AI & Robotics

Neural Networks

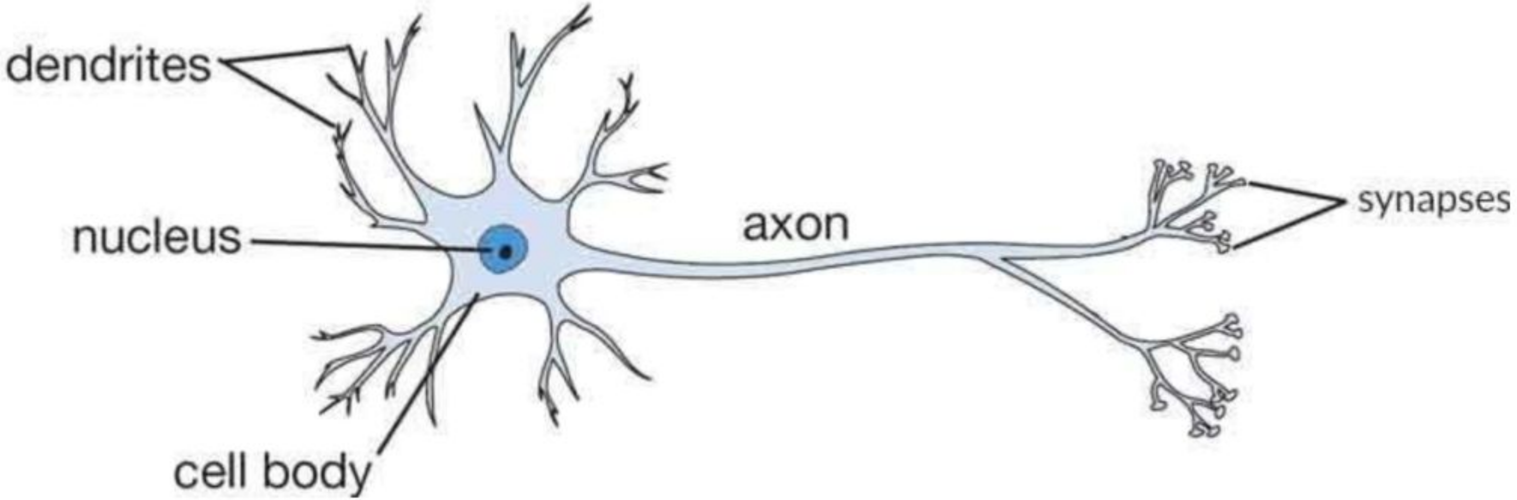
AI & ROBOTICS LAB

Goals

The **junior-colleague**

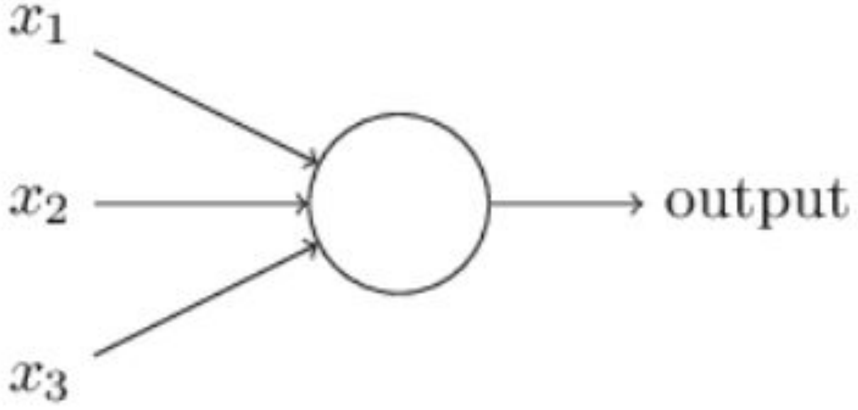
- can explain how a perceptron works in their own words
- can explain the importance of the activation function in the context of neural networks
- can explain the limitations of perceptrons
- can explain how Neural Networks can handle nonlinear separation
- can describe the concept of gradient descent in the context of neural networks in their own words
- can explain the importance of the learning rate in the context of gradient descent
- can explain how backpropagation works in their own words
- can explain how a Neural Networks is trained in their own words

