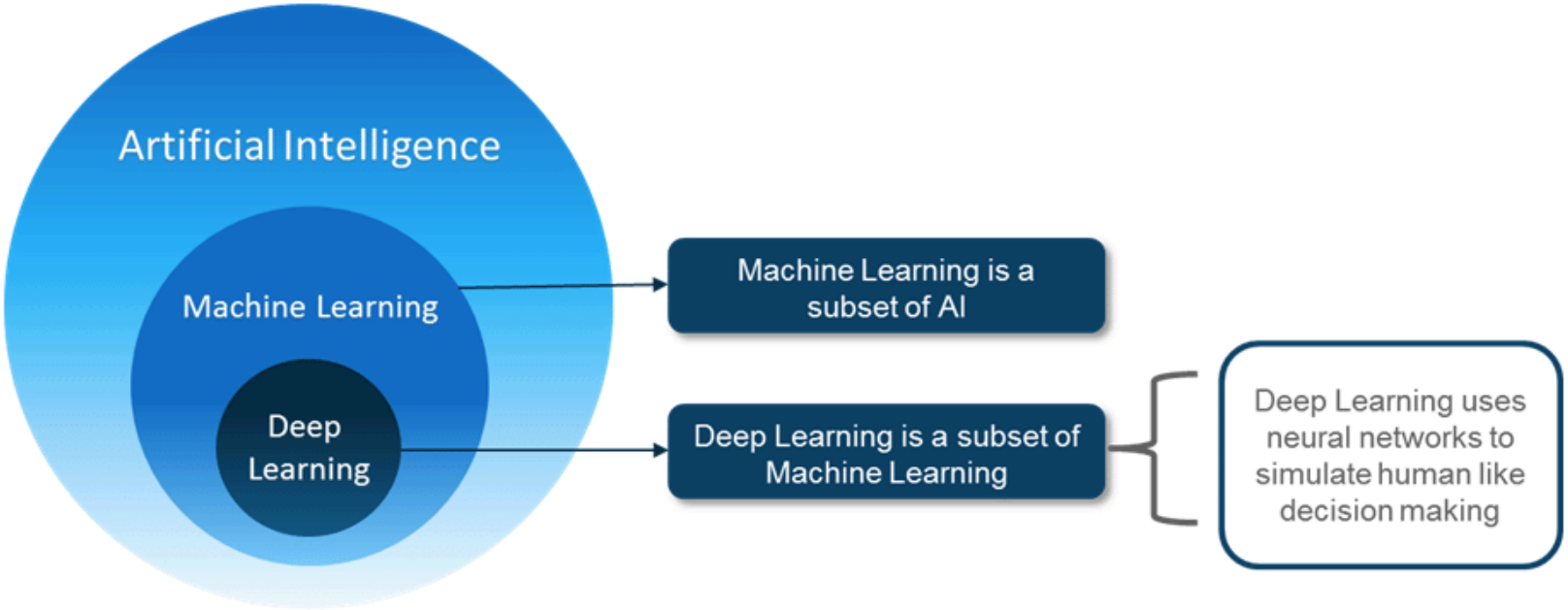
iknowlation
knowledge embedded in innovation

Machine Learning



Ref:https://www.edureka.co/blog/deep-learning-with-python/

Machine Learning

Types of ML (Neural / Non-neural)

