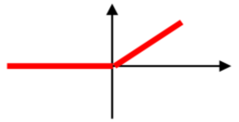
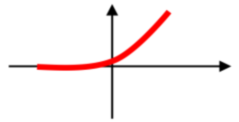
The feedforward neural network was the first and simplest type of artificial neural network devised.

In this network, the information moves in only one direction, forward, from the input nodes, through the hidden nodes (if any) and to the output nodes. There are no cycles or loops in the network.



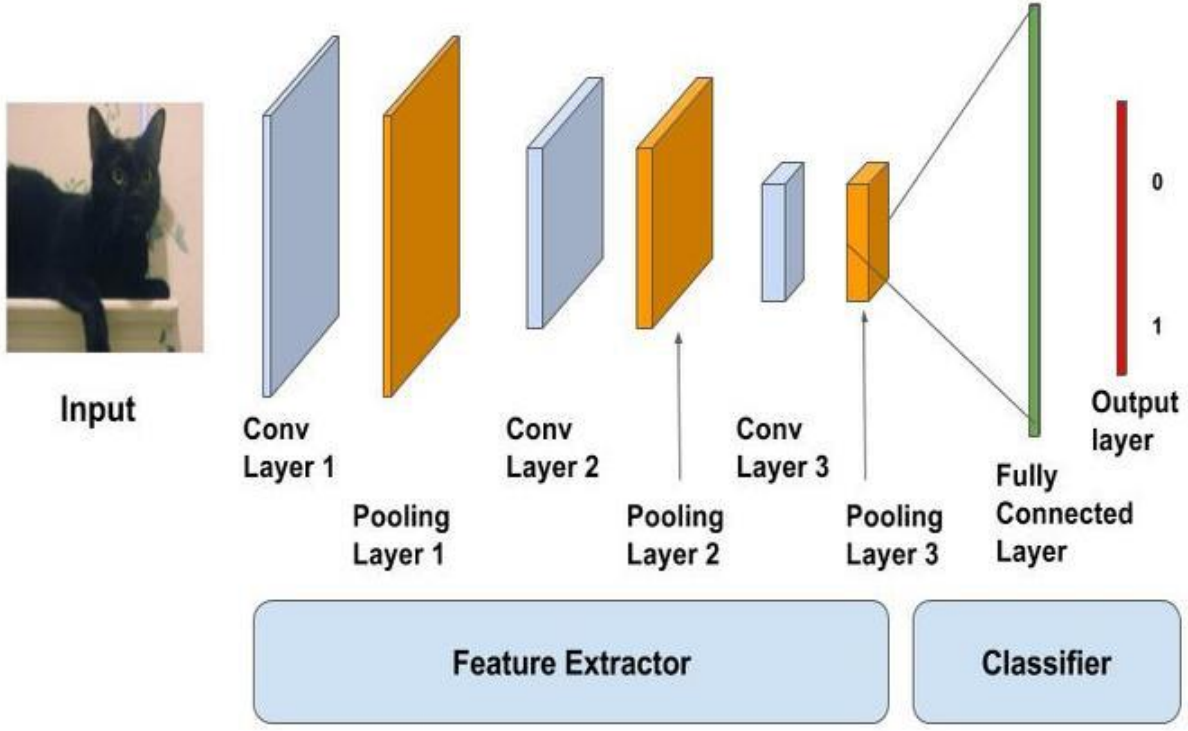


Activation Functions

Activation function	Equation	Example	1D Graph
Unit step (Heaviside)	$\phi(z) = \begin{cases} 0, & z < 0, \\ 0.5, & z = 0, \\ 1, & z > 0, \end{cases}$	Perceptron variant	
Sign (Signum)	$\phi(z) = \begin{cases} -1, & z < 0, \\ 0, & z = 0, \\ 1, & z > 0, \end{cases}$	Perceptron variant	
Linear	$\phi(z) = z$	Adaline, linear regression	
Piece-wise linear	$\phi(z) = \begin{cases} 1, & z \geq \frac{1}{2}, \\ z + \frac{1}{2}, & -\frac{1}{2} < z < \frac{1}{2}, \\ 0, & z \leq -\frac{1}{2}, \end{cases}$	Support vector machine	
Logistic (sigmoid)	$\phi(z) = \frac{1}{1 + e^{-z}}$	Logistic regression, Multi-layer NN	
Hyperbolic tangent	$\phi(z) = \frac{e^{z} - e^{-z}}{e^{z} + e^{-z}}$	Multi-layer Neural Networks	
Rectifier, ReLU (Rectified Linear Unit)	$\phi(z) = max(0, z)$	Multi-layer Neural Networks	
Rectifier, softplus	$\phi(z) = \ln(1 + e^{z})$	Multi-layer Neural Networks	

CNN

- CNNs (aka ConvNets) are feed forward neural networks that use a spatial-invariance trick to efficiently learn local patterns, most commonly, in images.
- Spatial-invariance means that a cat ear in the top left of the image has the same features as a cat ear in bottom right of the image.
- CNNs share weights across space to make the detection of cat ears and other patterns more efficient.