Biological Neuron

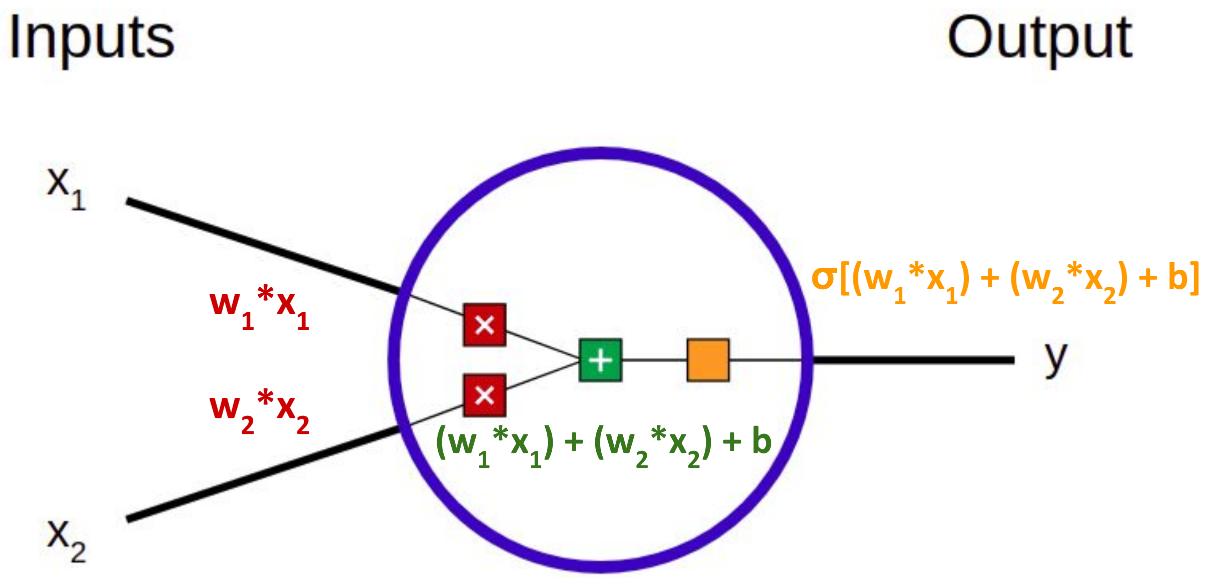


The human brain has some 1011 (one hundred billion) neurons with on average 7,000 synaptic connections to other neurons.