Supervised

- Knn, NB, SVM

Unsupervised

- Kmeans, Clustering, boosting

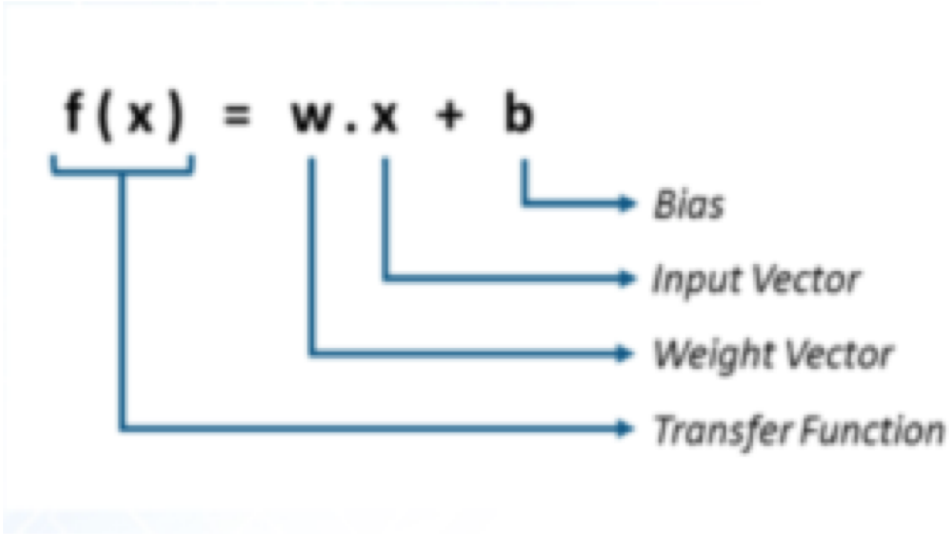
Reinforcement

- Q-Learning, DQN, DDPG

Incremental

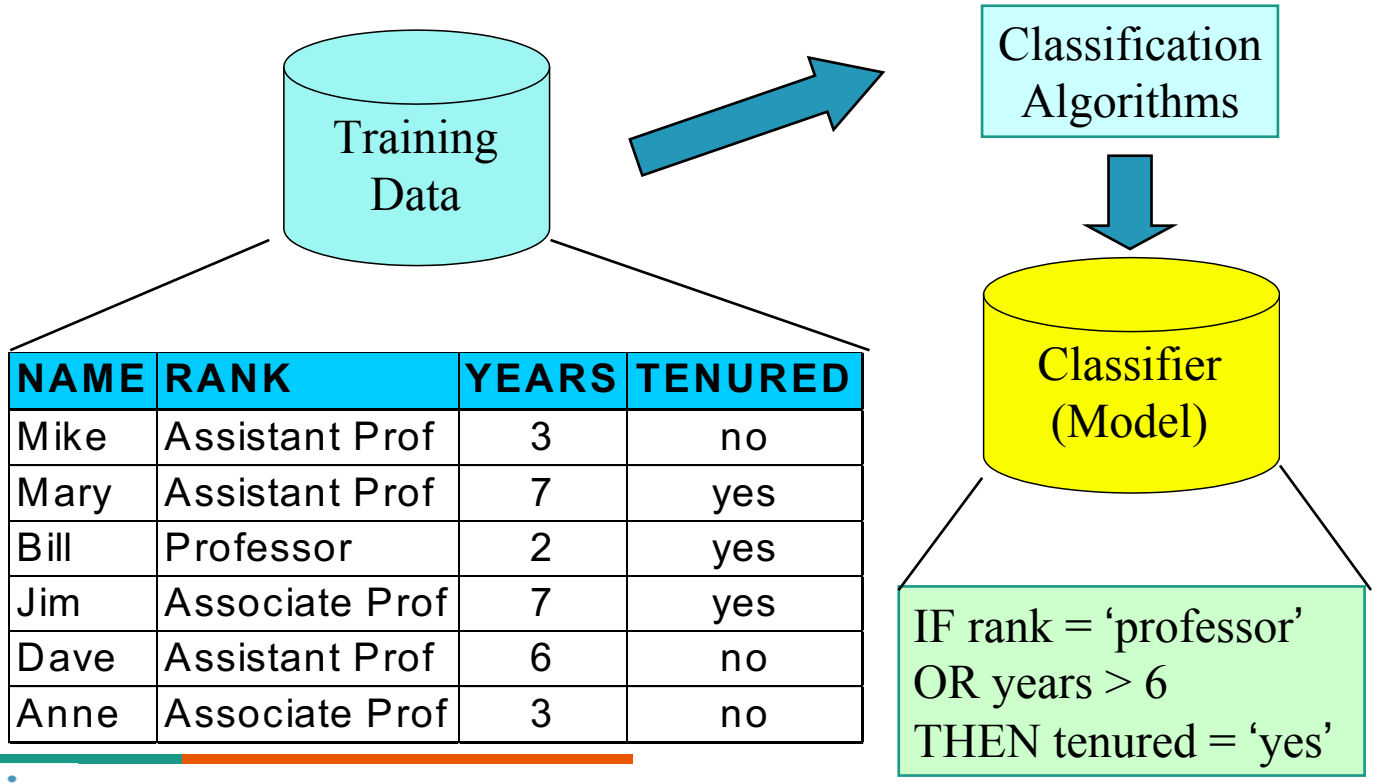
- Fuzzy ARTAMP, Extension of most of the above algos

