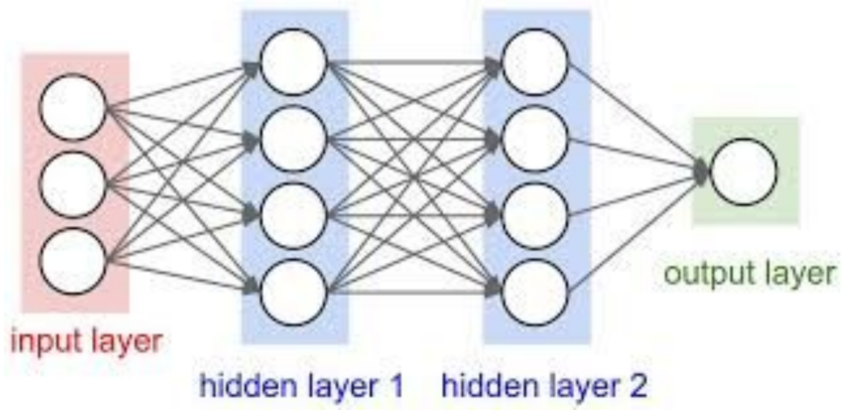
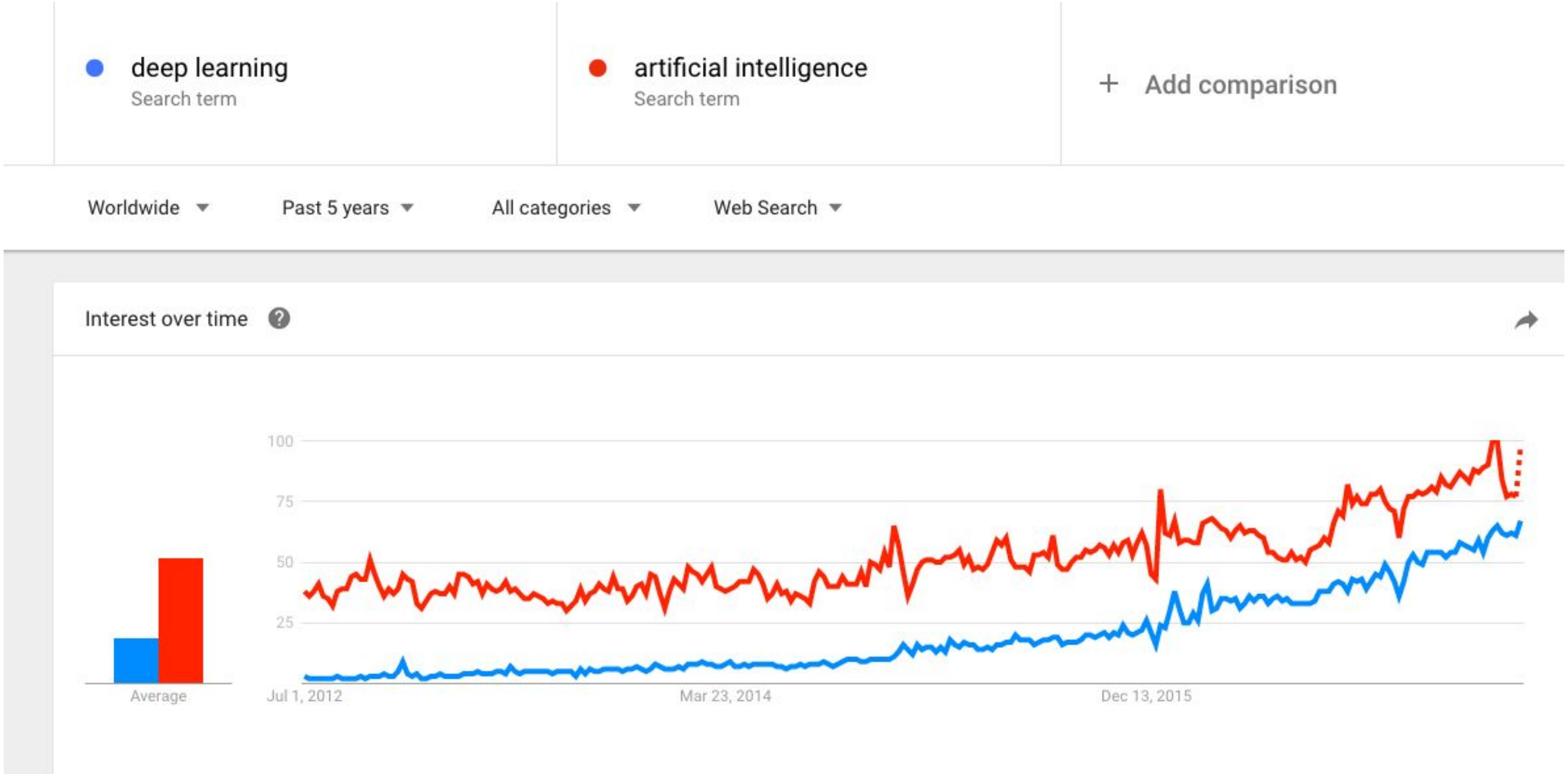
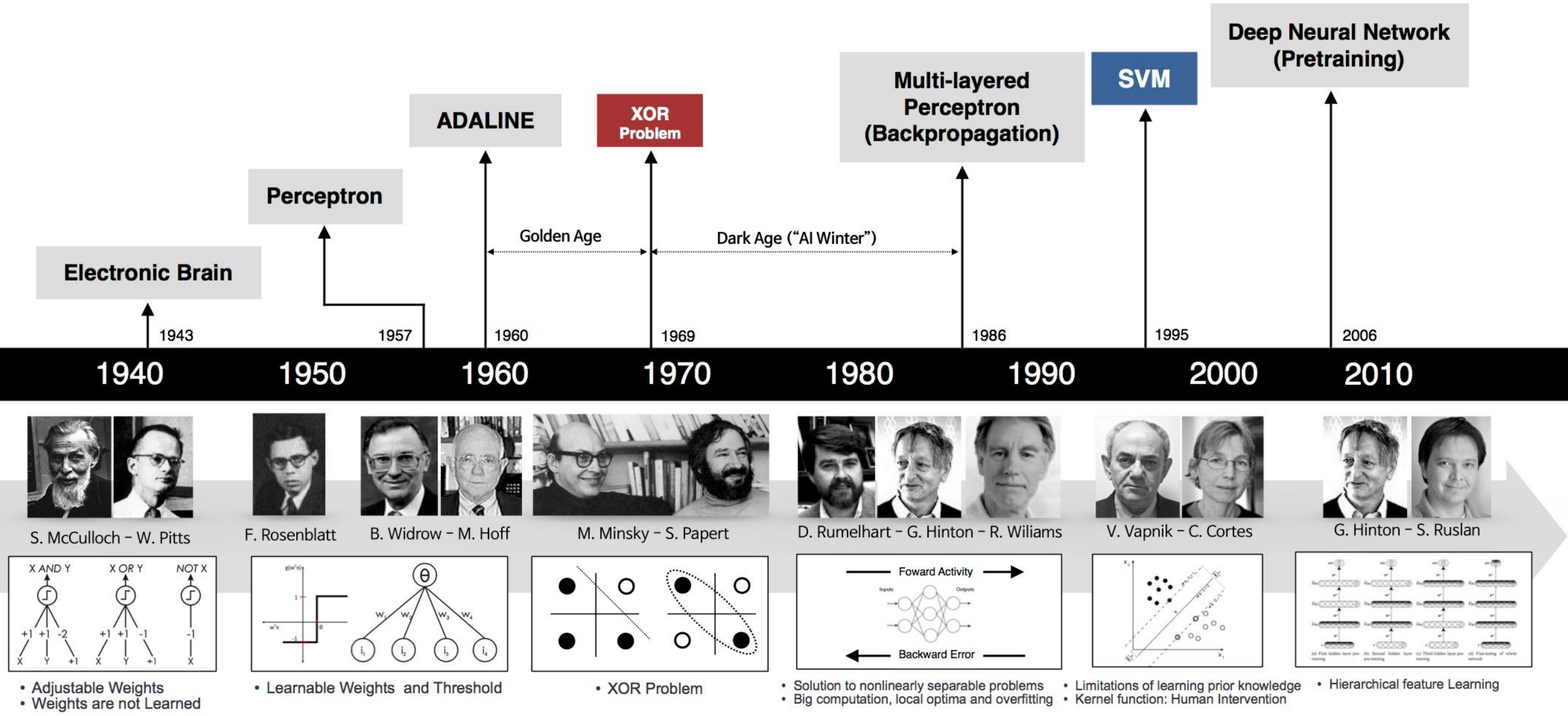
Introduction to Neural Networks