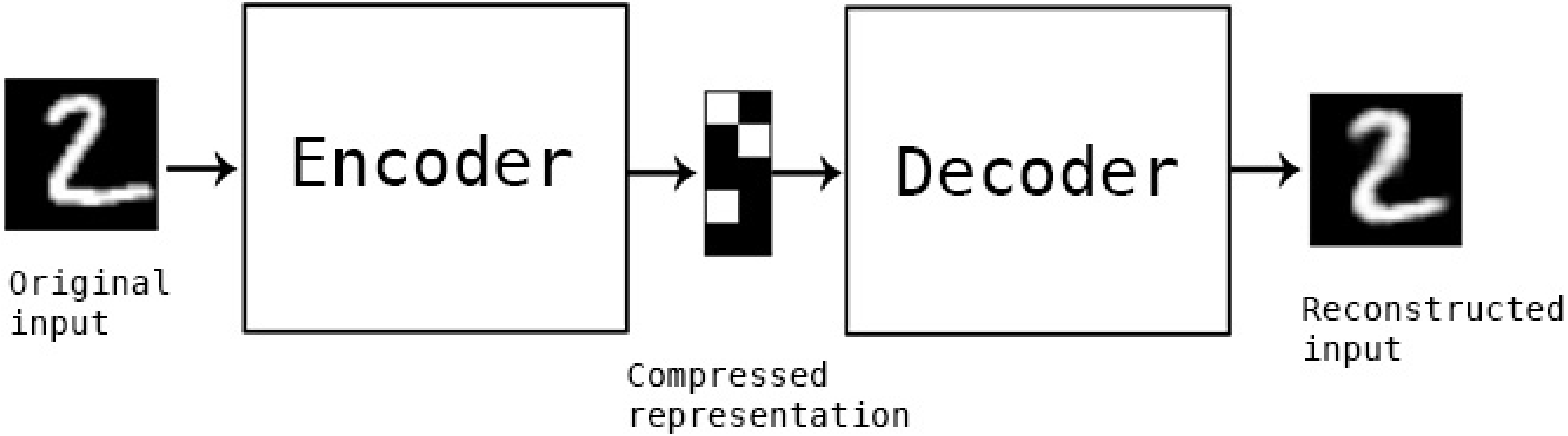


CNN Components

- Input layer — a single raw image is given as an input.
- A convolution layer - a convolution layer is a matrix of dimension smaller than the input matrix. It performs a convolution operation with a small part of the input matrix having same dimension. The sum of the products of the corresponding elements is the output of this layer.
- ReLU or Rectified Linear Unit — ReLU is mathematically expressed as max(0,x)
- Maxpool — Maxpool passes the maximum value from amongst a small collection of elements of the incoming matrix to the output. Usually it is a square matrix.
- Fully connected layer — The final output layer is a normal fully-connected neural network layer, which gives the output.

CNN Models

- LeNet — Developed by Yann LeCun to recognize handwritten digits is the pioneer CNN.
- AlexNet — Developed by Alex Krizhevsky, Ilya Sutskever and Geoff Hinton won the 2012 ImageNet challenge. It is the first CNN where multiple convolution operations were used.
- GoogleLeNet — Developed by Google, won the 2014 ImageNet competition. The main advantage of this network over the other networks was that it required a lot lesser number of parameters to train, making it faster and less prone to overfitting.
- VGGNet — This is another popular network, with its most popular version being VGG16. VGG16 has 16 layers which includes input, output and hidden layers.
- ResNet — Developed by Kaiming He, this network won the 2015 ImageNet competition. The 2 most popular variant of ResNet are the ResNet50 and ResNet34. Another complex variation of ResNet is ResNeXt architecture.