Learning Function



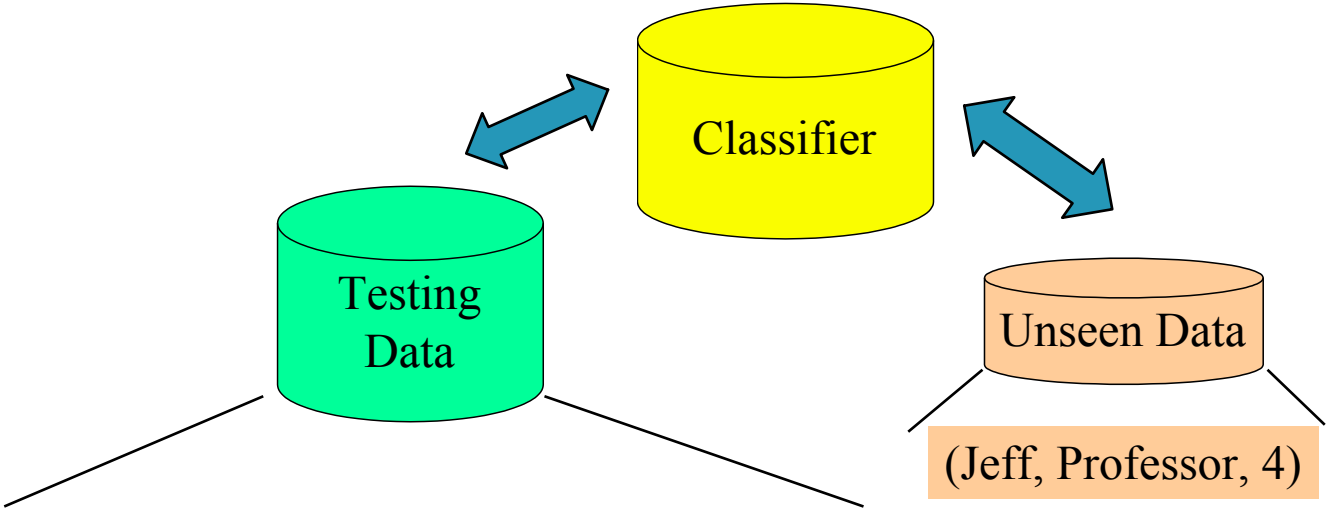
http://localhost:8888/notebooks/Documents/Metamagics/Perceptron.ipynb

Simple Model Construction



NAME	RANK	YEARS	TENURED
Mike	Assistant Prof	3	no
Mary	Assistant Prof	7	yes
Bill	Professor	2	yes
Jim	Associate Prof	7	yes
Dave	Assistant Prof	6	no
Anne	Associate Prof	3	no

Using the Model in Prediction



NAME	RANK	YEARS	TENURED
Tom	Assistant Prof	2	no
Merlisa	Associate Prof	7	no
George	Professor	5	yes
Joseph	Assistant Prof	7	yes

Cosine Similarity

- If d_1 and d_2 are two document vectors, then

$$\cos(d_1, d_2) = (d_1 \bullet d_2) / ||d_1|| \, ||d_2|| ,$$

where $\bullet$ indicates vector dot product and $||d||$ is the length of vector d.

- Example:

d_1 = **3 2 0 5 0 0 0 2 0 0**
d_2 = **1 0 0 0 0 0 0 1 0 2**

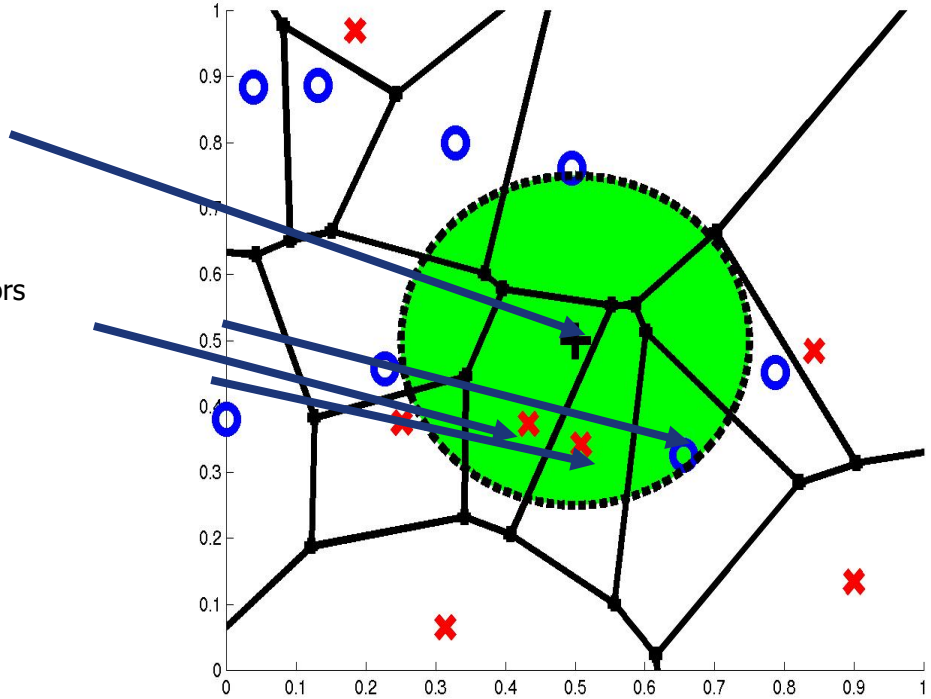
$d_1 \bullet d_2$= 3*1 + 2*0 + 0*0 + 5*0 + 0*0 + 0*0 + 0*0 + 2*1 + 0*0 + 0*2 = 5
$||d_1||$ = (3*3+2*2+0*0+5*5+0*0+0*0+0*0+2*2+0*0+0*0)$^{0.5}$ = (42) $^{0.5}$ = 6.481
$||d_2||$ = (1*1+0*0+0*0+0*0+0*0+0*0+0*0+1*1+0*0+2*2) $^{0.5}$ = (6) $^{0.5}$ = 2.245