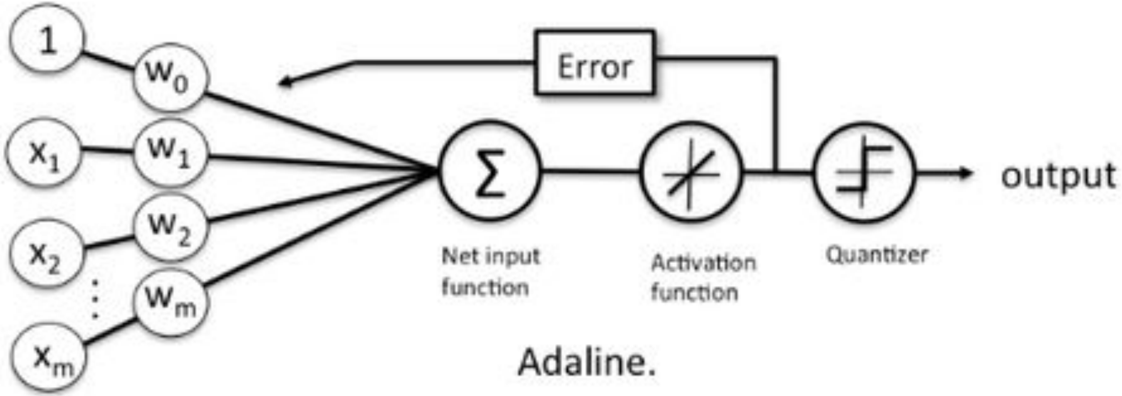


History of Deep Learning

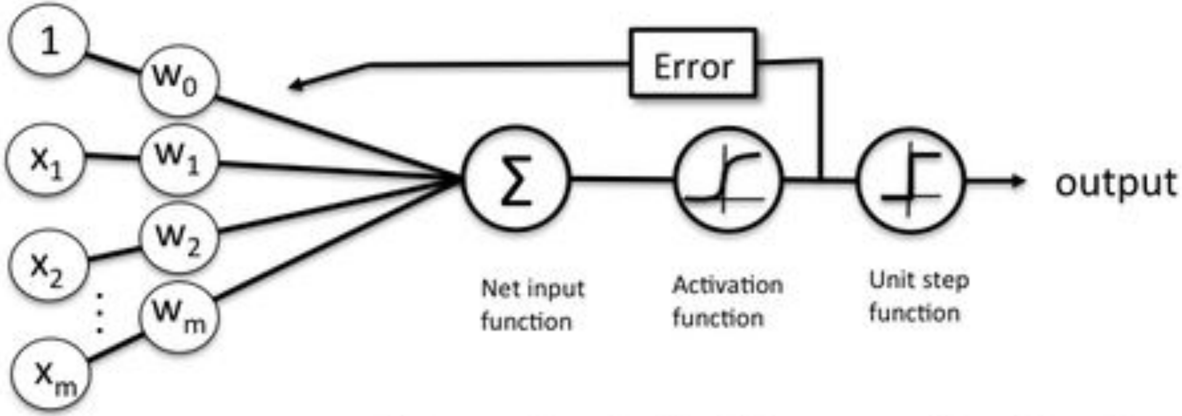




Deep Neural Network (Pretraining)
SVM
Multi-layered Perceptron (Backpropagation)
XOR Problem
ADALINE
Perceptron
Golden Age
Dark Age ("AI Winter")
Electronic Brain
1943
1957
1960
1969
1986
1995
2006
1940
1950
1960
1970
1980
1990
2000
2010
S. McCulloch - W. Pitts
F. Rosenblatt
B. Widrow - M. Hoff
M. Minsky - S. Papert
D. Rumelhart - G. Hinton - R. Wiliams
V. Vapnik - C. Cortes
G. Hinton - S. Ruslan
X AND Y
X OR Y
NOT X
Foward Activity
Backward Error
• Adjustable Weights
• Weights are not Learned
• Learnable Weights and Threshold
• XOR Problem
• Solution to nonlinearly separable problems
• Big computation, local optima and overfitting
• Limitations of learning prior knowledge
• Kernel function: Human Intervention
• Hierarchical feature Learning

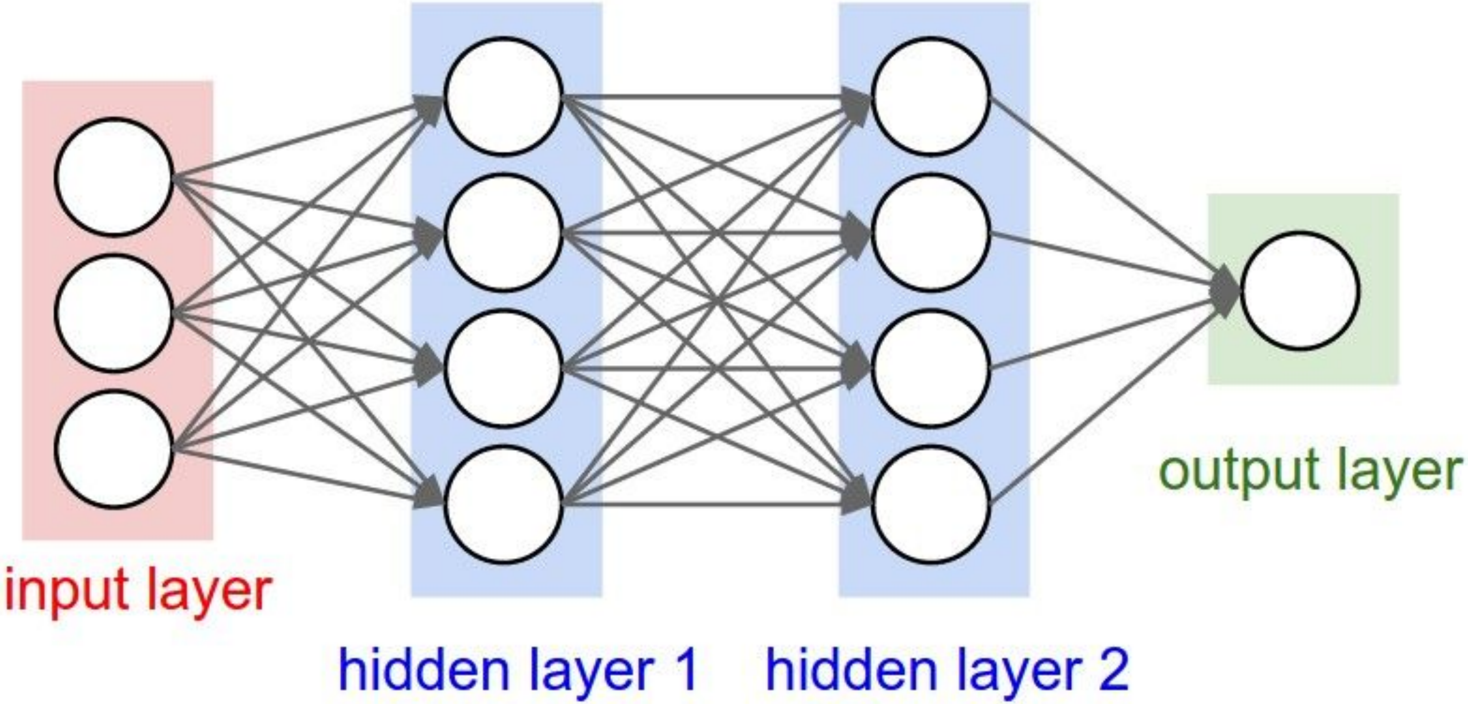


Adaline.

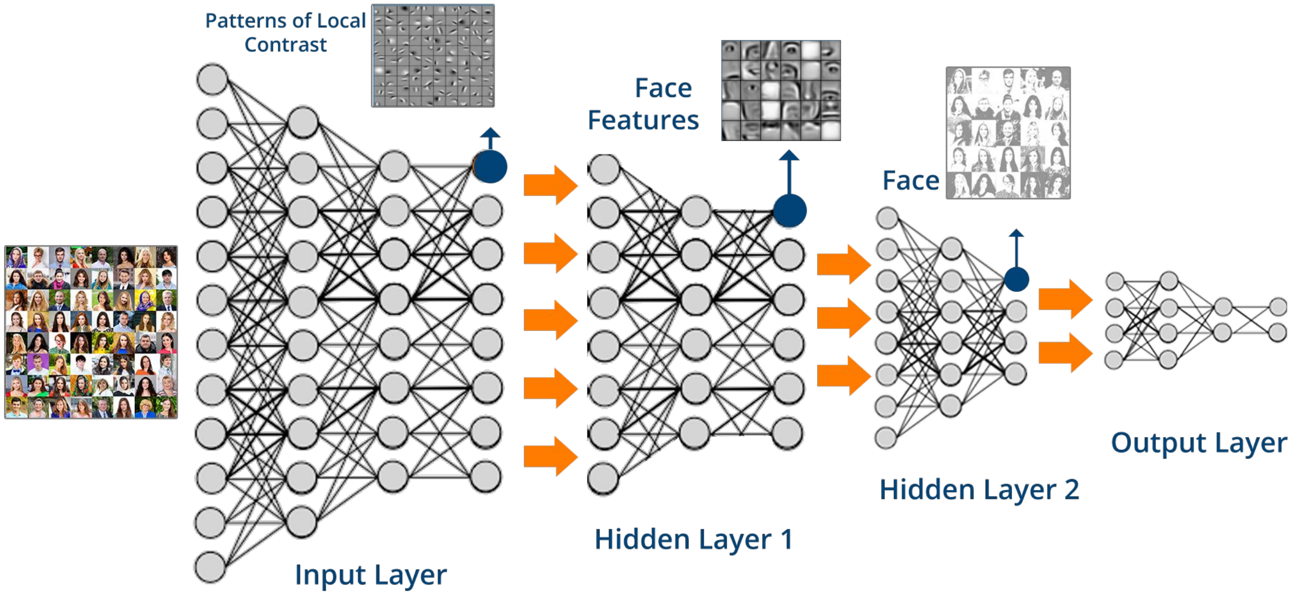


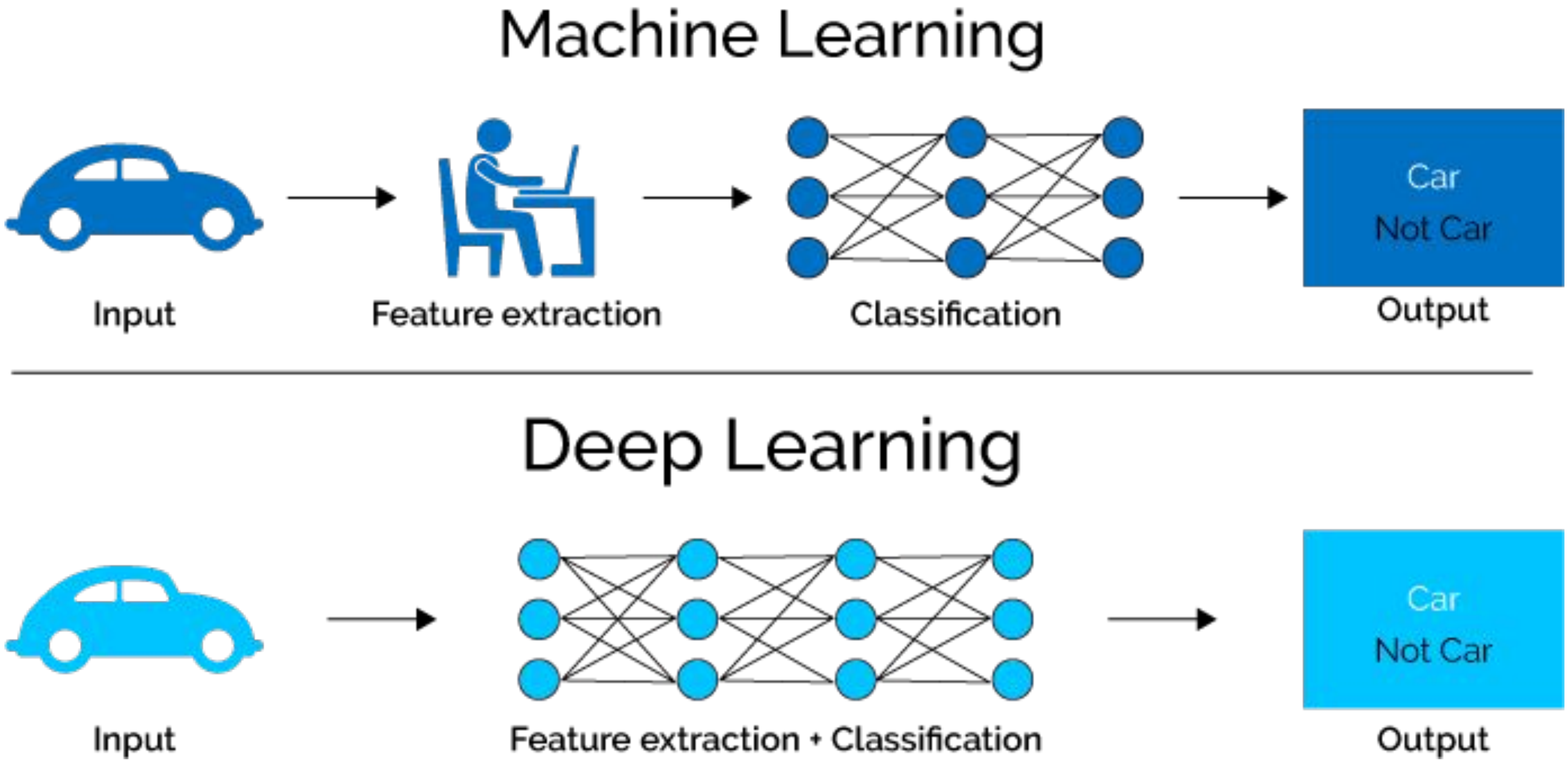
Schematic of a logistic regression classifier.

