Introduction to Neural Networks

Data Science Immersive

FLATIRON SCHOOL

Agenda today

- A brief history of Neural Networks
- Motivations for Neural Network
- Intuition of NN - hidden layers, forward propagation, and backpropagation

Students will be able to...

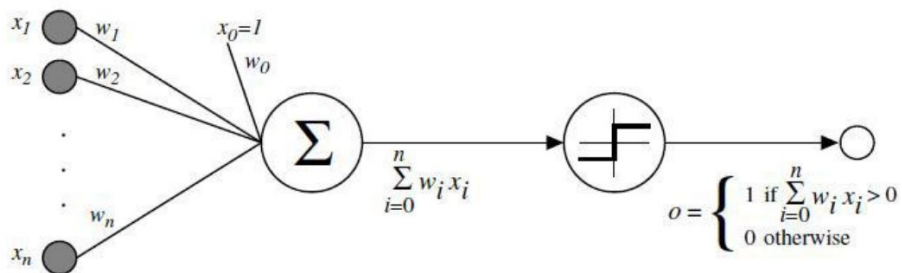
- Understand the motivation and need for neural nets
- Understand and explain the similarity and difference between linear regression and neural nets
- Explain the intuition behind a vanilla neural net

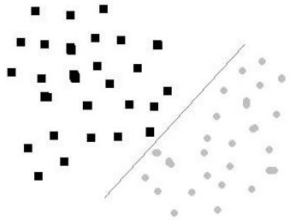
Neural Networks as representation learning

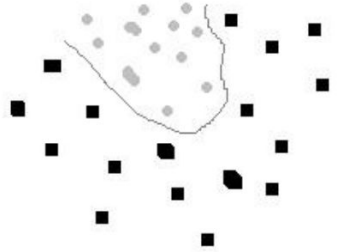
- So much of machine learning is using mathematics and statistics to represent how we as human perceive, interpret and make sense of certain phenomena
  - Natural Language Processing
  - Decision Trees
  - Linear regression
- Neural network is no exception - it is originally used to model how our neurons activate when presented with certain stimuli

Brief History of Neural Networks

- Motivated by the desire to harness the thinking process, William McCulloch and Walter Pitts wrote a paper on how neurons might work by modeling the logical process of AND/OR/NOT in 1943
- Inspired by Pitts and McCulloh, Hebb published Organization of Behavior in 1949 which proposed that neural pathways are strengthened each time after stimulation
- Frank Rosenblatt, a Psychologist, created the first Perceptron (Mark I Perceptron) 1958 that is capable of separating linearly separable cases, but failed at non-linear ones.
- Research later found out that the Mark I perceptron did not fail at its ability to make prediction, but scale

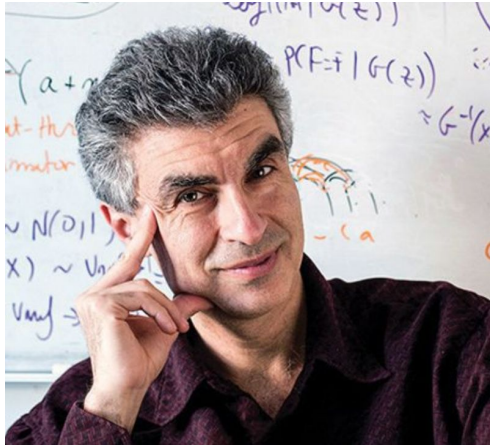






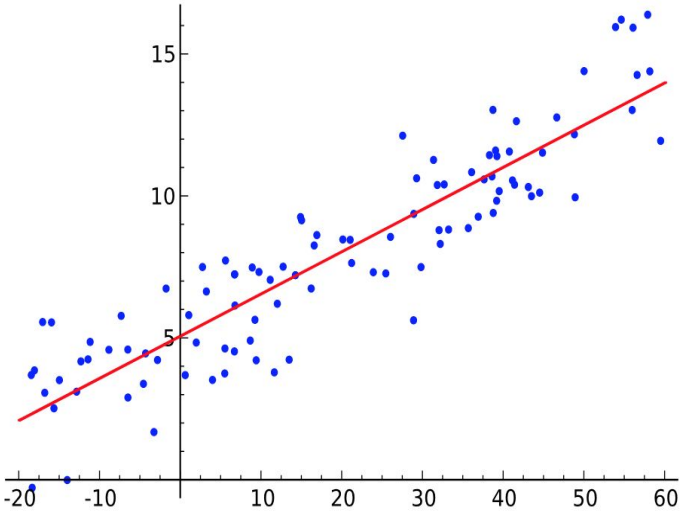
Brief History of Neural Networks

- Research in NN was dormant for a while until the advent of backpropagation emerge (1975), which effectively solved the problem of non-linearly separable cases by creating multiple layers and train the weights and biases by propagating the error term back up through the layers.
- Max-pooling is introduced in 1992
- Between 2009 and 2012, recurrent neural nets were developed
- Ian Goodfellow and others developed GANs (Generative Adversarial Networks)