Perceptron

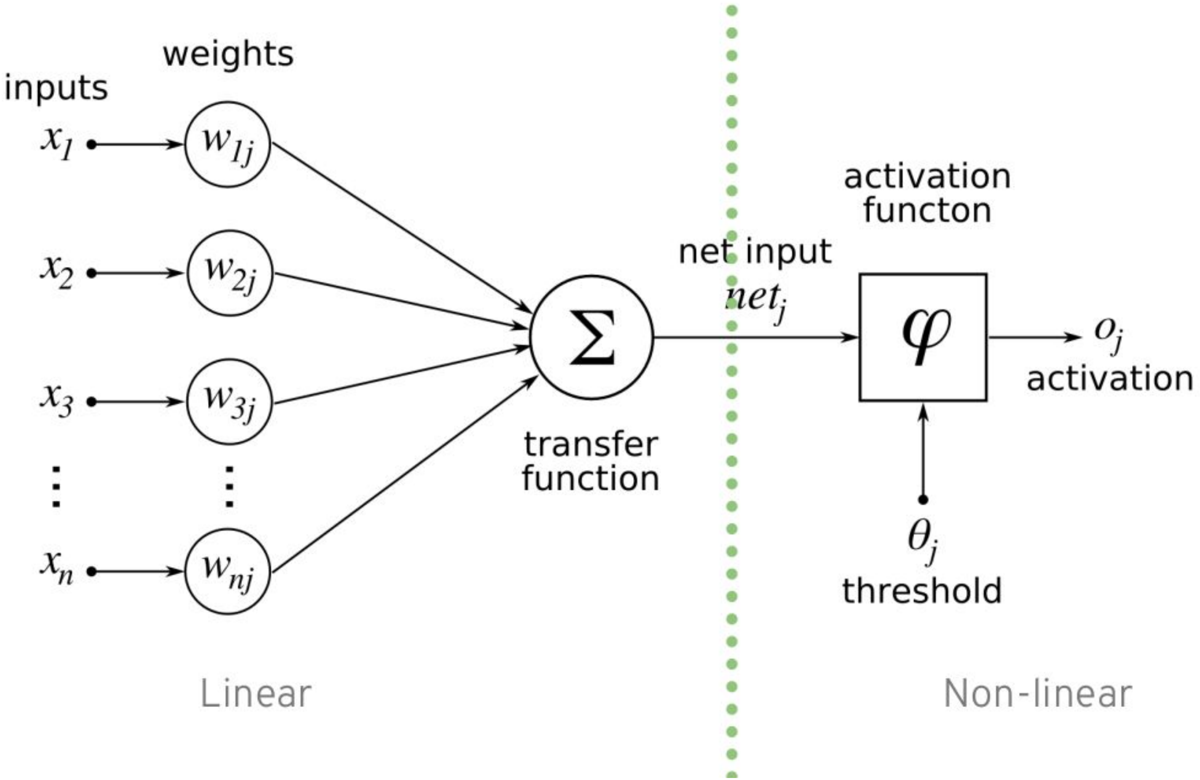


Perceptron



Perceptron

$\sigma(wx + b)$



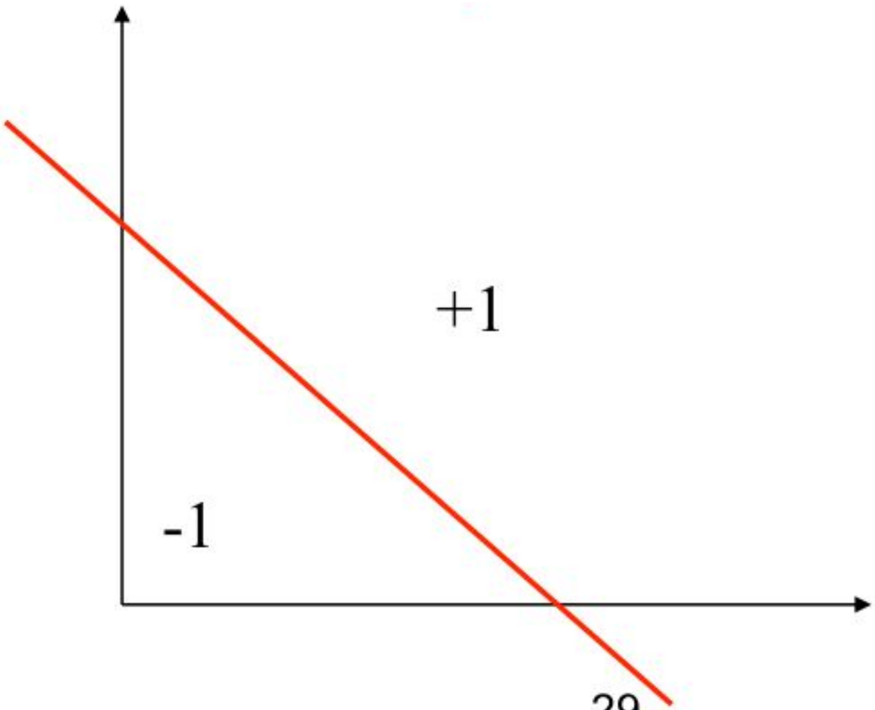
Activation Functions

- Important feature of NN’s!
- Decide whether a neuron should be activated or not

  => whether the information that the neuron is receiving is relevant for the given information or if it should be ignored
- Output is sent to the next layer of neurons as input
- Non linear transformation

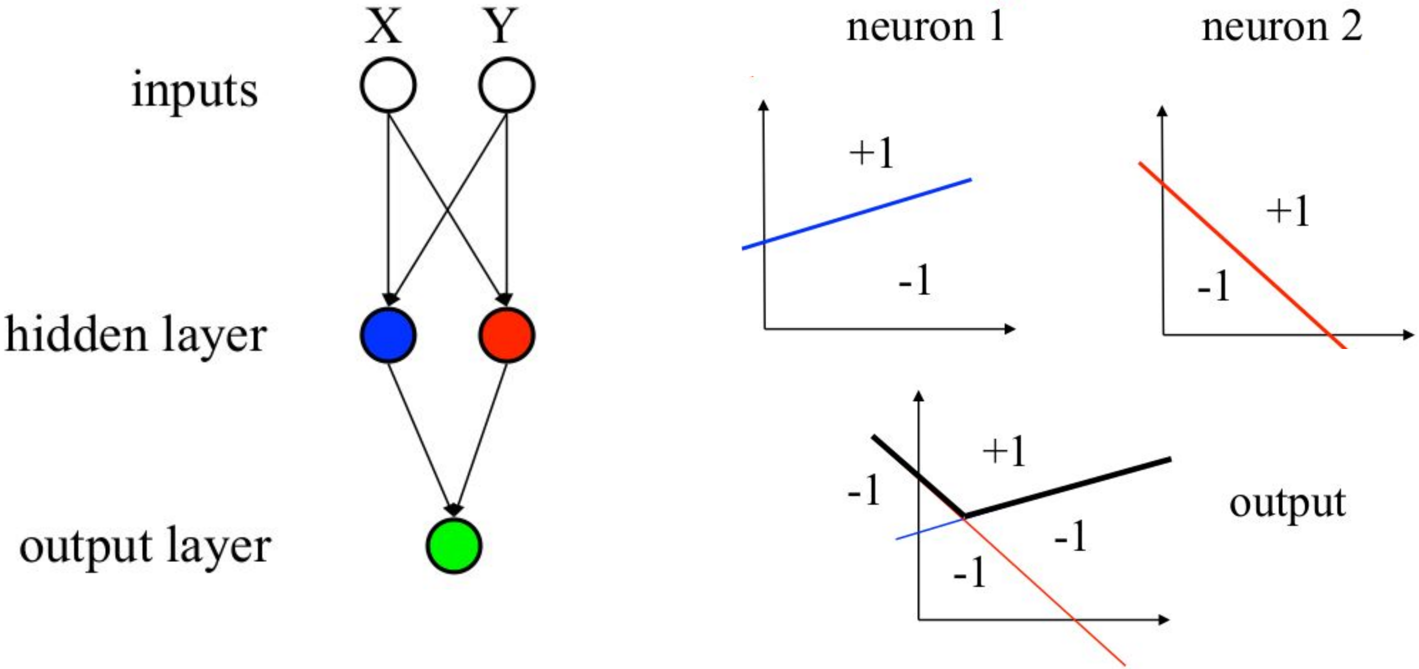
  => Linear equation is too limited to solve complex problems
- Make the back-propagation algorithm possible