Compared to the ML algorithms we have studied so far, **Neural Networks** are more complex function approximators with nonlinearities in between. Neural Networks are composition of functions learned using *gradient descent*.



Deep Learning refers to Neural Networks with high degree of compositions. The *depth* in Deep Learning refers to this very fact. Current advances in AI are driven by progress in Deep Learning.

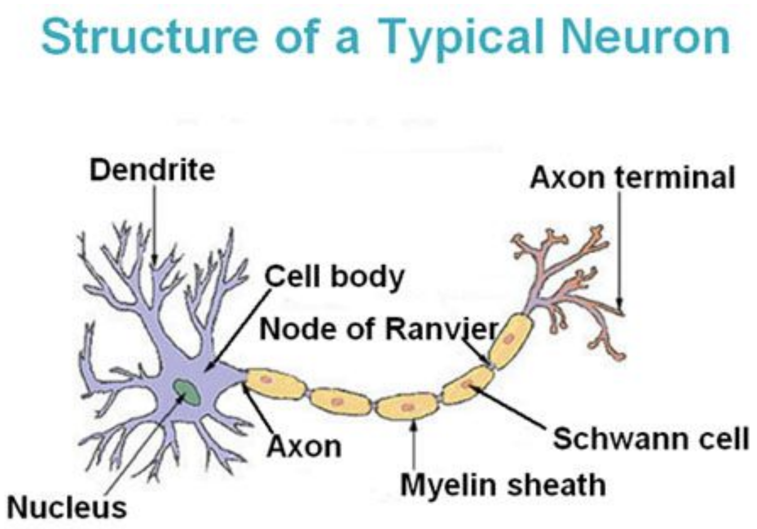


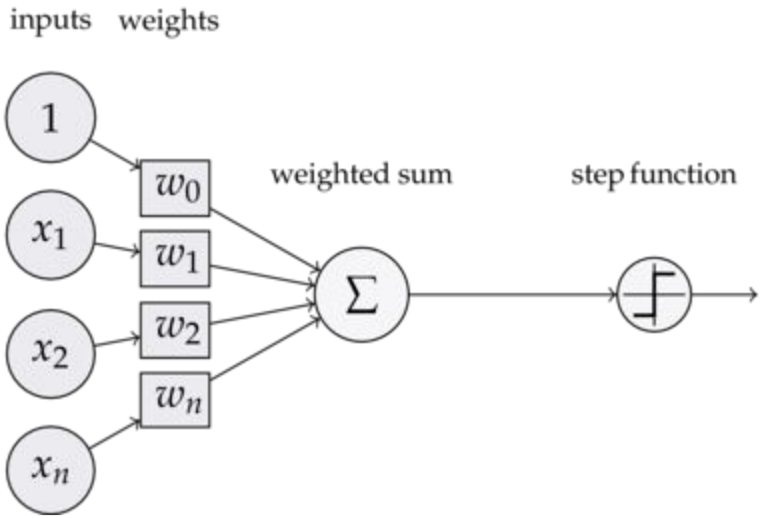

Machine Learning
Car
Not Car
Input
Feature extraction
Classification
Output
Deep Learning
Car
Not Car
Input
Feature extraction + Classification
Output