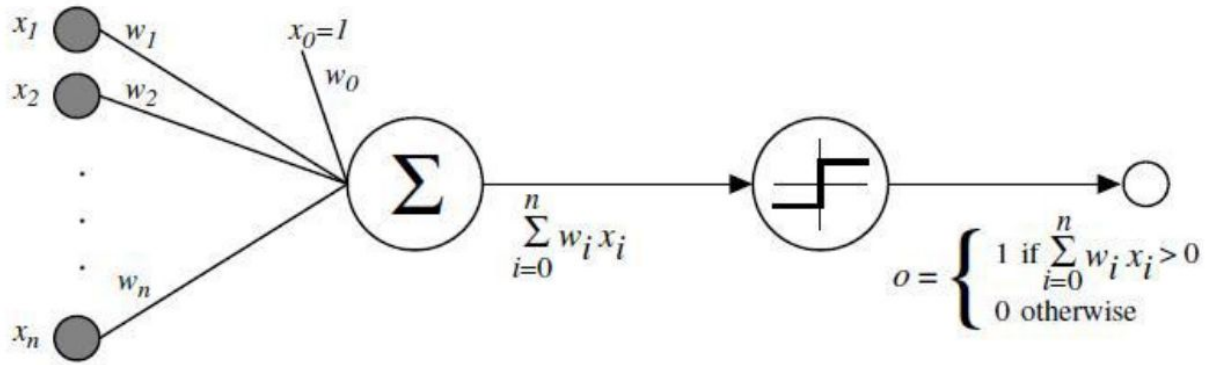
Why Neural Nets?

- Before jumping into concepts in artificial neural networks, let's revisit our old friend linear regression
- An example would be how many job offers received based on predictors such as:
  - Age
  - Years of working experiences
  - Gender
  - Field of expertise
  - Number of people networked with
  - etc...



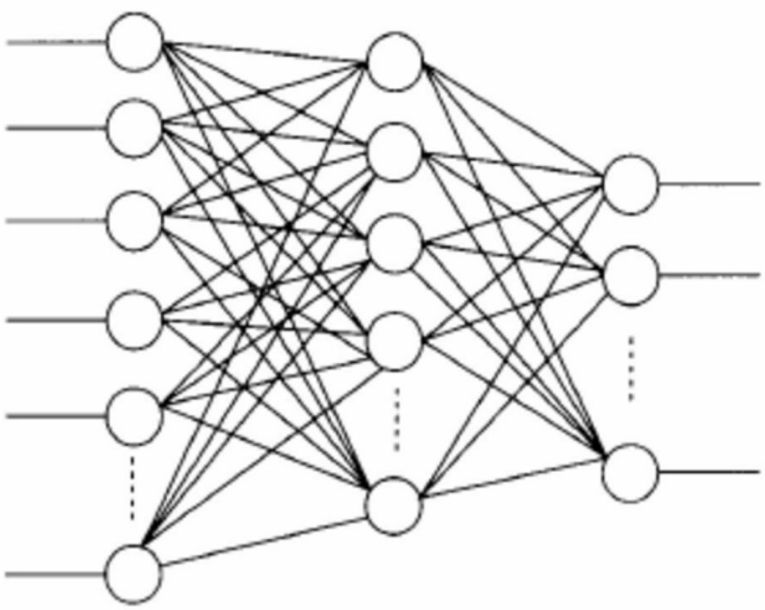
What is Neural Nets?

- An artificial neural network is a network of simple elements called artificial neurons, which receive input, change their internal state (activation) according to that input, and produce output depending on the input and activation.