$\cos(d_1, d_2)$ = .3150, distance=1-cos(d1,d2)

3-Nearest Neighbors

query point q_f

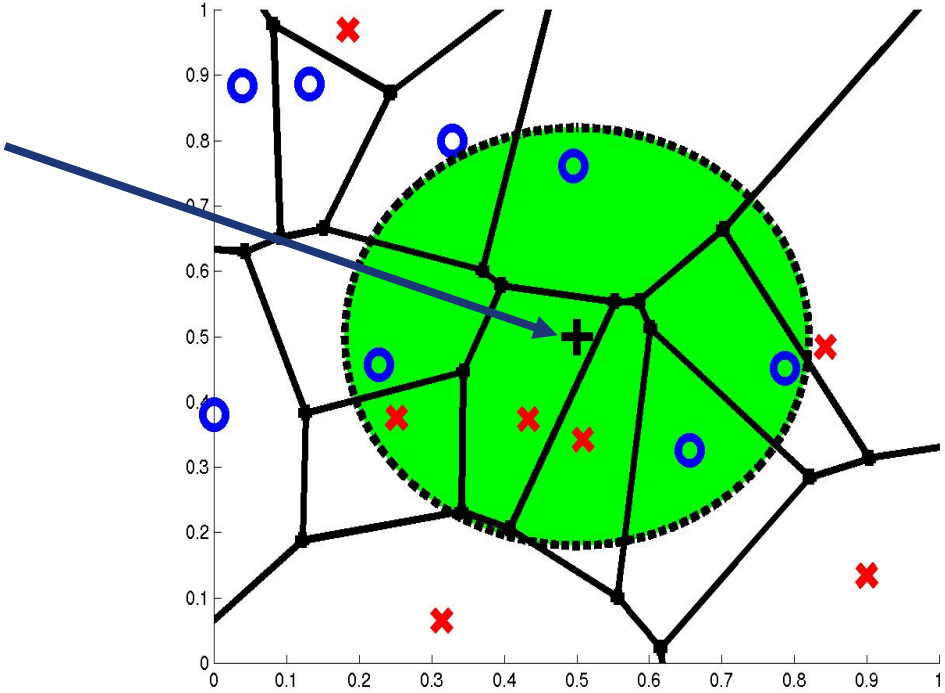
3 nearest neighbors



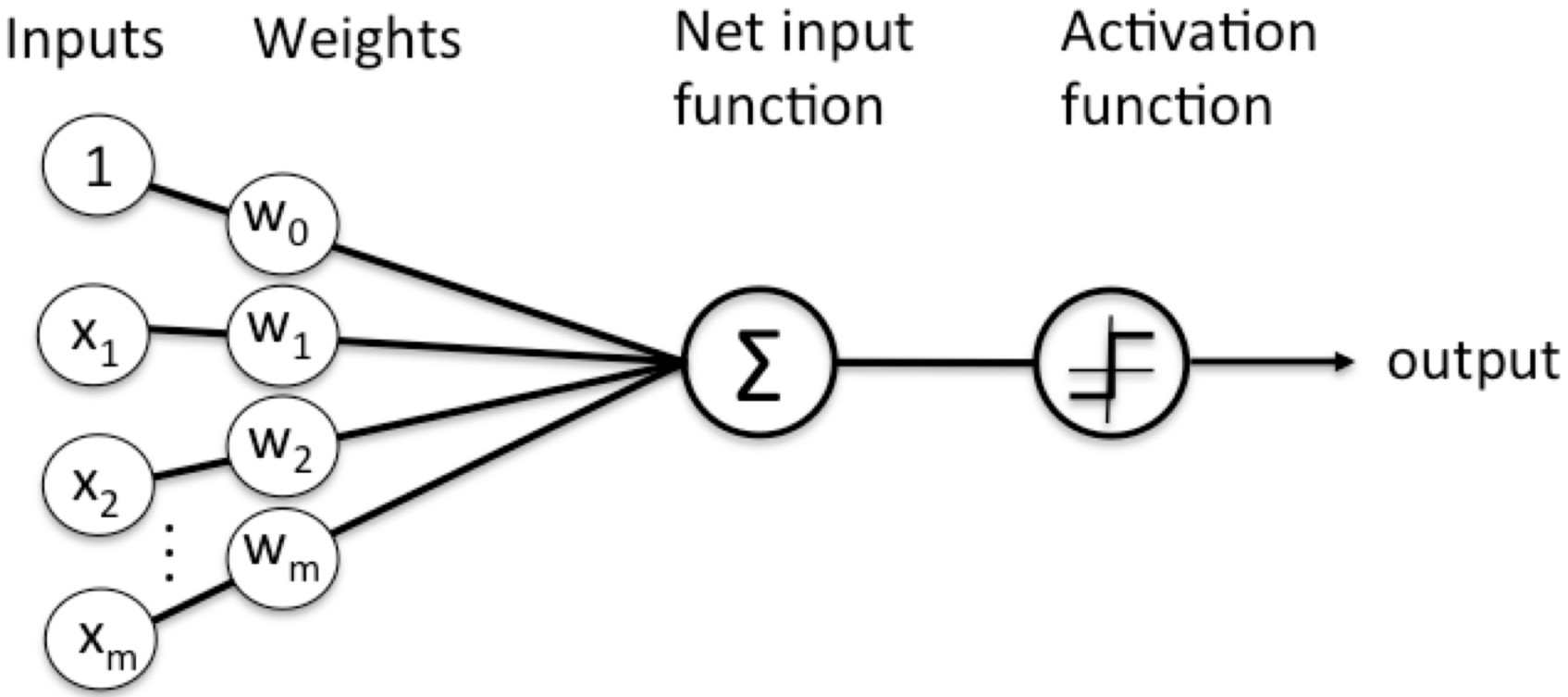
7-Nearest Neighbors

query point q_f

7 nearest neighbors



Deep Learning



Deep Learning

- **What is it:**
  Extract useful patterns from data.
- **How:**
  Neural network + optimization
- **How(Practical):**
  Python + TensorFlow
- **Hard Part:**
  Good Questions + Good Data
- **Why now:**
  Data, hardware, community, tools, investment
- **Where do we stand?**
  Most big questions of intelligence have not been answered nor properly formulated

- **Exciting progress:**
  - Face recognition
  - Image classification
  - Speech recognition