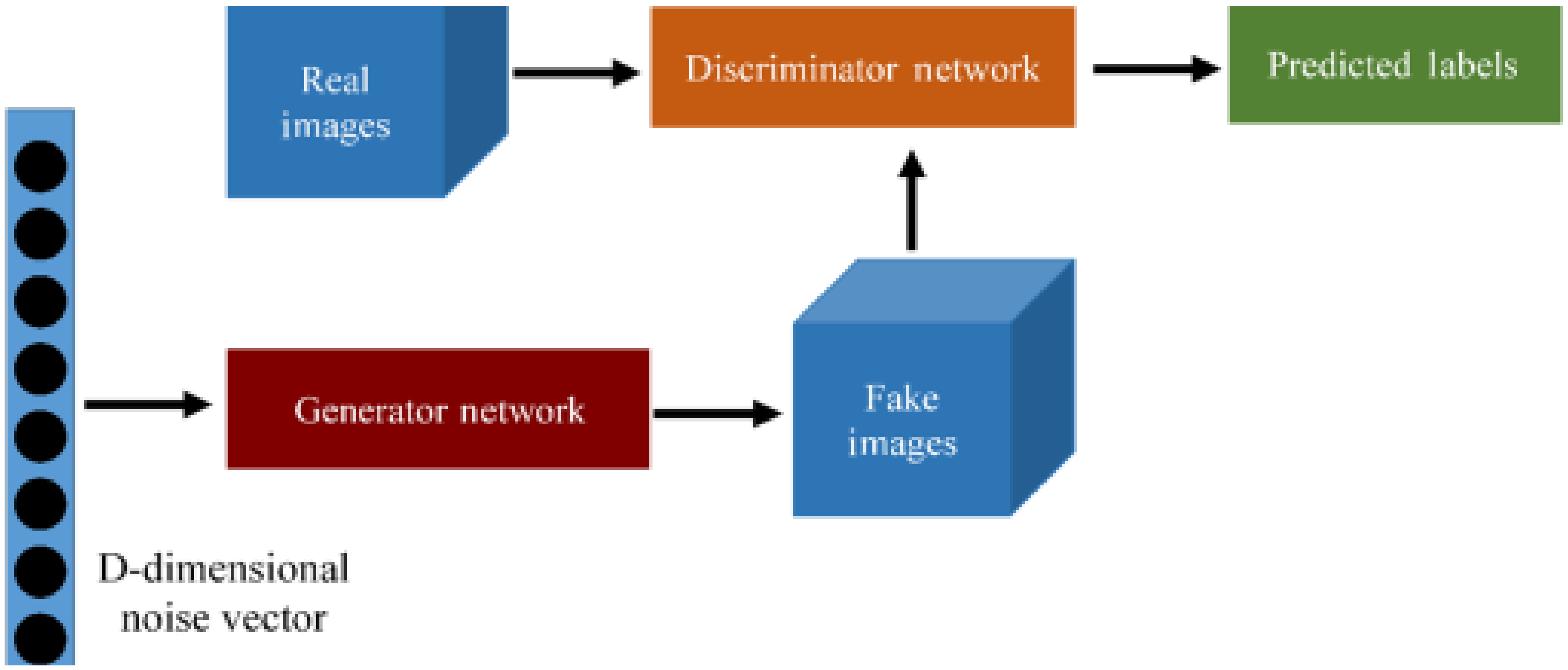
Autoencoder

- An autoencoder is a special type of neural network that is trained to copy its input to its output.
- For example, given an image of a handwritten digit, an autoencoder first encodes the image into a lower dimensional latent representation, then decodes the latent representation back to an image.
- An autoencoder learns to compress the data while minimizing the reconstruction error.