Perceptron

Perceptrons are the simplest example of a neural network

The idea is to emulate a single neuron



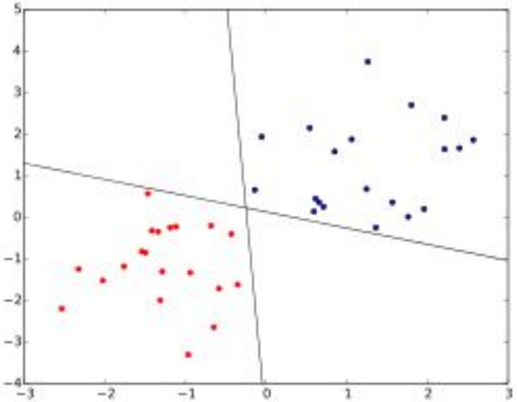


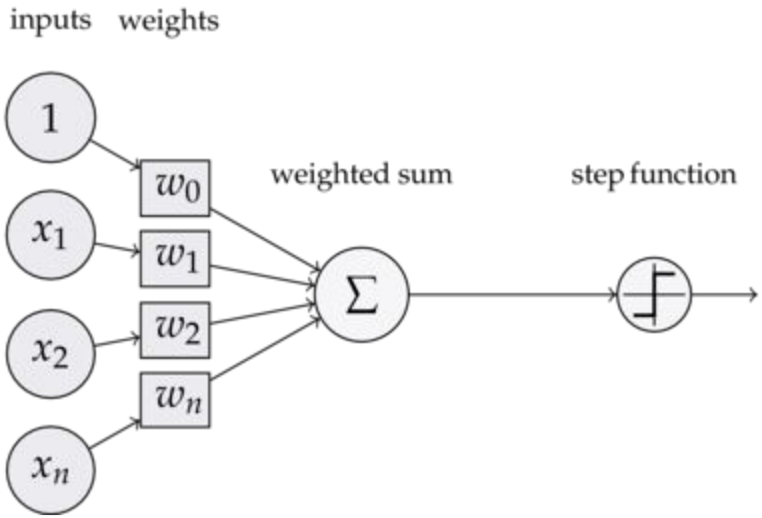
Perceptron

Given inputs and an activation or link function

Common activations are softmax, tanh, linear, logistic, tanh

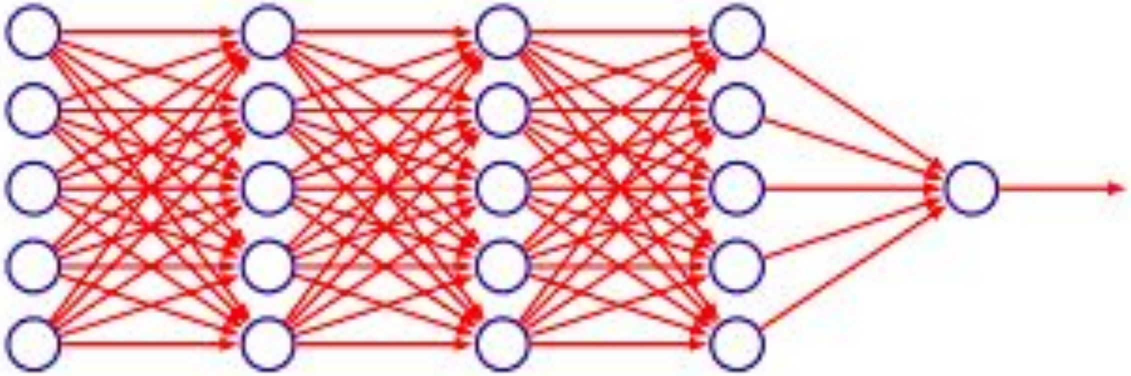
Computes a linear separation



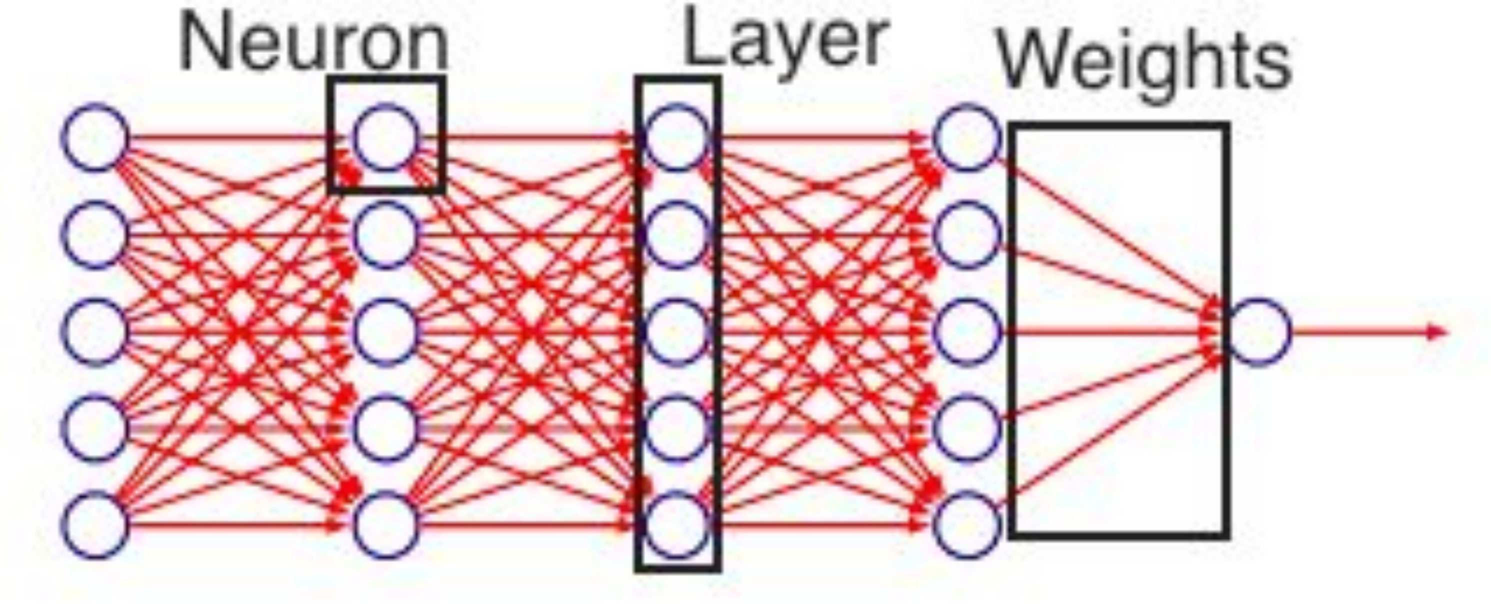


Feedforward Neural Networks

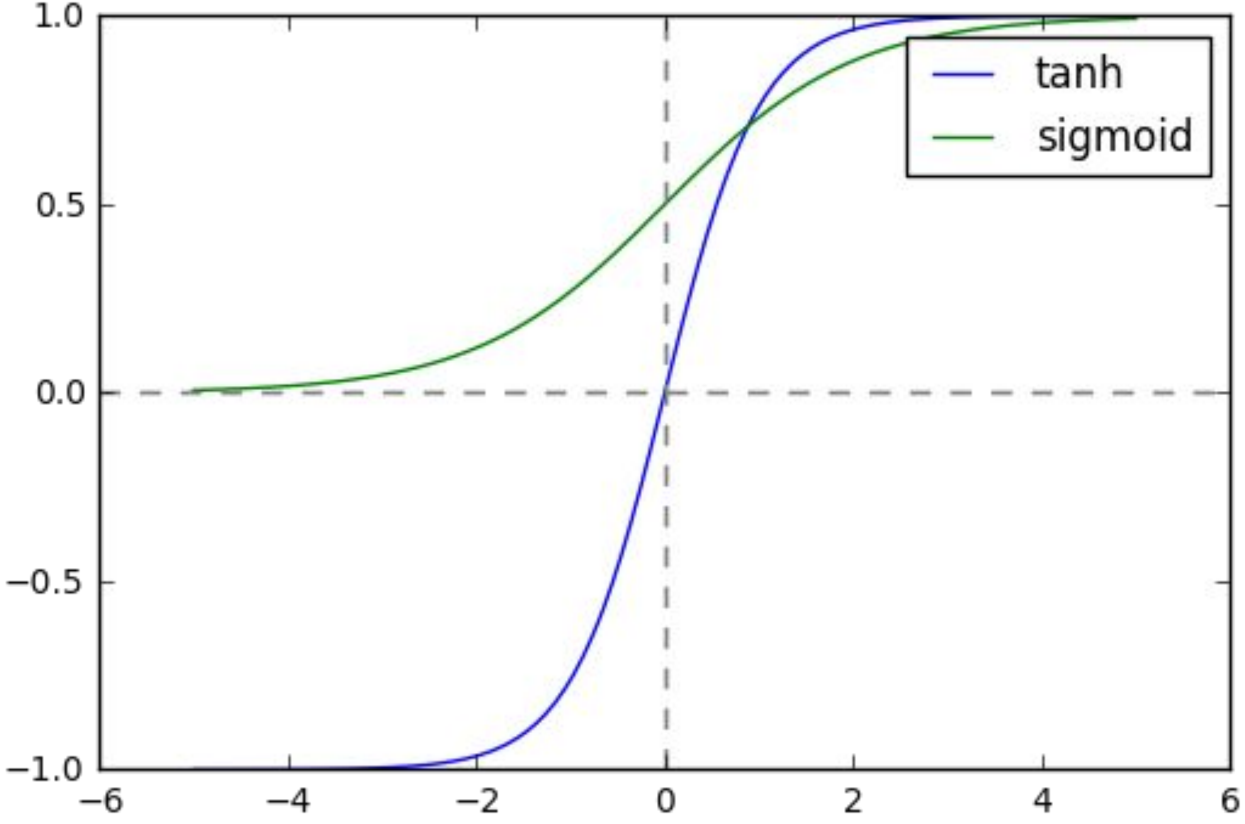
Perceptrons combined into multilayer perceptrons from feedforward network