Perceptron



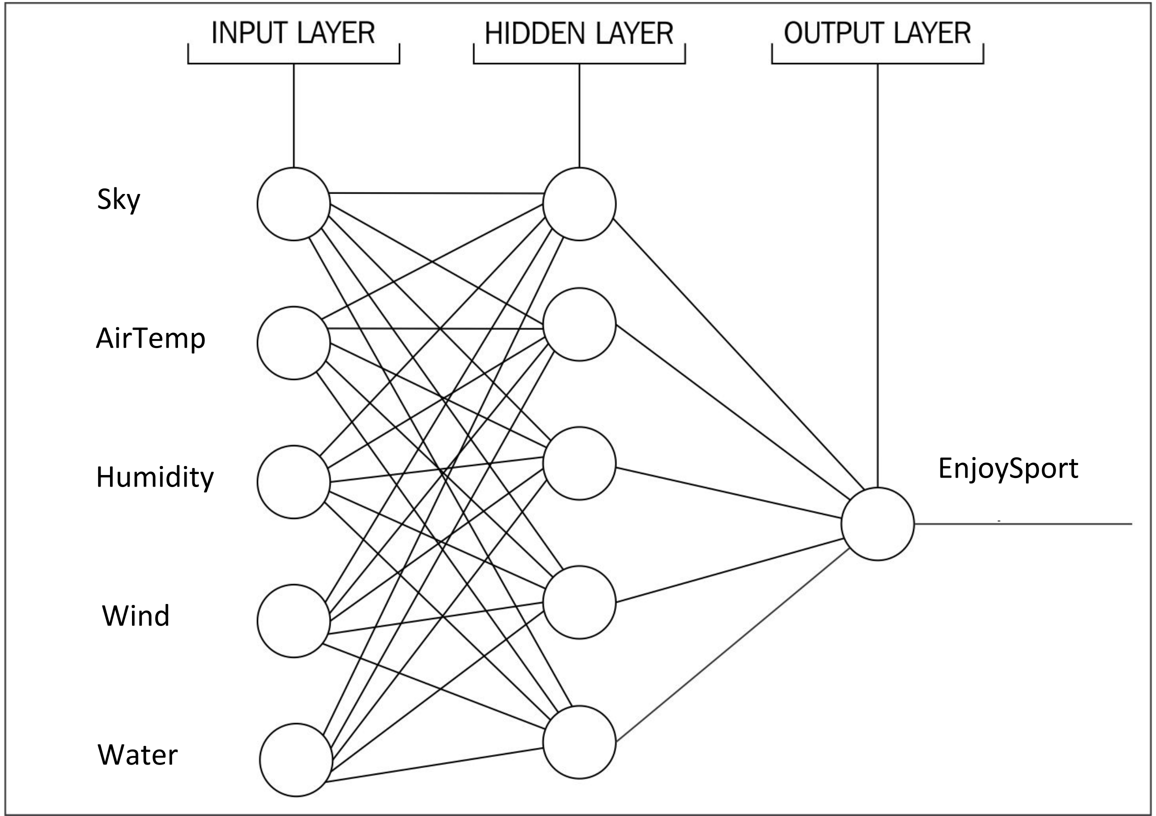
- Creates linear separation between classes +1 and -1
  => Classes must be linearly separable
- But what about nonlinear data? :-(

Multi-layer networks

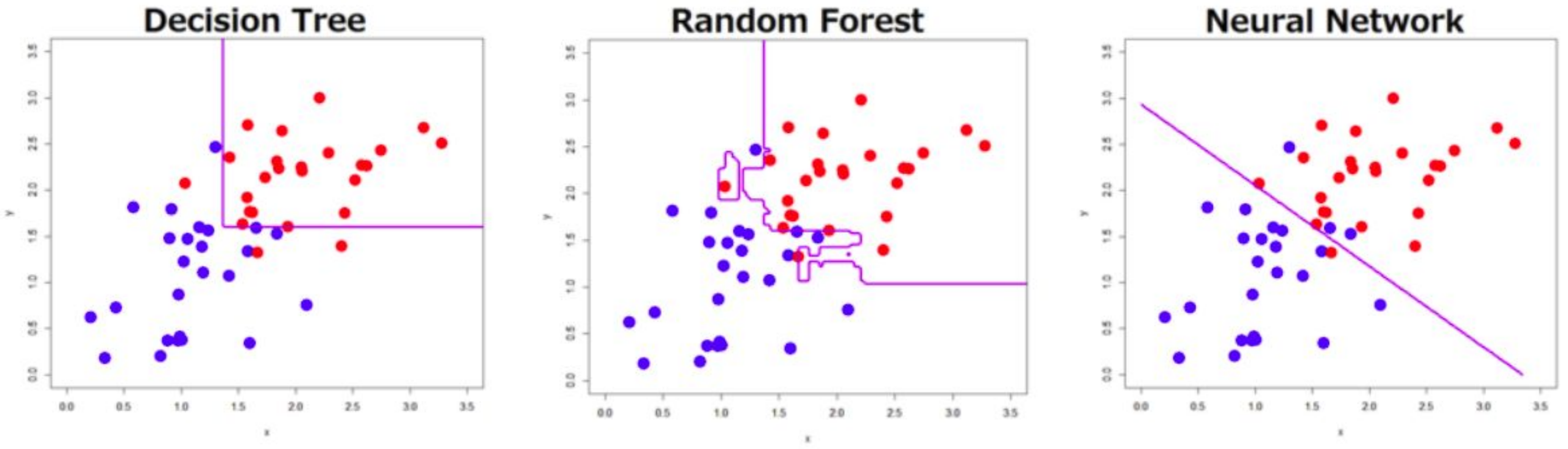


A multi-layer network with just 1 hidden layer can model any (continuous) function!
=> But the layer may be super large and may fail to generalize correctly