  - Text-to-speech generation
  - Handwriting transcription
  - Machine translation
  - Medical diagnosis
  - Cars: drivable area, lane keeping
  - Digital assistants
  - Ads, search, social recommendations

  - Game playing with deep RL

Deep Learning vocabulary

- Deep Learning, Machine Learning, Artificial Intelligence
- Deep Belief Net, Artificial Neural Net, Boltzmann Machine
- Convolutional Neural Nets
- Recurrent NN & LSTM (Long Short Term Memory)
- Activation Function: ReLU (Rectified Linear Unit)
- Deep Learning libraries and frameworks

  -- TensorFlow, Caffe, Theano, Torch, DL4J
- Backpropagation, gradient descent, loss function

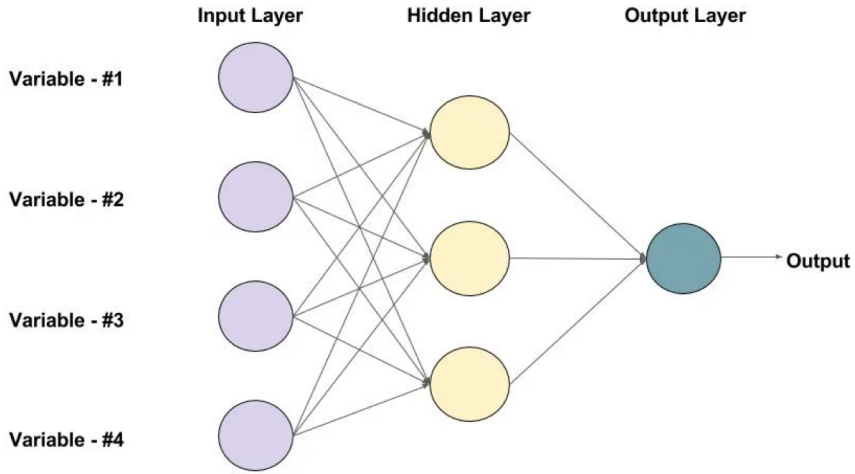
History of Deep Learning Ideas, Milestones, Tools