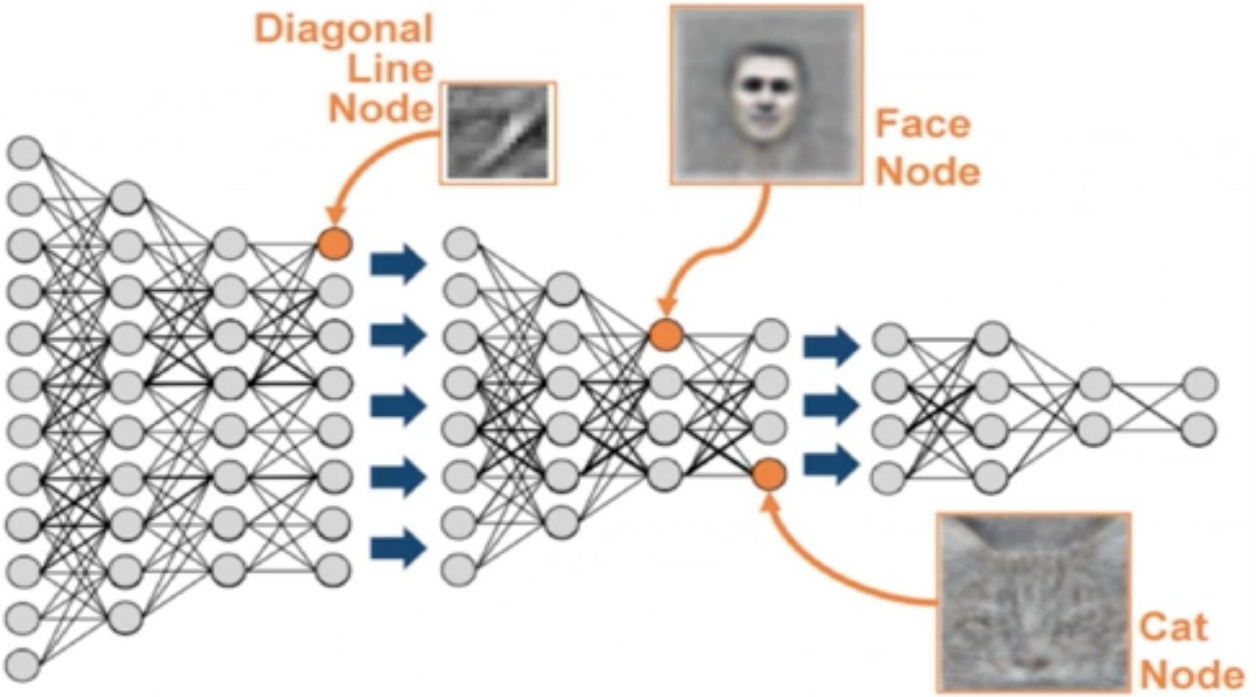


Why Neural Networks

- Instead of modeling the relationship of variables as strictly linear, nn can capture and account for the interaction between variables really well
- How does NN behave intelligently?

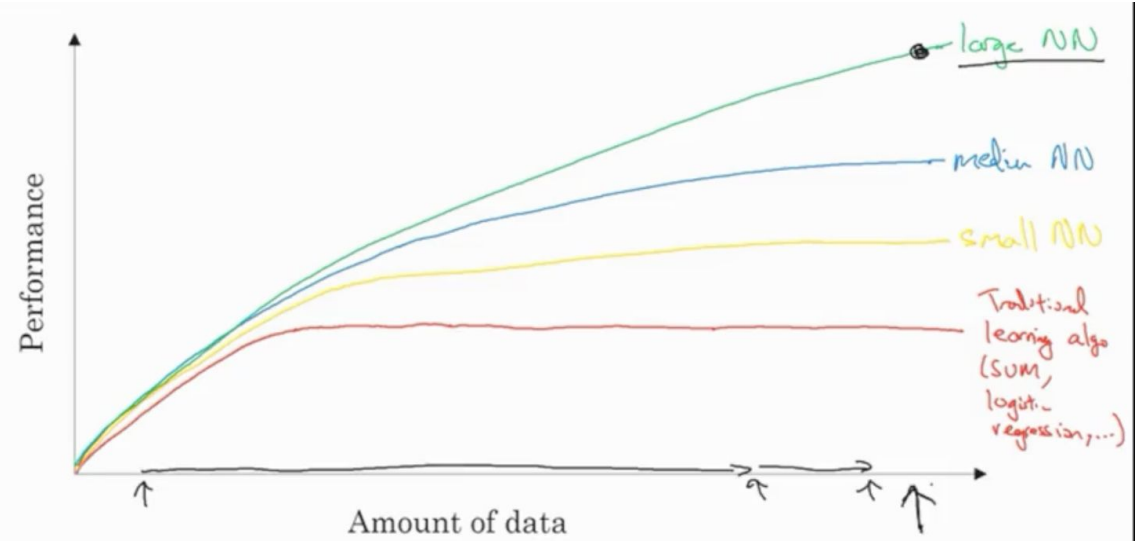


Why Neural Networks



Why Neural Networks?