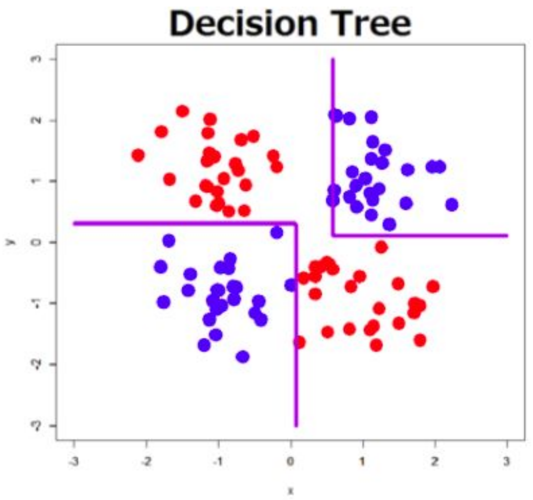
Neural Networks

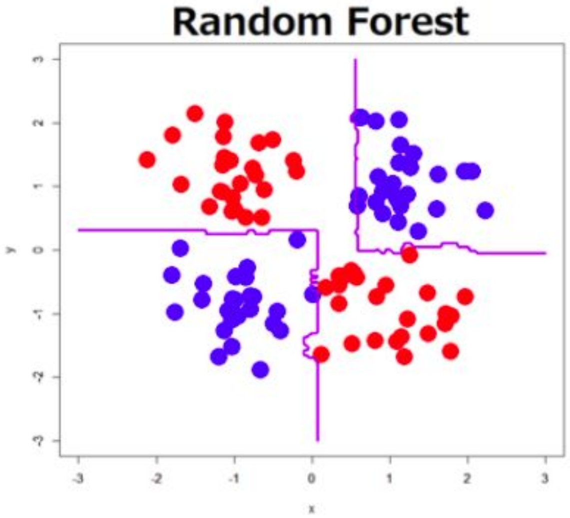


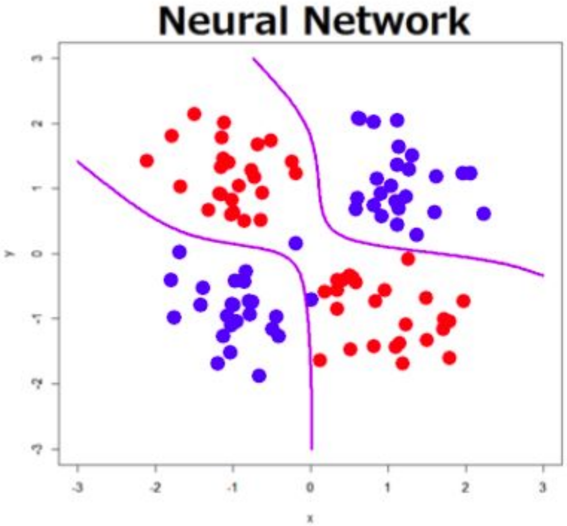
Decision boundaries: linear separation



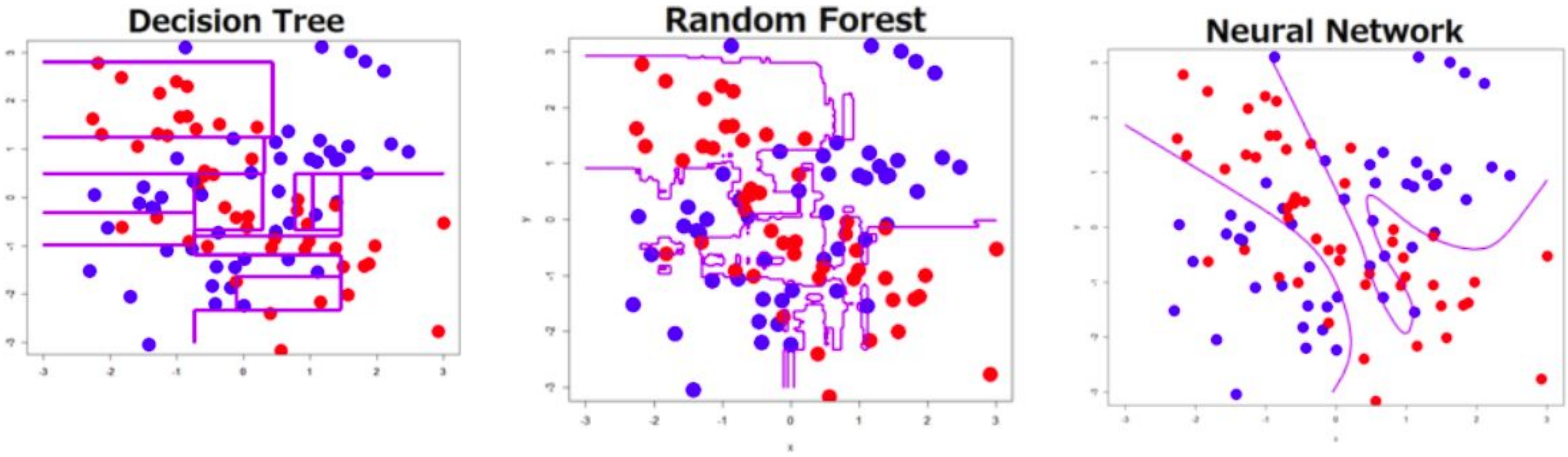
Decision boundaries: simple nonlinear separation