- 1943:Neuralnetworks
- 1957:Perceptron
- 1974-86:Backpropagation,RBM,RNN
- 1989-98: CNN, MNIST, LSTM, Bidirectional RNN
- 2006:“DeepLearning”,DBN
- 2009:ImageNet
- 2012:AlexNet,Dropout
- 2014: GANs
- 2014:DeepFace
- 2016:AlphaGo
- 2017:AlphaZero,CapsuleNetworks
- 2018:BERT

- Mark 1 Perceptron – 1960
- Torch–2002
- CUDA–2007
- Theano – 2008
- Caffe – 2014
- DistBelief – 2011
- TensorFlow 0.1 – 2015
- PyTorch0.1–2017
- TensorFlow 1.0 – 2017
- PyTorch1.0–2017
- TensorFlow 2.0 – 2019

Neural network -> Deep Learning

- Multiple - Layers
- Feed Forward Network - do not form cycles
- Backpropagation - Cycle in network node



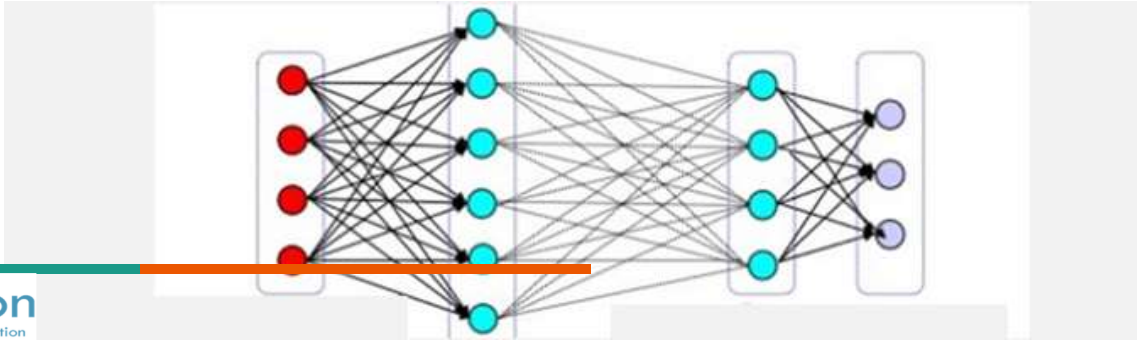
An example of a Feed-forward Neural Network with one hidden layer (with 3 neurons)

Deep Learning vs Machine Learning

- Handling of High Dimensional Data
- Feature Selection
- universal approximator f(x) = y

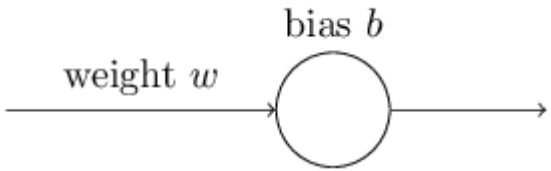
What happens in the Hidden Layers

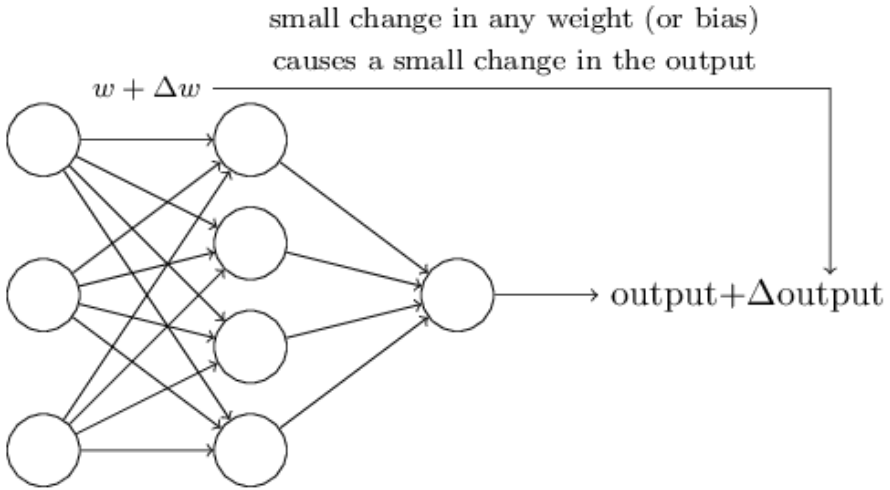
- First layer learns primitive features (line, edge, tiniest unit of sound) by finding combinations of the input vector data that occur more frequently than by chance
- Logistic regression performed and encoded at each processing node (Y/N (0,1)), does this example have this feature?
- Feeds these basic features to next layer, which trains itself to recognize slightly more complicated features (corner, combination of speech sounds)
- Feeds features to new layers until recognizes full objects