GAN

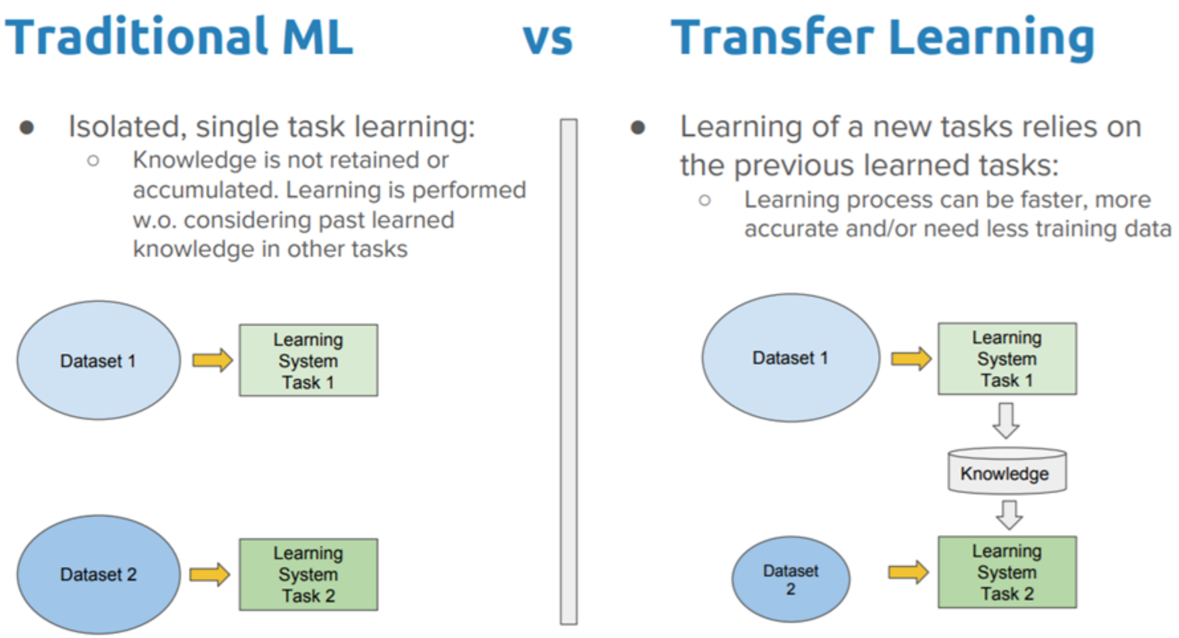
- Generative adversarial networks (GANs) are an exciting recent innovation in deep learning.
- The GAN architecture was first described in the 2014 paper by Ian Goodfellow, et al. titled "Generative Adversarial Networks."
- GANs are *generative* models create new data instances that resemble your training data.
- For example, GANs can create images that look like photographs of human faces, even though the faces don't belong to any real person. These images were created by a GAN

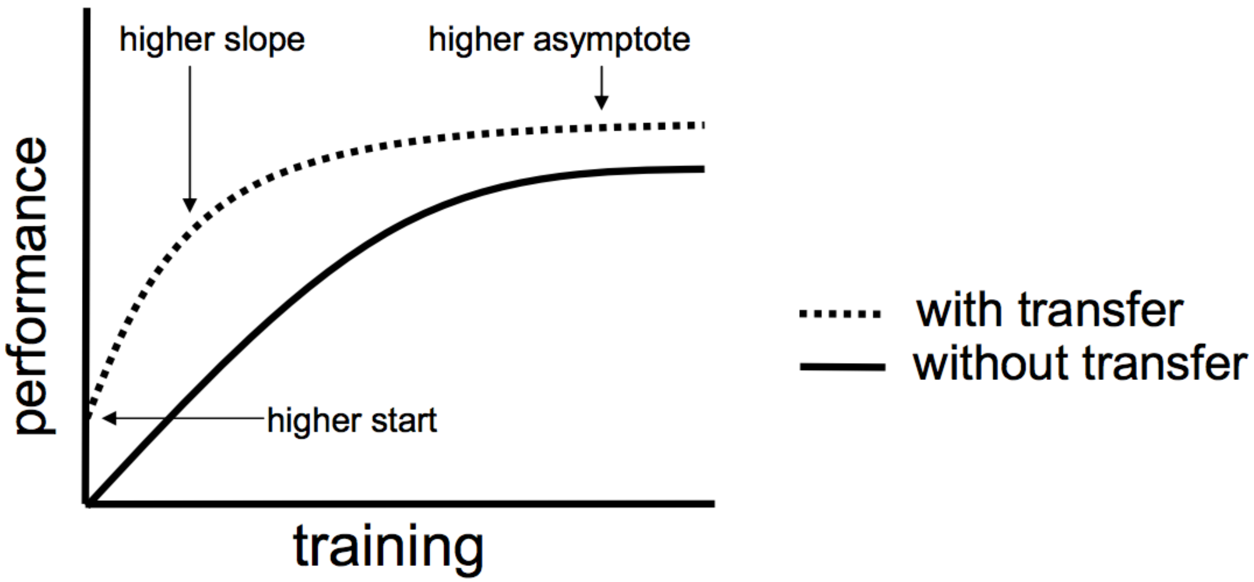
GAN


Real images
Discriminator network
Predicted labels
Generator network
Fake images
D-dimensional noise vector

GAN Applications

- Generate Examples for Image Datasets
- Generate Photographs of Human Faces
- Generate Realistic Photographs
- Generate Cartoon Characters
- Image-to-Image Translation
- Text-to-Image Translation
- Semantic-Image-to-Photo Translation
- Face Frontal View Generation
- Generate New Human Poses
- Photos to Emojis
- Photograph Editing
- Face Aging
- Photo Blending
- Super Resolution
- Photo Inpainting
- Clothing Translation
- Video Prediction
- 3D Object Generation