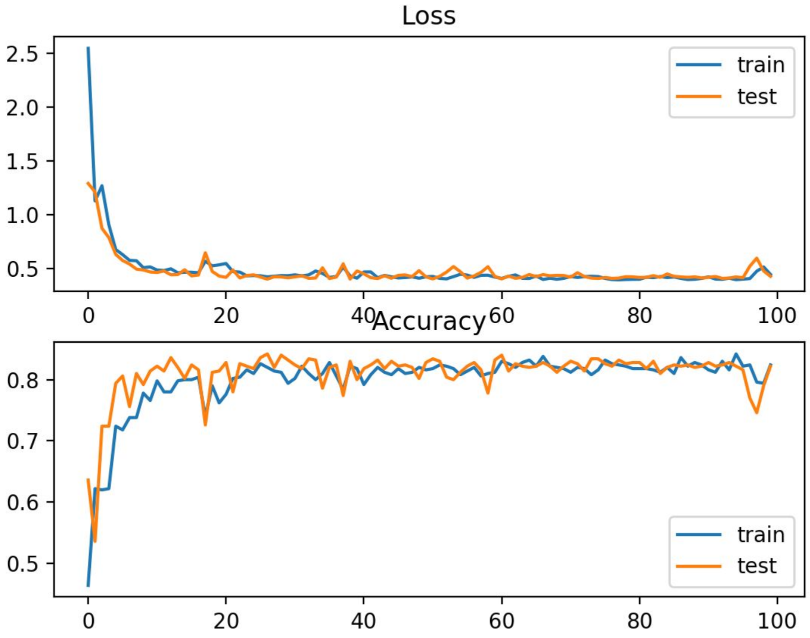




Decision boundaries: complex nonlinear separation



Loss functions



- Optimization objective
  - Minimize Errors
  - == Minimize Loss
- Regression: MSE / MAE