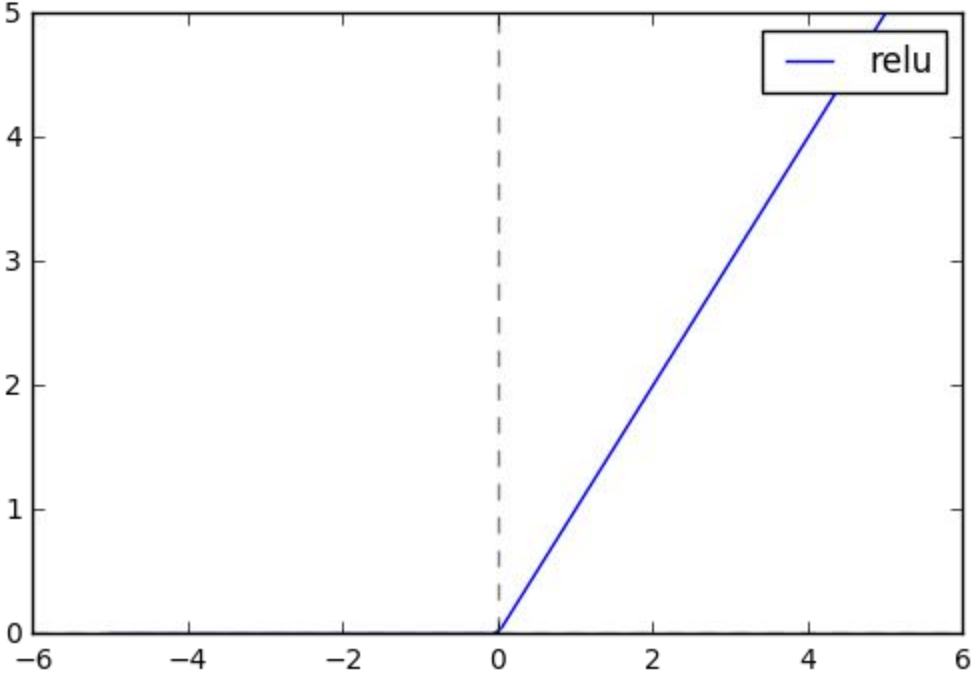
Feedforward Neural Networks

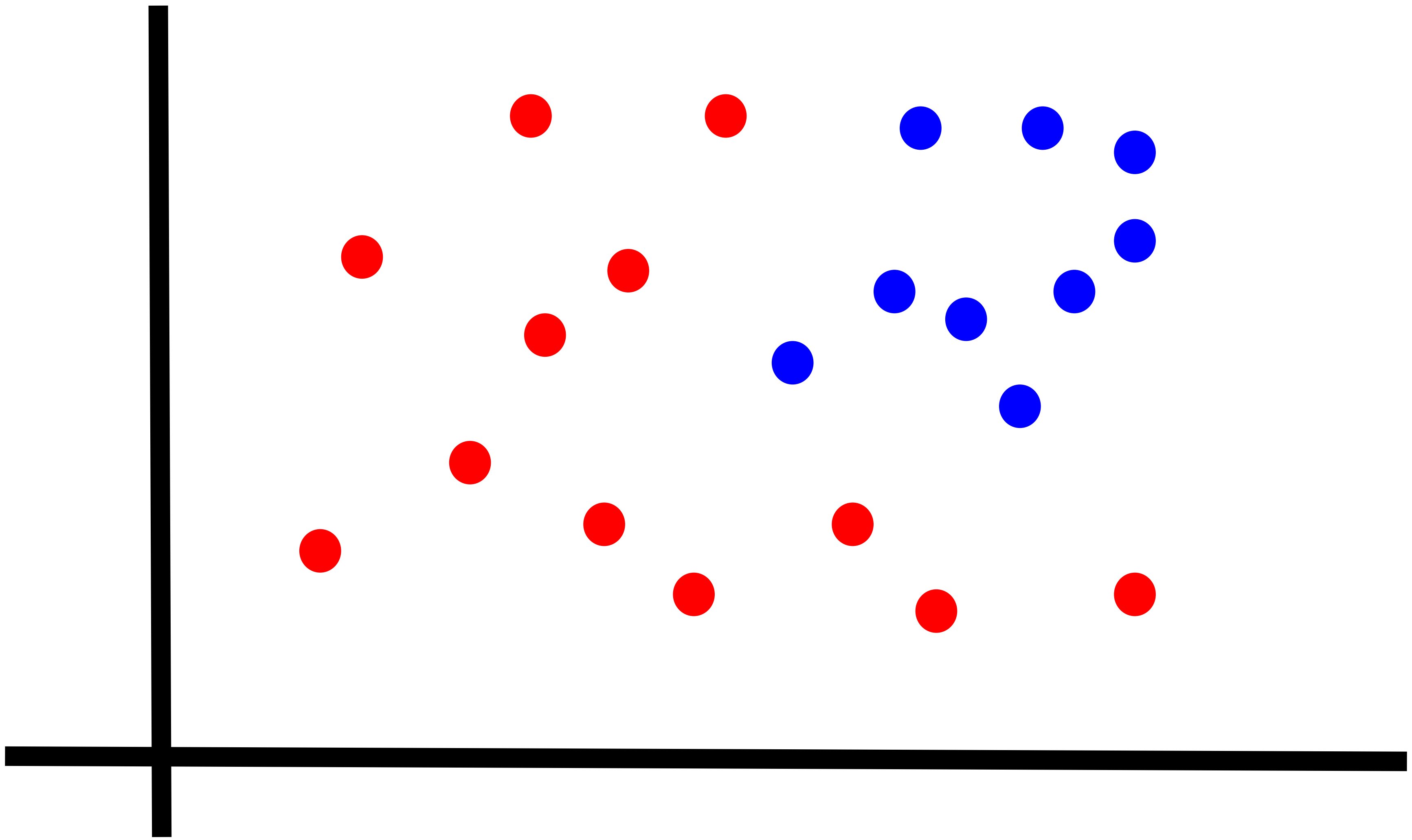


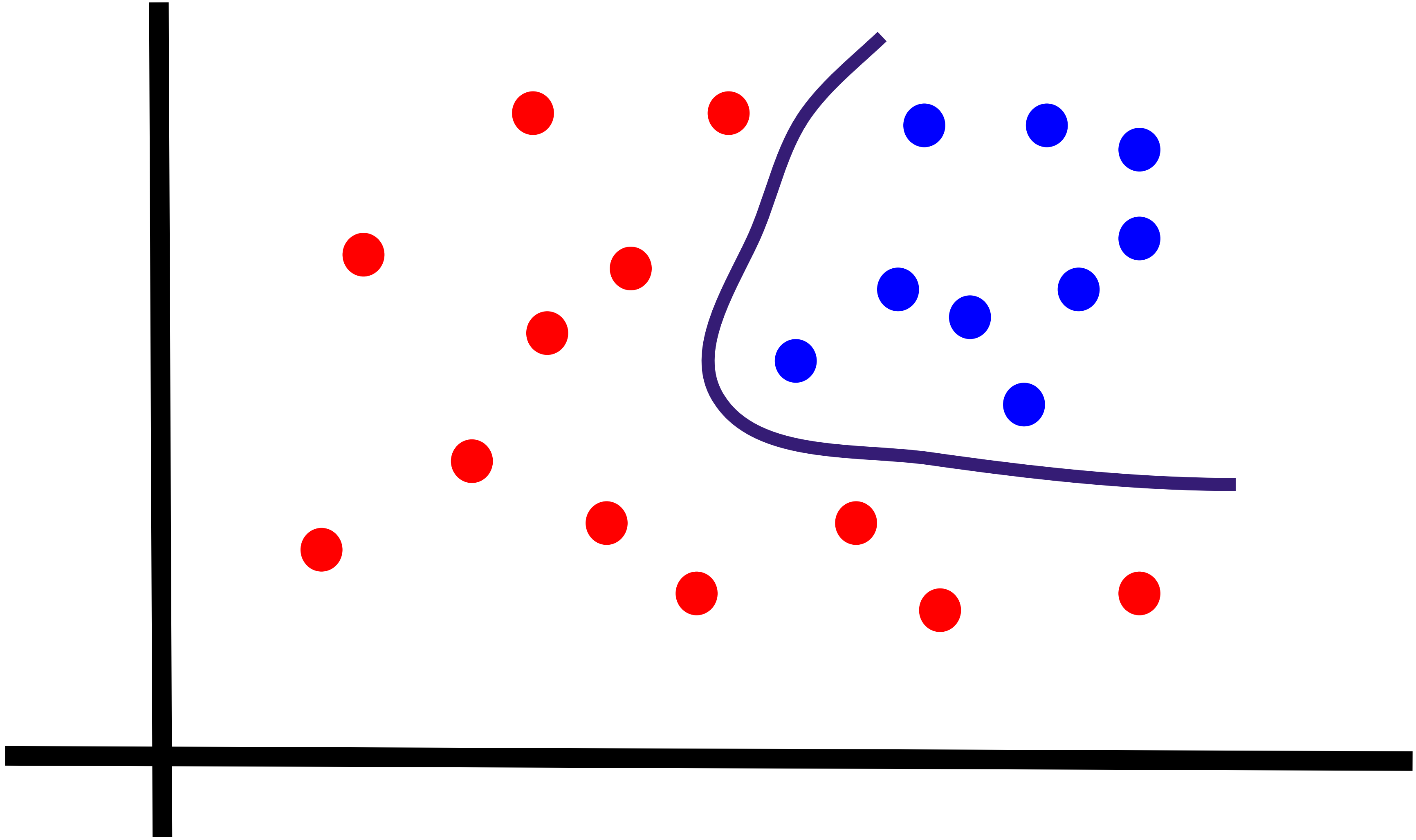
Tanh and Sigmoid Activation

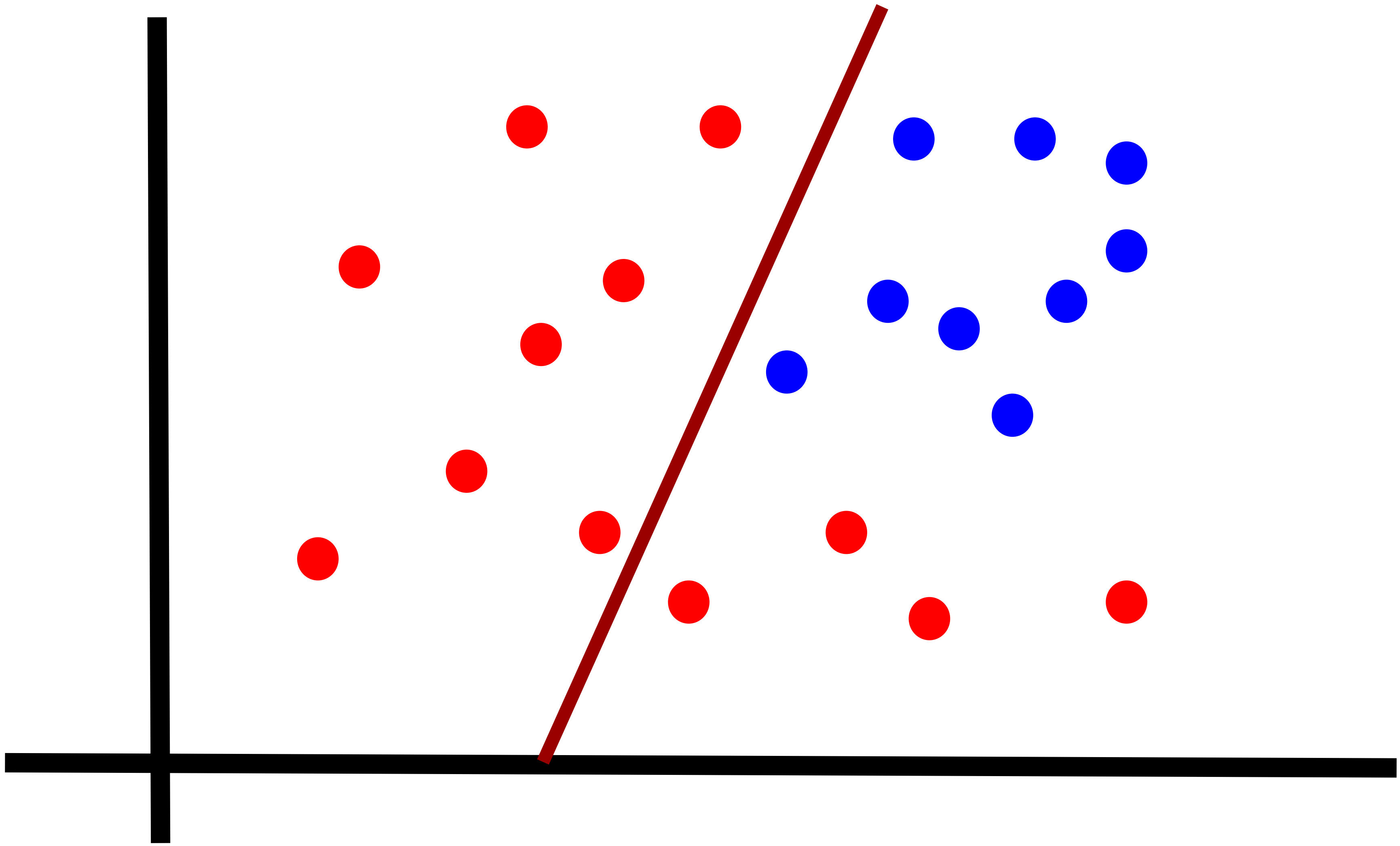


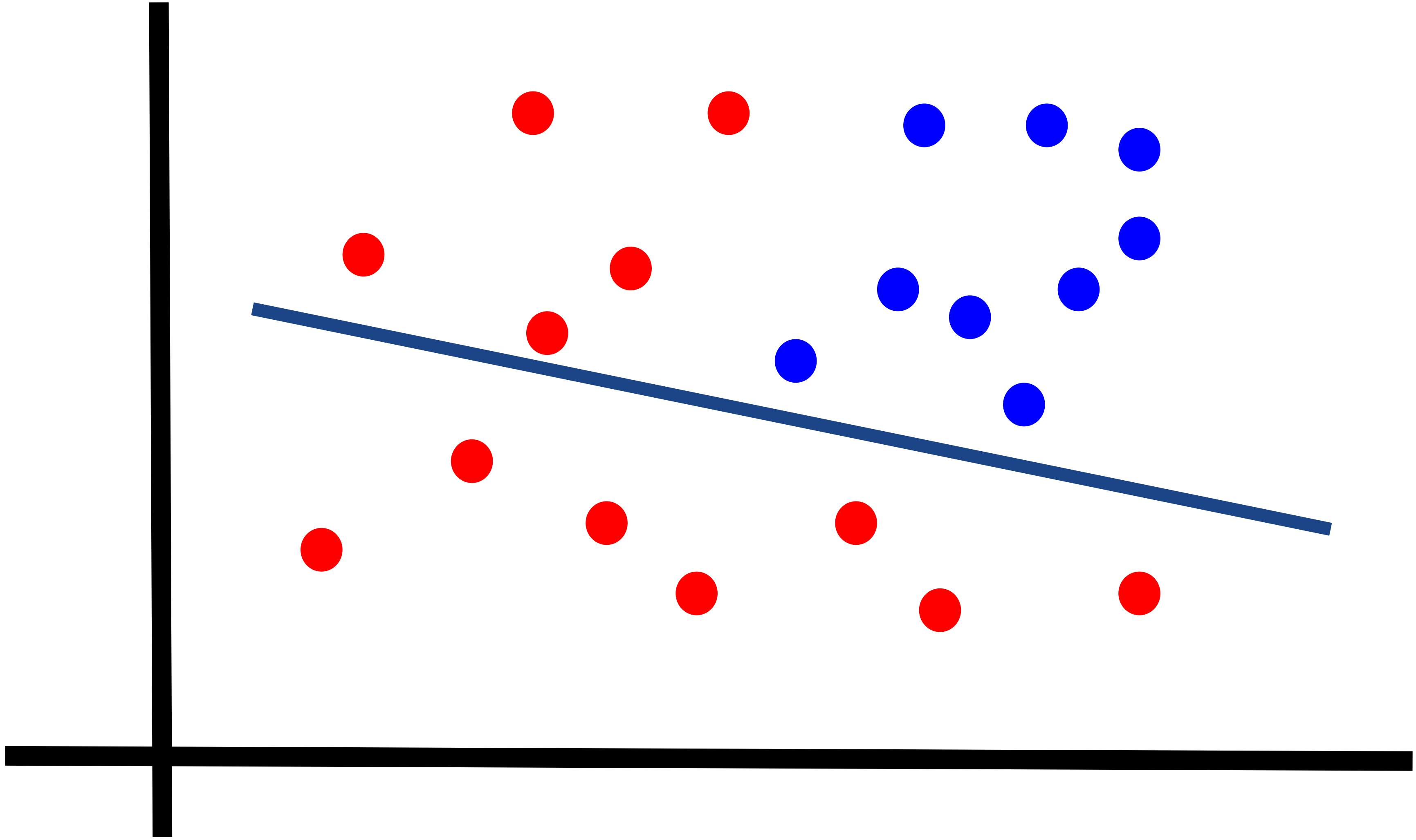
ReLu Activation

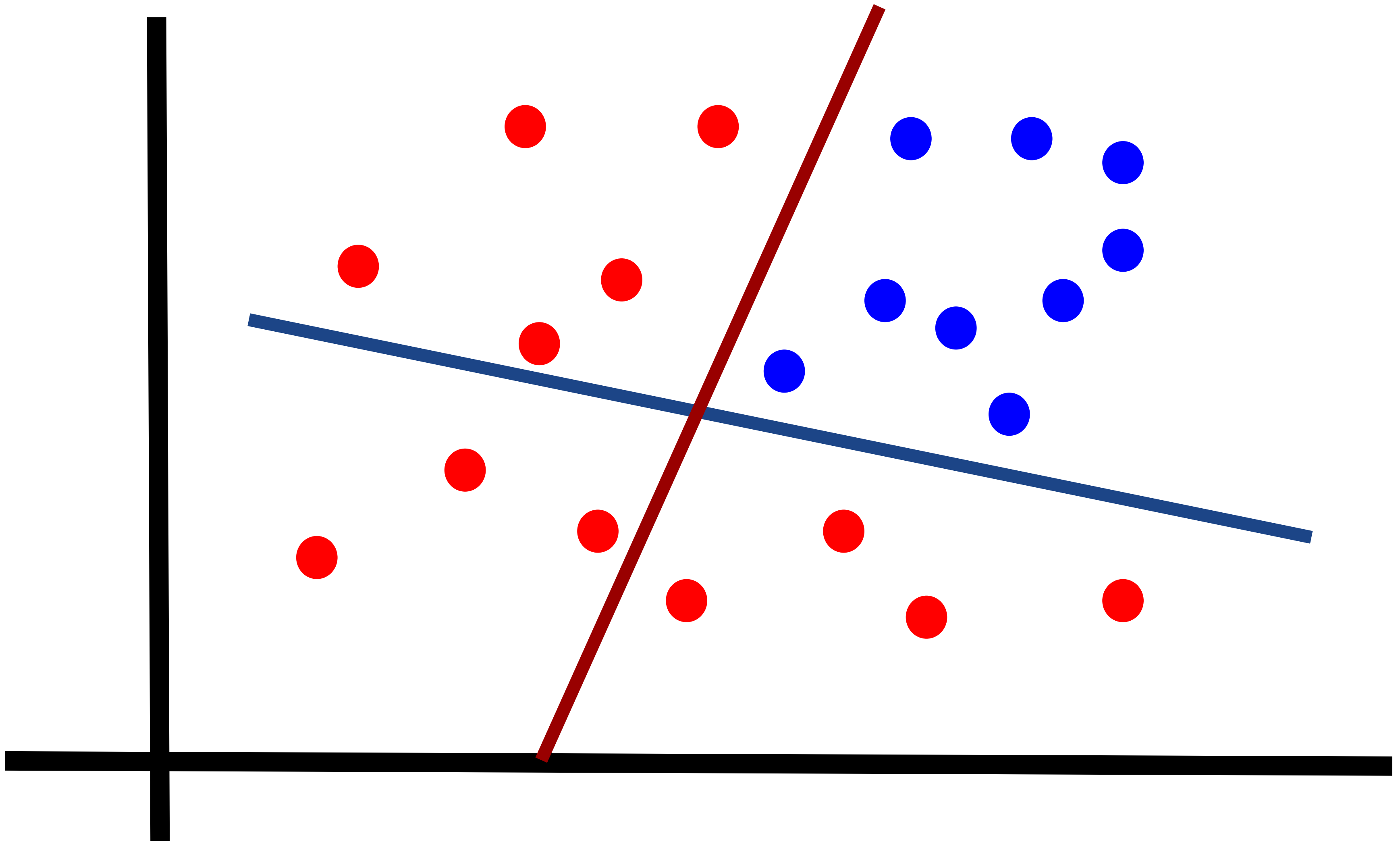




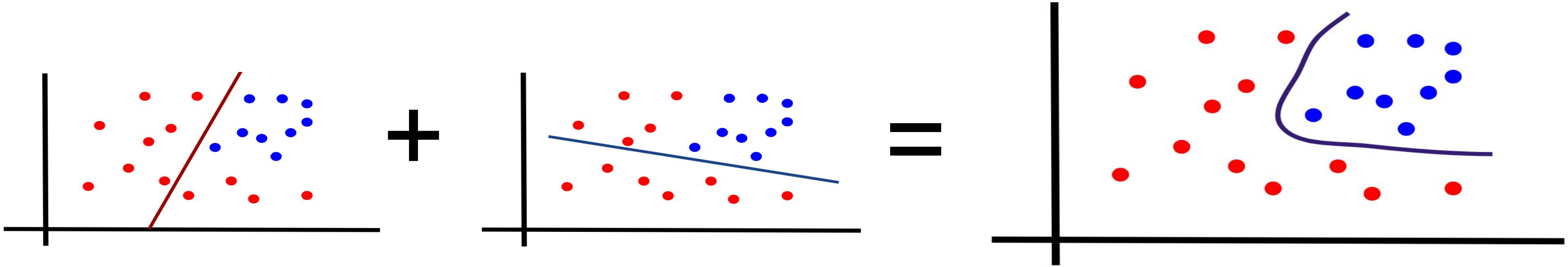






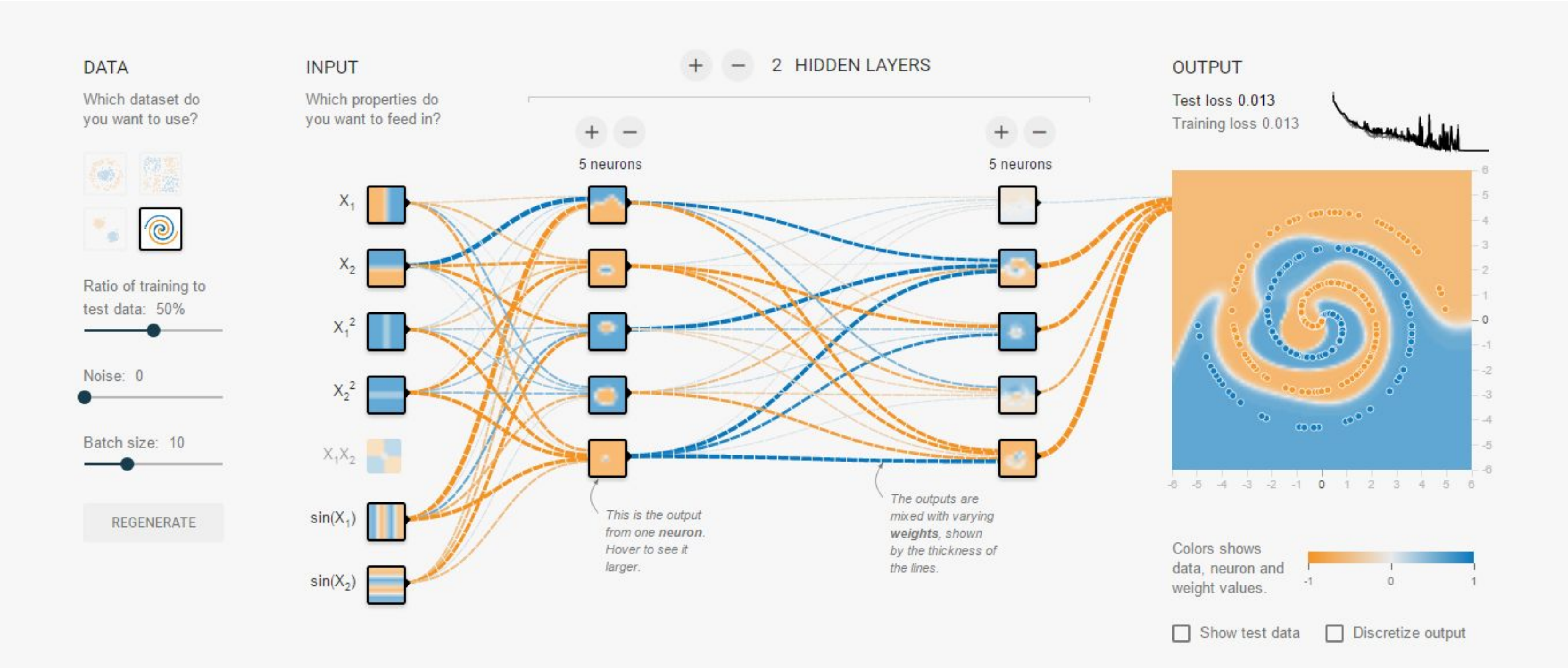


Combining Regions

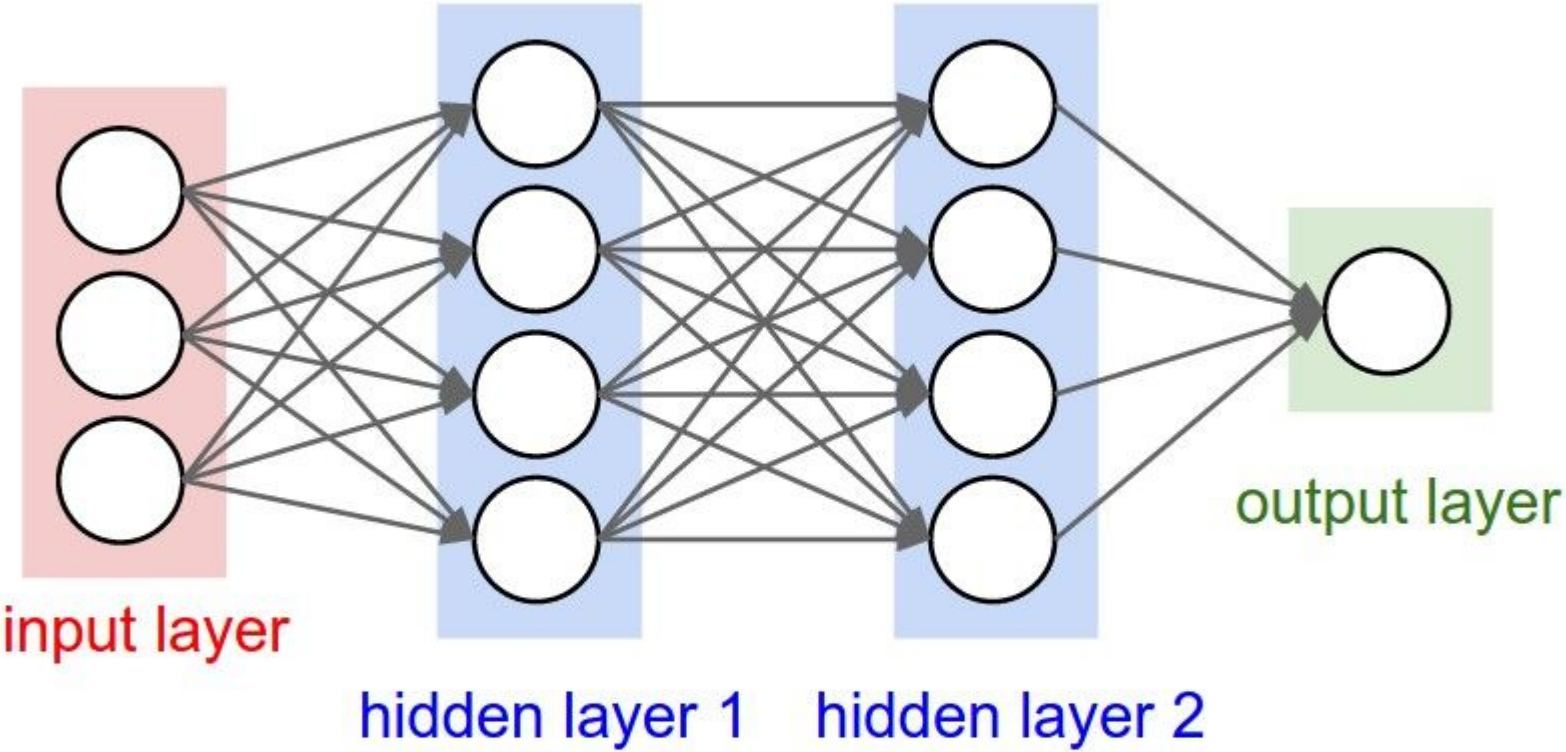


Tensorflow Playground

http://playground.tensorflow.org

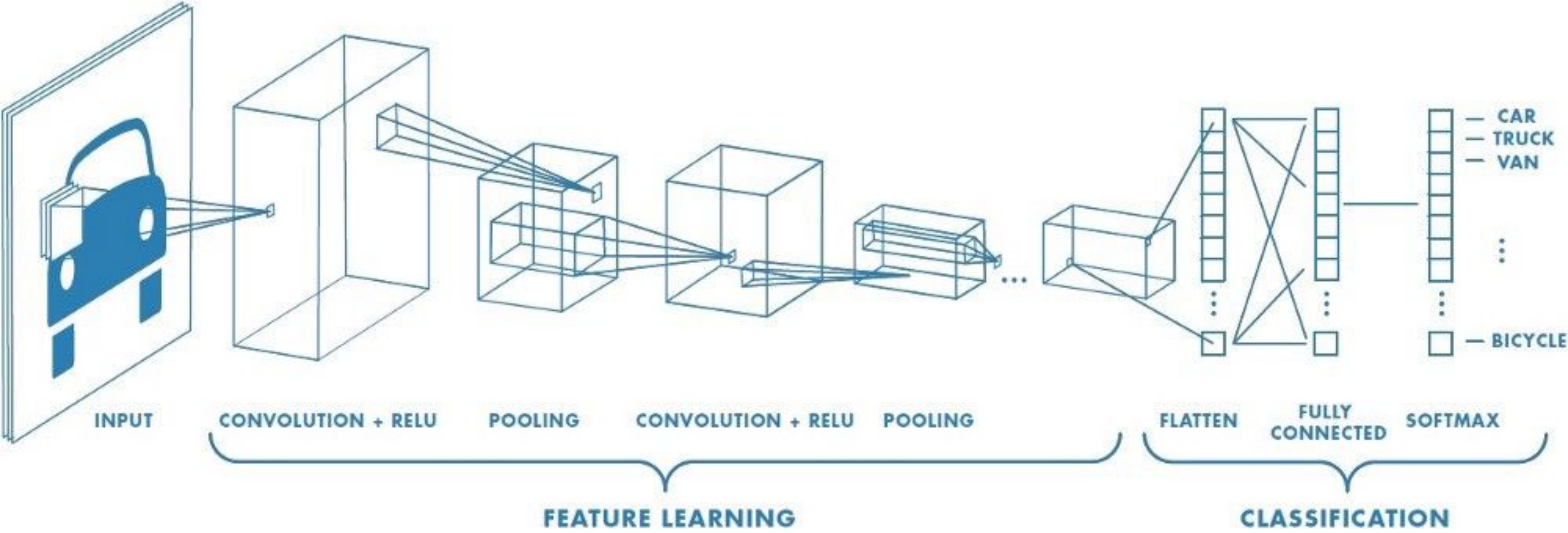