- Why have neural networks really taken off and became a major area of research in the past few years?



Why Neural Networks?

- Data
- Computation
  - CPU and GPU
- Algorithmic improvements
  - Changing activation function to make your nn run faster

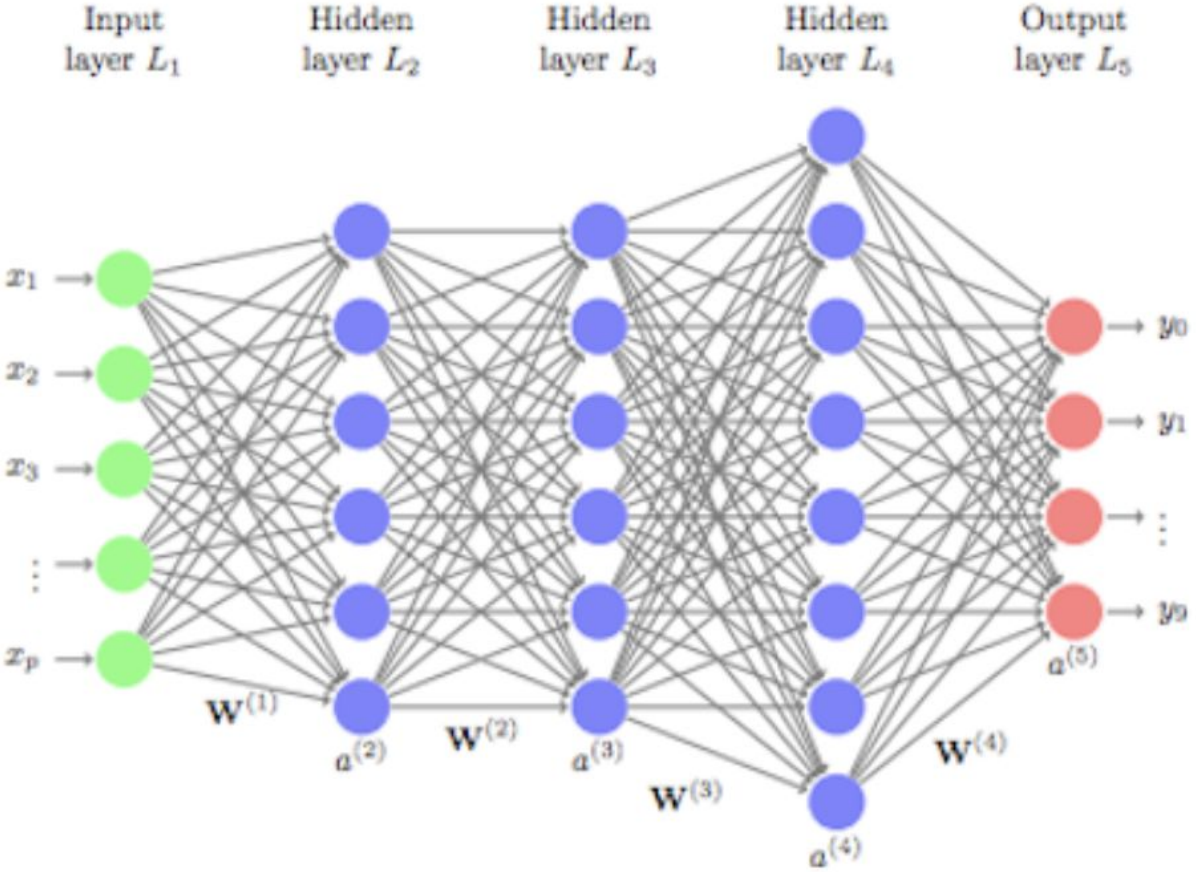
Why Neural Networks?