$$\frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2$$

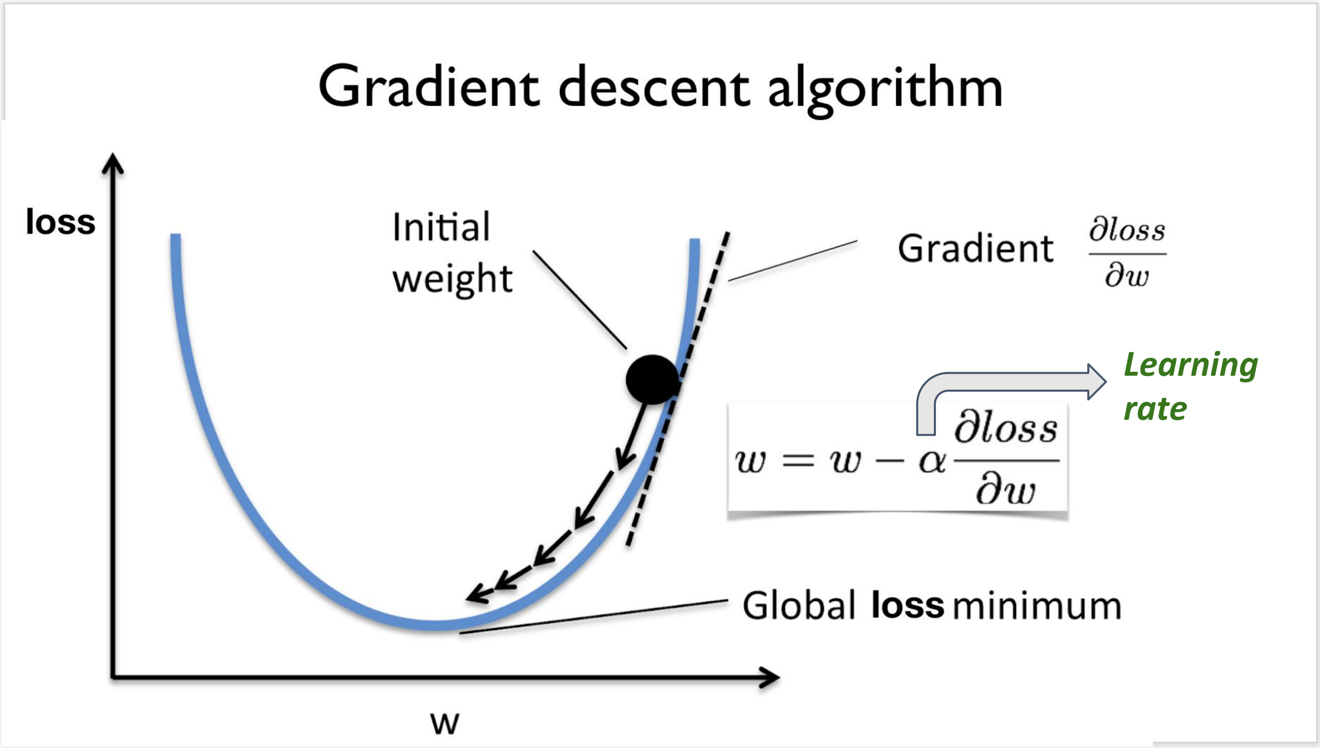
- Classification: Log Loss

$$-\frac{1}{N}\sum_{i=1}^{N}(y_i \log(p_i) + (1 - y_i)\log(1 - p_i))$$

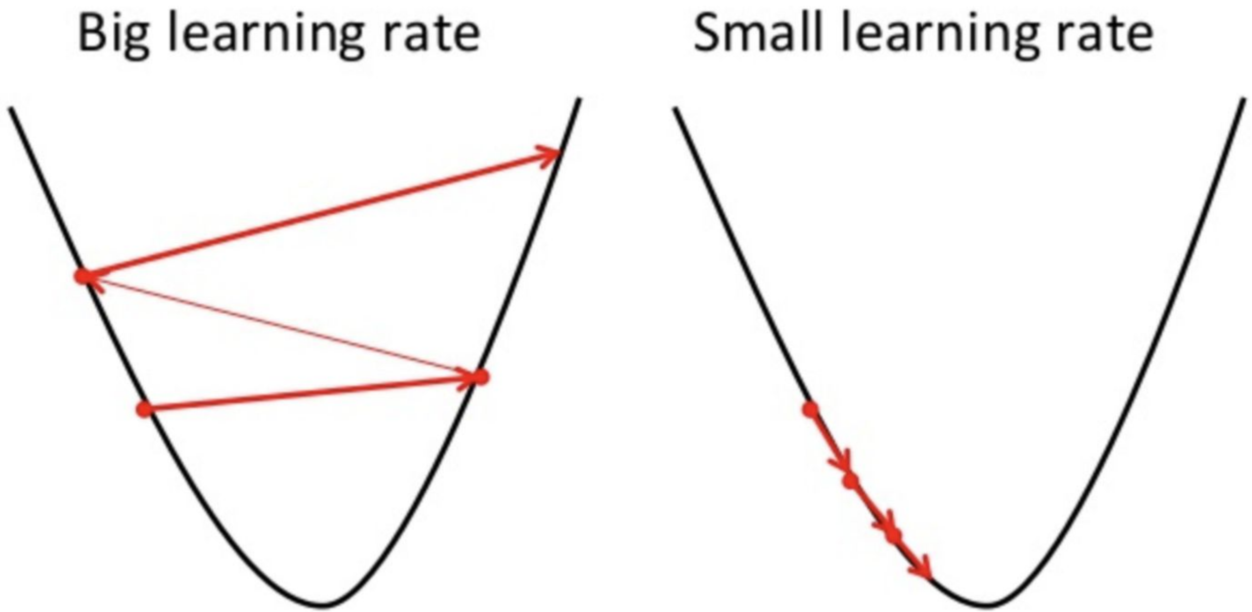
- Multi classification: M Log Loss

$$-\frac{1}{N}\sum_{i=1}^{N}\sum_{j=1}^{M} y_{i,j} \log(p_{i,j})$$