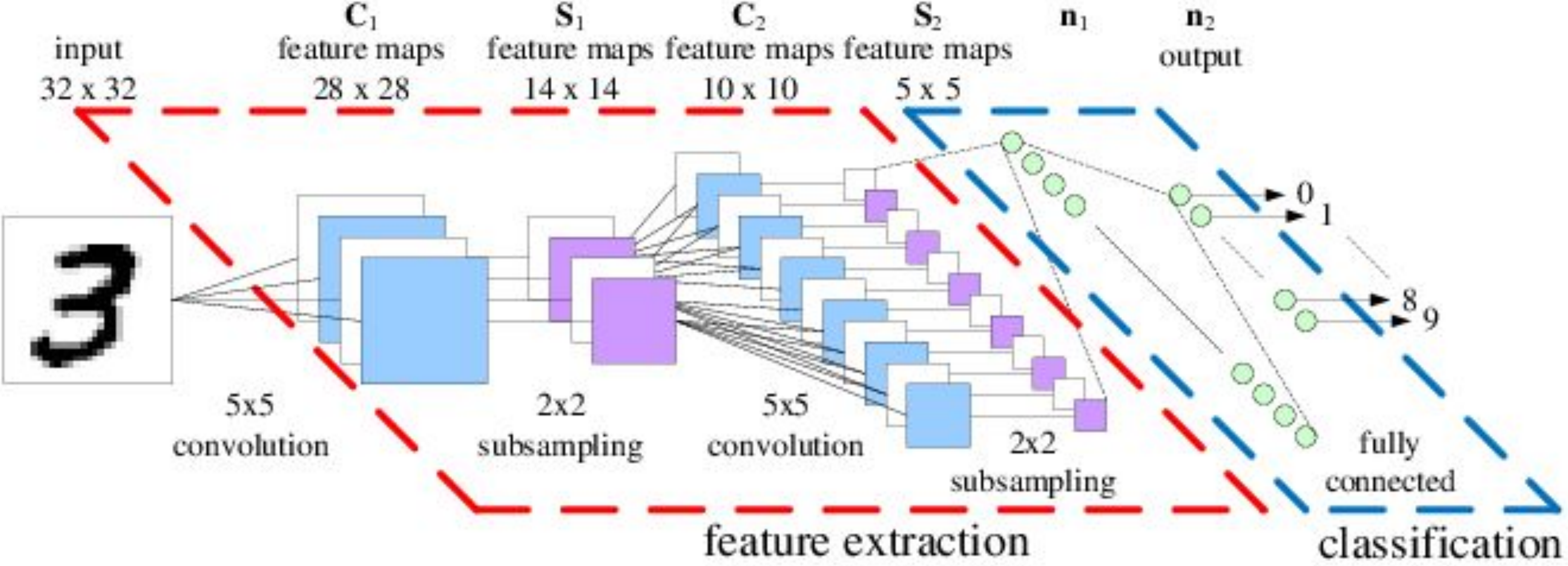
Feedfoward, Backpropagation, Overfitting



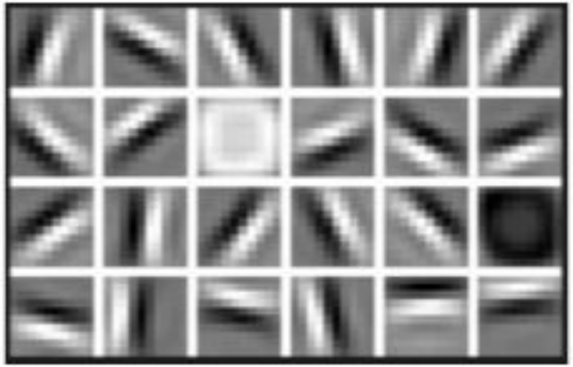
Deep Learning

- Deep learning architectures combine feature extraction with supervised learning
- Feature extraction happens in a hierarchy of multiple feature extraction layers
- Initial layers deal with primitive, low-level features, and each layers output is the input for the next layer
- All features are used in the final supervised learning model


CAR
TRUCK
VAN
BICYCLE
INPUT
CONVOLUTION + RELU
POOLING
CONVOLUTION + RELU
POOLING
FLATTEN
FULLY CONNECTED