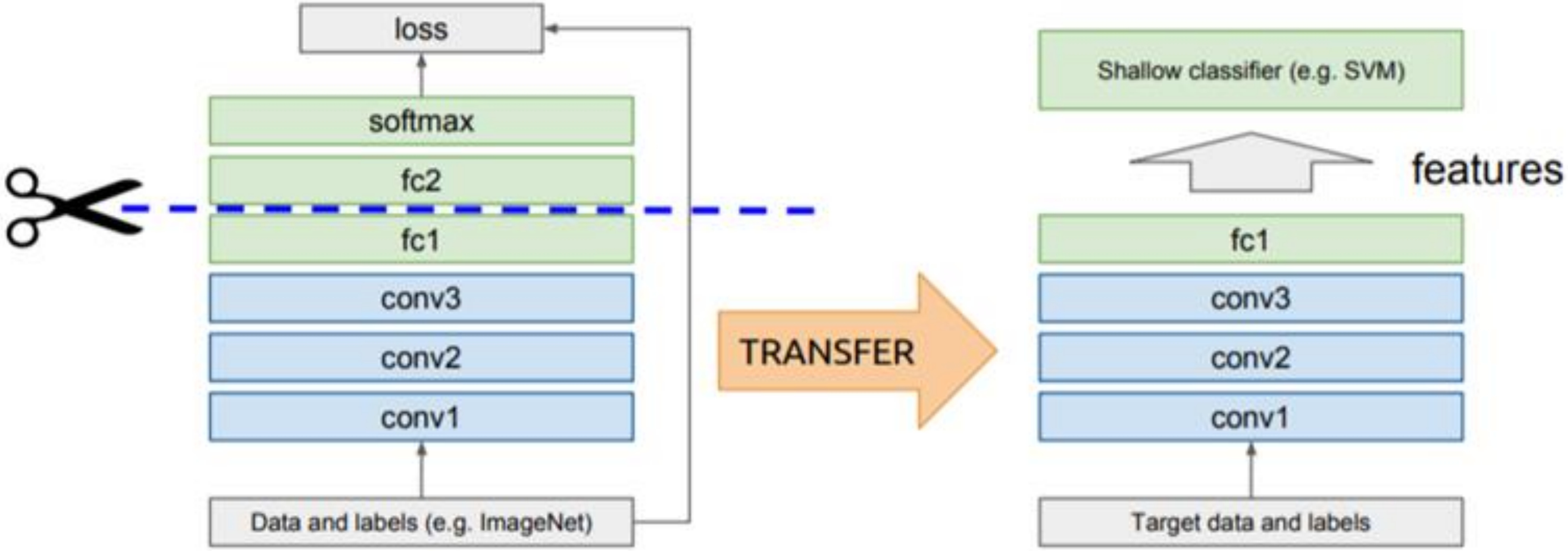
Transfer Learning

- Transfer learning is a machine learning method where a model developed for a task is reused as the starting point for a model on a second task.

Off-the-shelf Pre-trained Models as Feature Extractors

Idea: use outputs of one or more layers of a network trained on a different task as generic feature detectors. Train a new shallow model on these features.

Assumes that $D_S = D_T$



Quiz

- Develop a GAN for Generating MNIST Handwritten Digits.

- Explore different types of autoencoders and try to implement at least one of them in python and keras.

- Investigate the RNN and LSTM, and see how does back propagation works there?

References

- http://deeplearning.net/tutorial/
- https://medium.com/tensorflow/mit-deep-learning-basics-introduction-and-overview-with-tensorflow-355bcd26baf0
- https://www.guru99.com/deep-learning-tutorial.html
- https://news.mit.edu/2017/explained-neural-networks-deep-learning-0414
- https://wiki.pathmind.com/backpropagation
- https://bernardmarr.com/default.asp?contentID=1789
- https://medium.com/datadriveninvestor/convolution-neural-networks-vs-fully-connected-neural-networks-8171a6e86f15
- https://www.tensorflow.org/tutorials/generative/autoencoder
- https://machinelearningmastery.com/impressive-applications-of-generative-adversarial-networks/
- https://machinelearningmastery.com/how-to-use-transfer-learning-when-developing-convolutional-neural-network-models/
- https://machinelearningmastery.com/transfer-learning-for-deep-learning/
- https://towardsdatascience.com/a-comprehensive-hands-on-guide-to-transfer-learning-with-real-world-applications-in-deep-learning-212bf3b2f27a