How does the neural net actually learn?

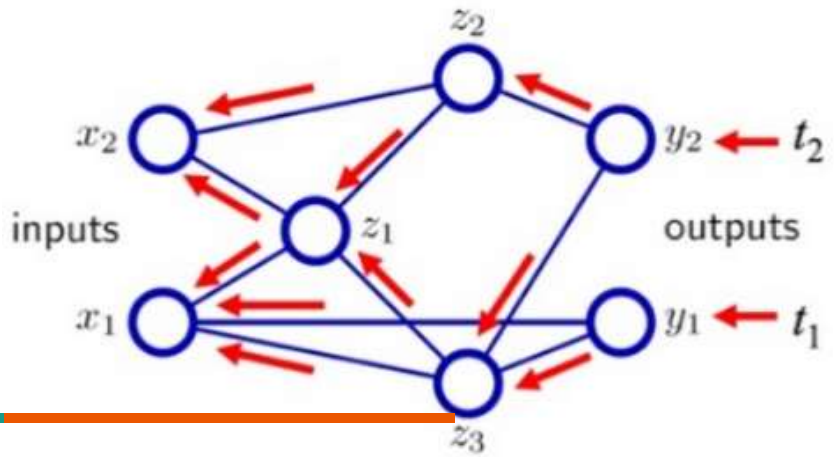
- Structural system based on cascading layers of neurons with variable parameters: weight and bias
- Vary the weights and biases to see if a better outcome is obtained
- Repeat until the net correctly classifies the data





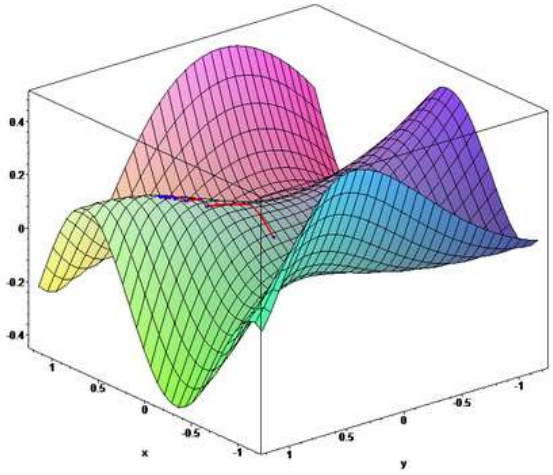
FF and Backpropogation - How #W + B combinations works

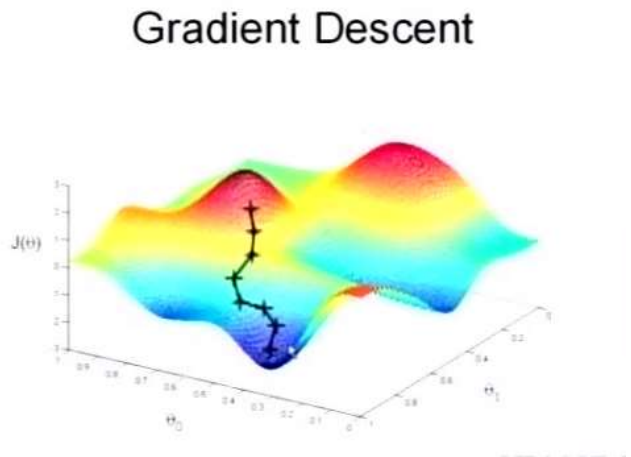
- Problem: Inefficient to test the combinatorial explosion of all possible parameter variations
- Backpropagation is an optimization method used to calculate the error contribution of each neuron after a batch of data is processed



Gradient Descent

- Gradient: derivative to find the minimum of a function
  - Gradient descent: optimization algorithm to find the biggest errors (minima) most quickly
  - Error = MSE, log loss, cross-entropy; e.g.; least correct predictions to correctly identify data





http://www.briandolhansky.com/blog/2013/9/27/artificial-neural-networks-backpropagation-part-4