SOFTMAX
FEATURE LEARNING
CLASSIFICATION



input
32 x 32
C1
feature maps
28 x 28
S1
feature maps
14 x 14
C2
feature maps
10 x 10
S2
feature maps
5 x 5
n1
n2
output
0
1
8
9
5x5
convolution
2x2
subsampling
5x5
convolution
2x2
subsampling
fully
connected
feature extraction
classification



First Layer Representation

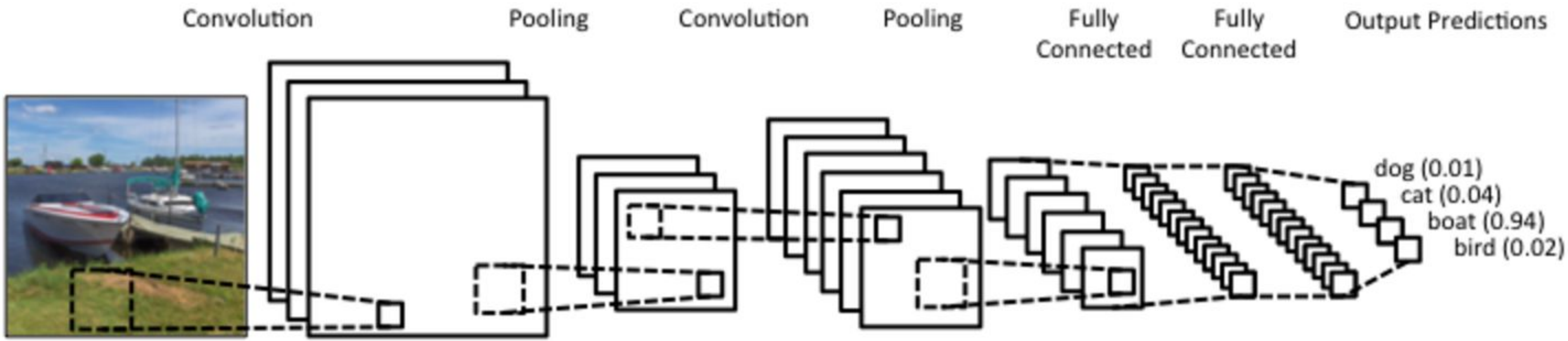
Second Layer Representation

Third Layer Representation

Convolutional Neural Networks

Inspired by animal visual cortex neurons

Applications in image and video recognition, recommender systems, and natural language processing