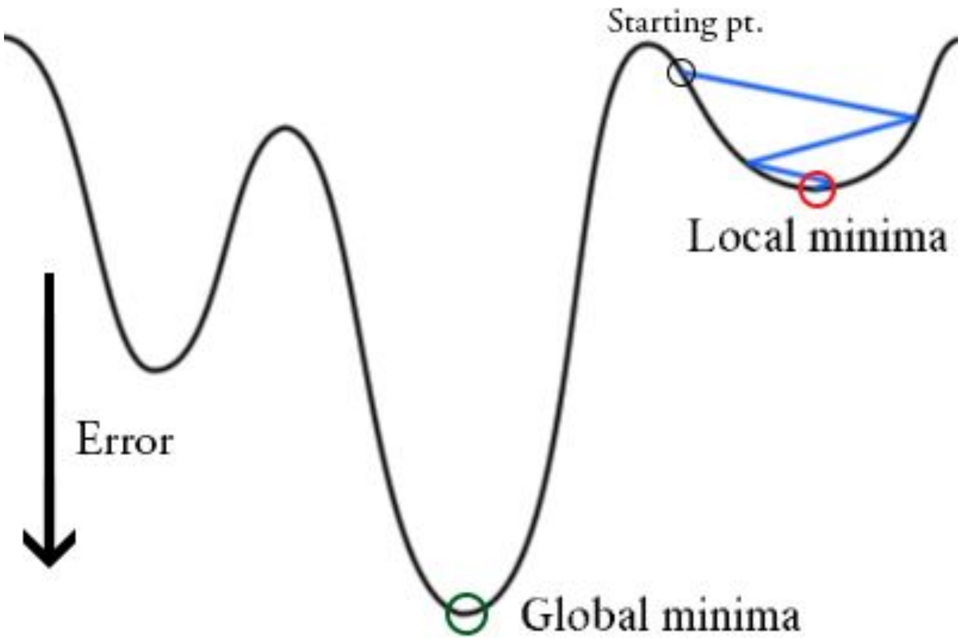
Gradient Descent

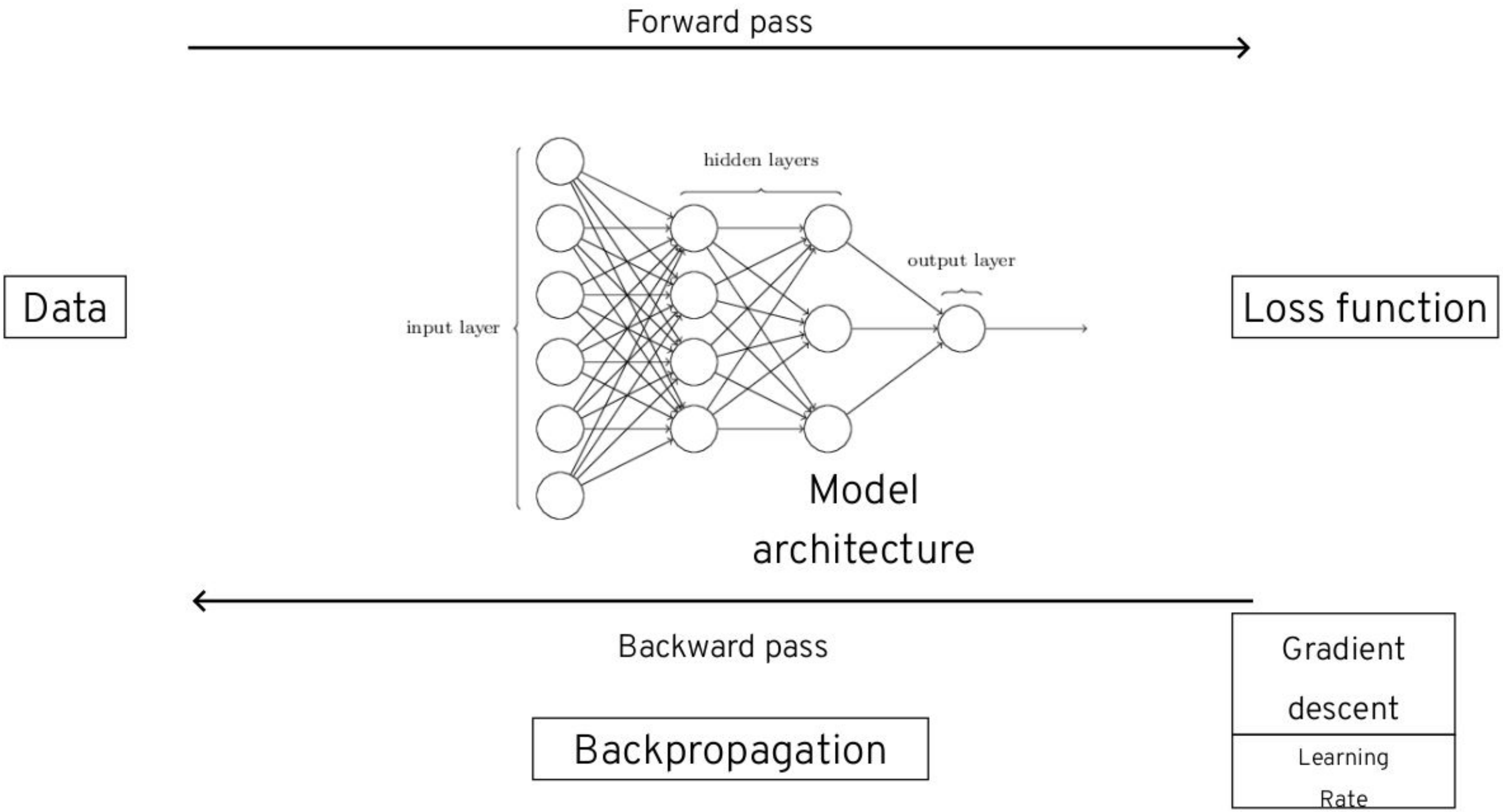


Gradient Descent



Gradient Descent




Forward pass
Data
hidden layers
output layer
input layer
Loss function
Model
architecture
Backward pass
Gradient
descent
Learning
Rate
Backpropagation

A mostly complete chart of

Neural Networks