- Deep neural network internally build up representation of patterns in the data, and they go through increasingly complex patterns as we go through successively hidden layers of the networks.
- Therefore, NN partially replace the need for feature engineering
- because as the layers increase, the more sophisticated the representations are built from raw data

How Neural Nets work?

- As mentioned earlier, neural networks can learn various patterns that underlie our data, structured and unstructured
- The networks represent the pattern in our data by learning and adjusting the weight associated with each input at each layer
- What we do to train our networks is assign a *weights* on the connection between our neurons and the next layer, and compute the weighted sum of that layer
- We can then apply a certain activation function, e.g. Sigmoid, to confine the weighted sum between 0 to 1 to signify classes
- We also want to apply a bias to our weighted sum, depending on the specific question

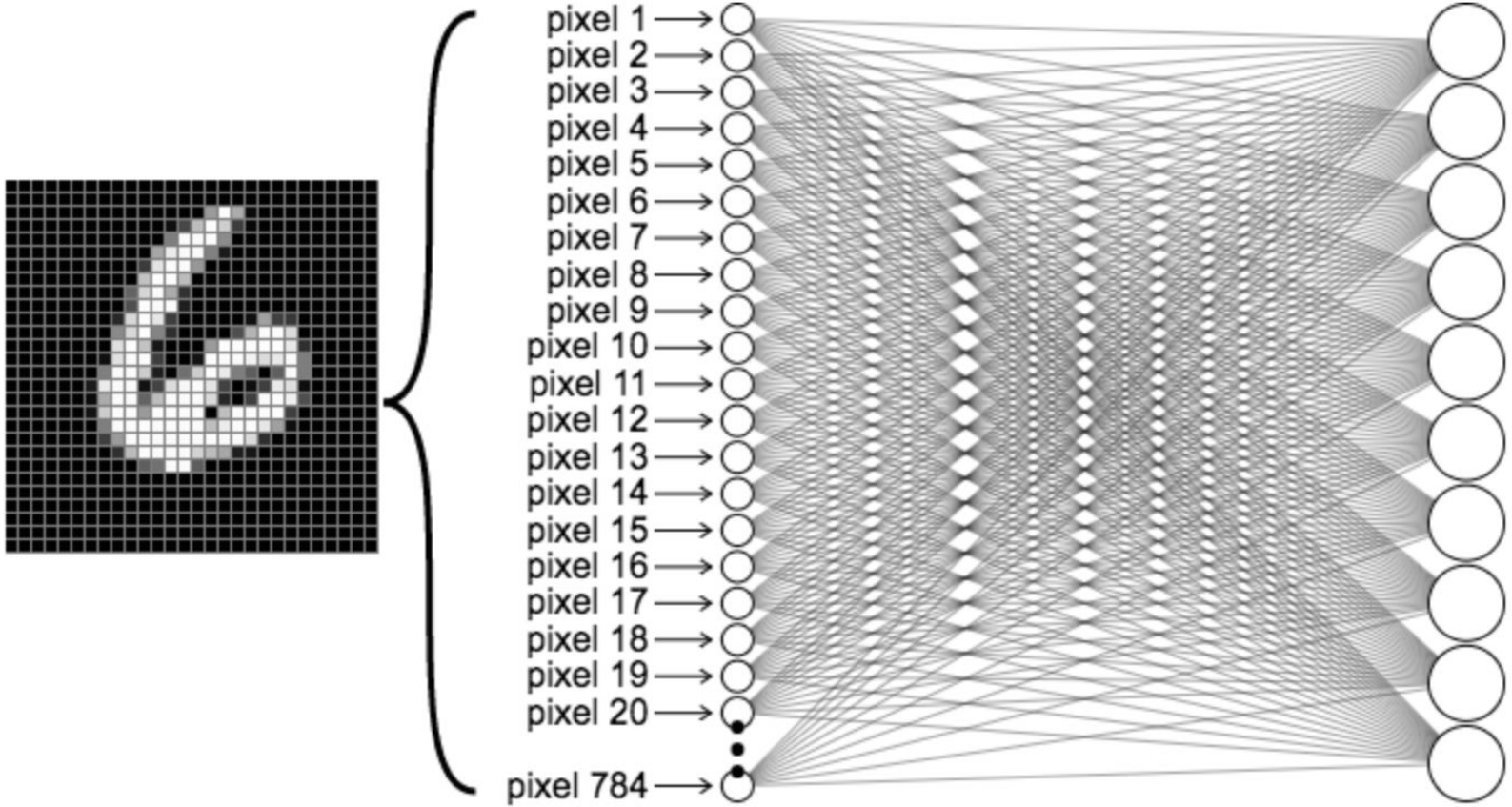
How Neural Nets work?