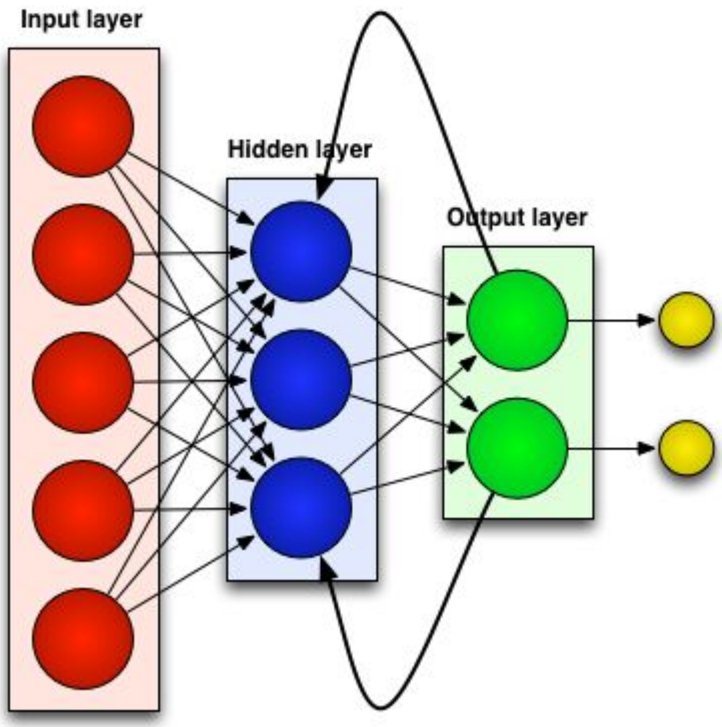
Recurrent Neural Networks

Contain loops

Implements feedback and gives the network "memory" or context

Good for predicting sequences, translating text, predicting objects in images, speech translation

LSTM is a Recurrent Neural Network



Neural Networks

Pros

Flexible

Good for a variety of tasks

Good for many types of data

Can capture lots of complexity

Good for when you have lots of similar feature types

Cons

Require a lot of data

Training may be slow

Preprocessing needed

Many parameters to tune

Many layer types and activations to choose from

Bad for when feature types are very different

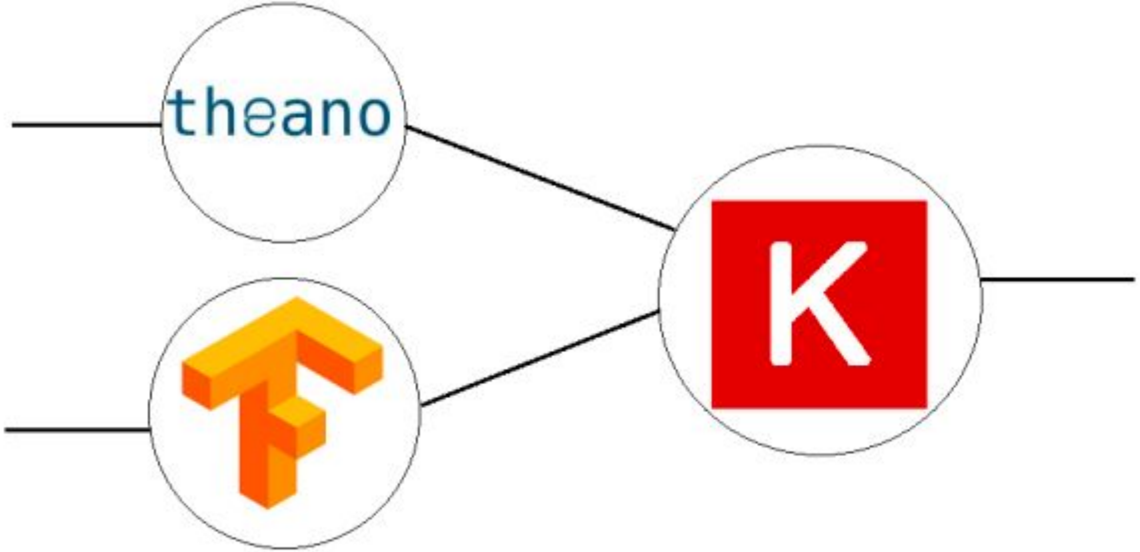
Black Box model

Deep Learning Frameworks

Caffe Chainer DL4J Deeplearning4j KERAS Microsoft CNTK MatConvNet

MINERVA mxnet Purine TensorFlow theano torch

Keras, Theano, Tensorflow



Practical Considerations

Overfitting (Dropout, Regularization)

Learning Rate Schedules

Interpretability vs. Performance

MiniBatch

Overfitting

Dropout

Regularization

Solver

Atom tends to work for most large datasets

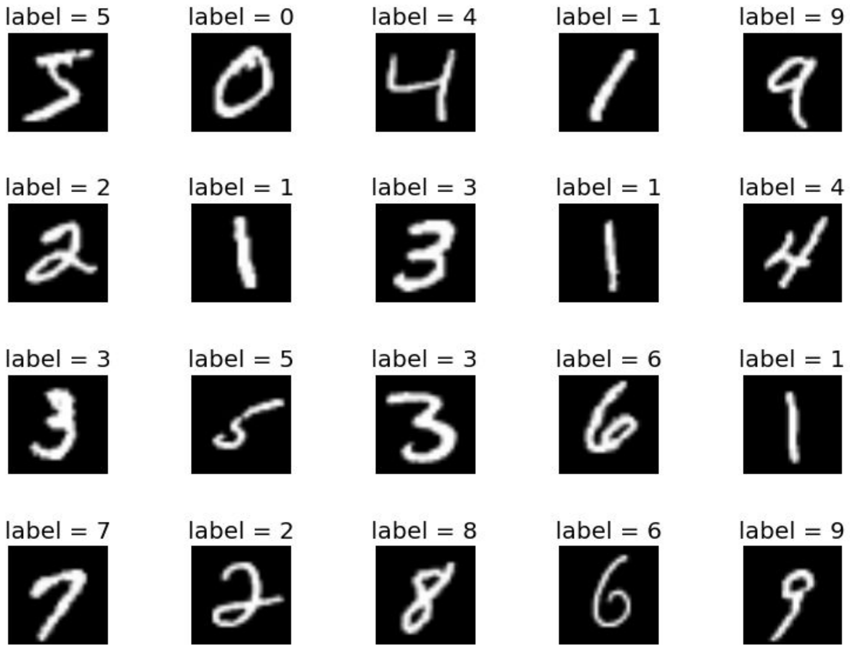
LBFG is good for smaller datasets

The MNIST Dataset
(the Hello World of Machine Learning)

http://yann.lecun.com/exdb/mnist/

Split into three parts (training, testing, and validation)

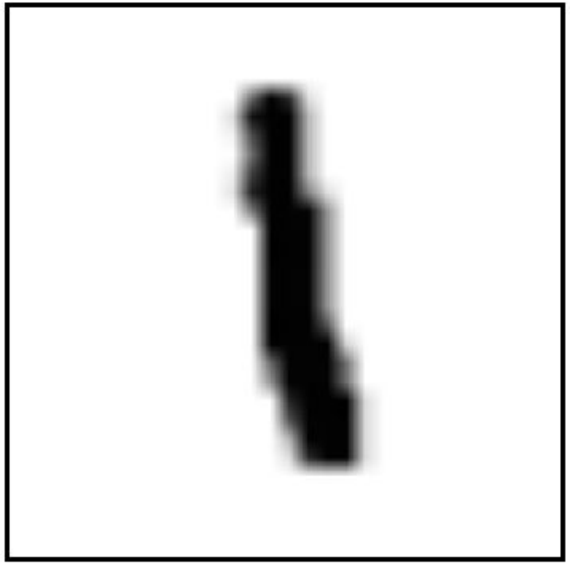
Each data point has two parts: image (“x”) and label (“y”)



MNIST Dataset

Each image is 28 by 28 pixels. We can interpret this as a big array of numbers.

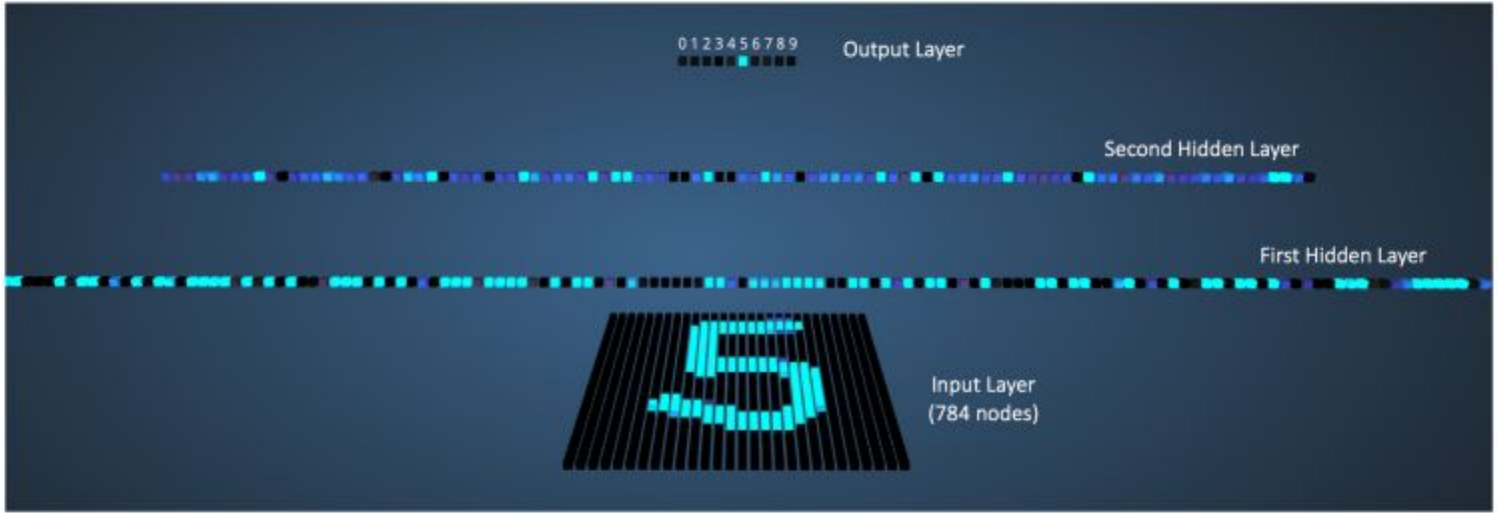
We can flatten this array into a vector of numbers (28 x 28 = 784 numbers).



$$\simeq \begin{bmatrix}
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & .6 & .8 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & .7 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & .7 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & .5 & 1 & .4 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & .4 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & .4 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & .7 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & .9 & 1 & .1 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & .3 & 1 & .1 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0
\end{bmatrix}$$

Visualizing a Multilayer Neural Network

http://scs.ryerson.ca/~aharley/vis/fc/



Build a Multilayer Neural Network to Classify the MNIST Dataset with Keras