Learning Model: Components

- Data (Train, Test)
- Model
- Cost Function
- Optimization Function
- Evaluation Matrix

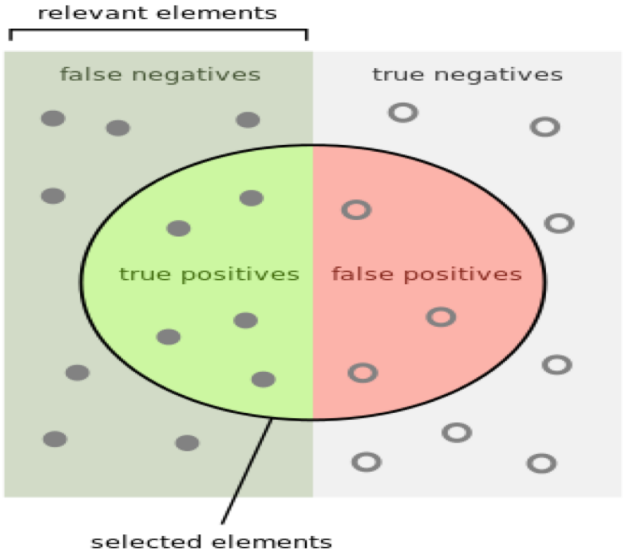
Accuracy Metric

Accuracy

F1 – Measure

Precision

Recall



How many selected items are relevant?

$$\text{Precision} = \frac{\text{true positives}}{\text{selected elements}}$$

How many relevant items are selected?

$$\text{Recall} = \frac{\text{true positives}}{\text{relevant elements}}$$

TensorFlow

What is it: Deep Learning Library *(and more)*

Facts : Open Source, Python, Google

Community:

117,000+GitHubstars

TensorFlow.org: Blogs, Documentation, Dev Summit, YouTube talks

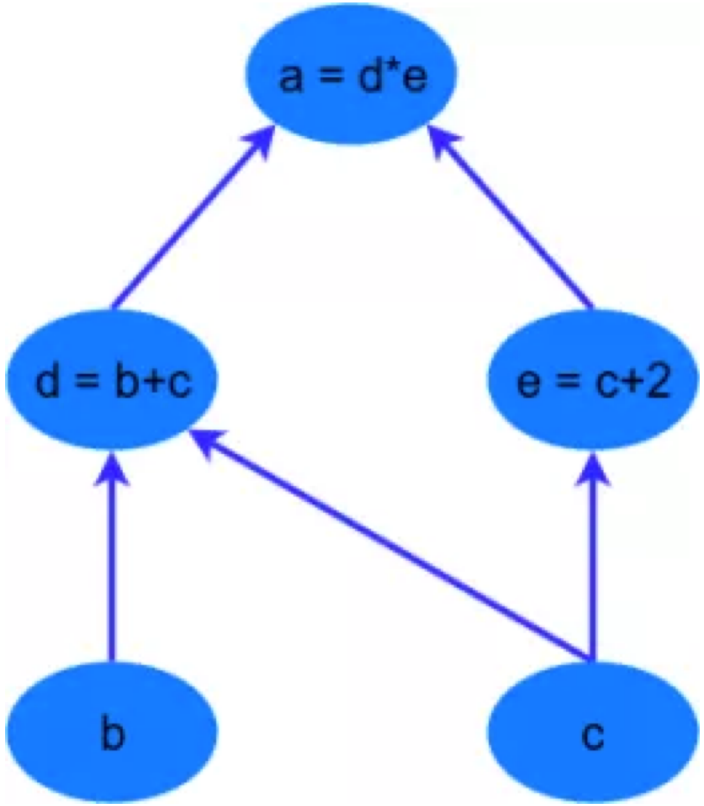
Extras:

•Swift for TensorFlow
•TensorFlow Serving
•TensorFlow Extended (TFX)
•TensorFlow Probability
•Tensor2Tensor

• **Ecosystem:**
•Keras: high-level API
•TensorFlow.js: in the browser
•TensorFlow Lite: on the phone
•Colaboratory: in the cloud
•TPU: optimized hardware
•TensorBoard: visualization
•TensorFlowHub:graphmodules
•• **Alternatives**: PyTorch, MXNet, CNTK

TensorFlow

Tensor - Multidimensional Array
Flow - Graph



Notebook

http://localhost:8888/notebooks/Documents/Metamagics/Basics_tensorflow_1.ipynb

Supervised Learning