How Neural Nets work?

- The weight associated with each neuron tells us which signal it’s picking up on
- The bias controls how high or low the weighted sum needs to be before the neuron gets activated
- Each neuron has its own weight and bias associated with the next layer, indicating the specific pattern
- Deep **learning** → get our machines to learn the optimal weight and biases that model this process by propagating through multiple layers
- The above process and be represented as the vectorized format

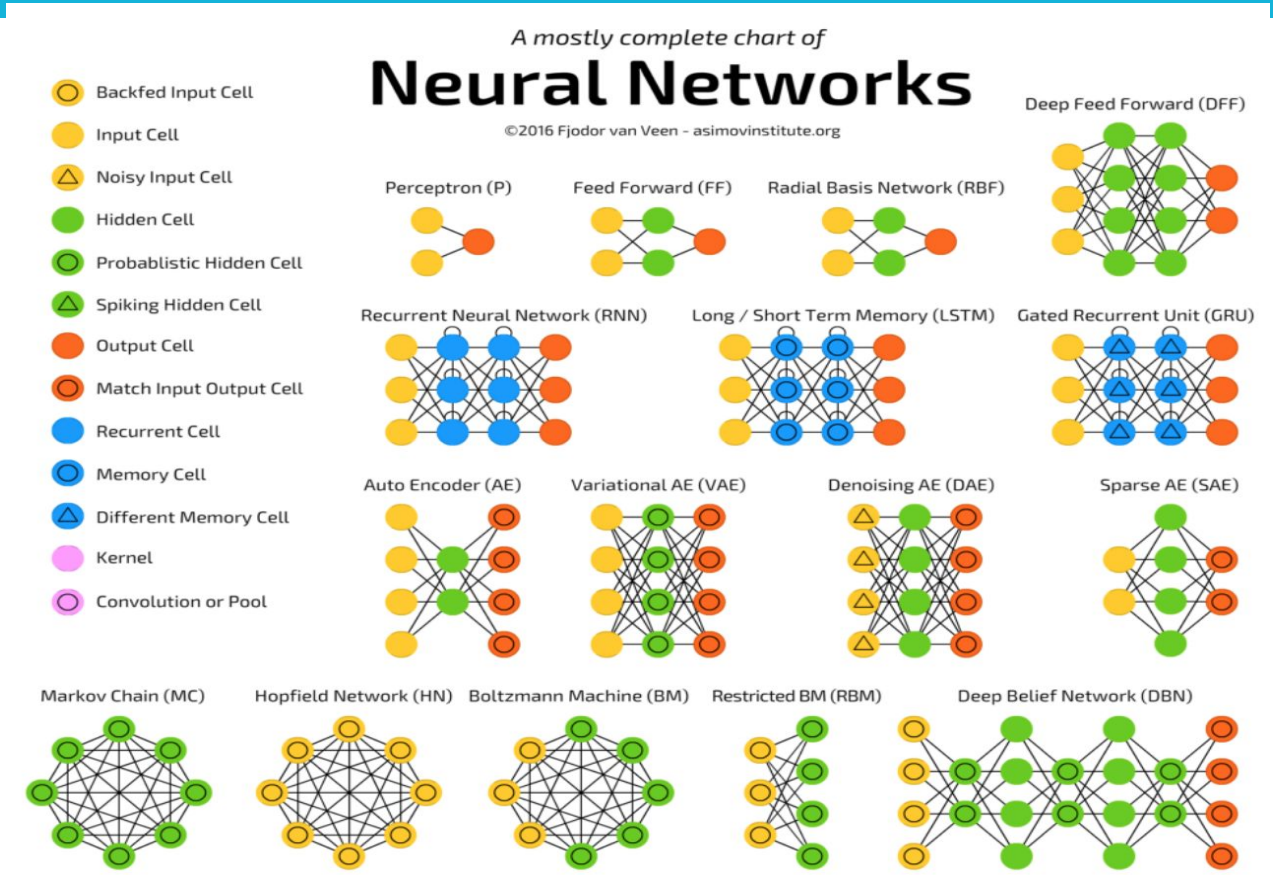
How Neural Nets work?



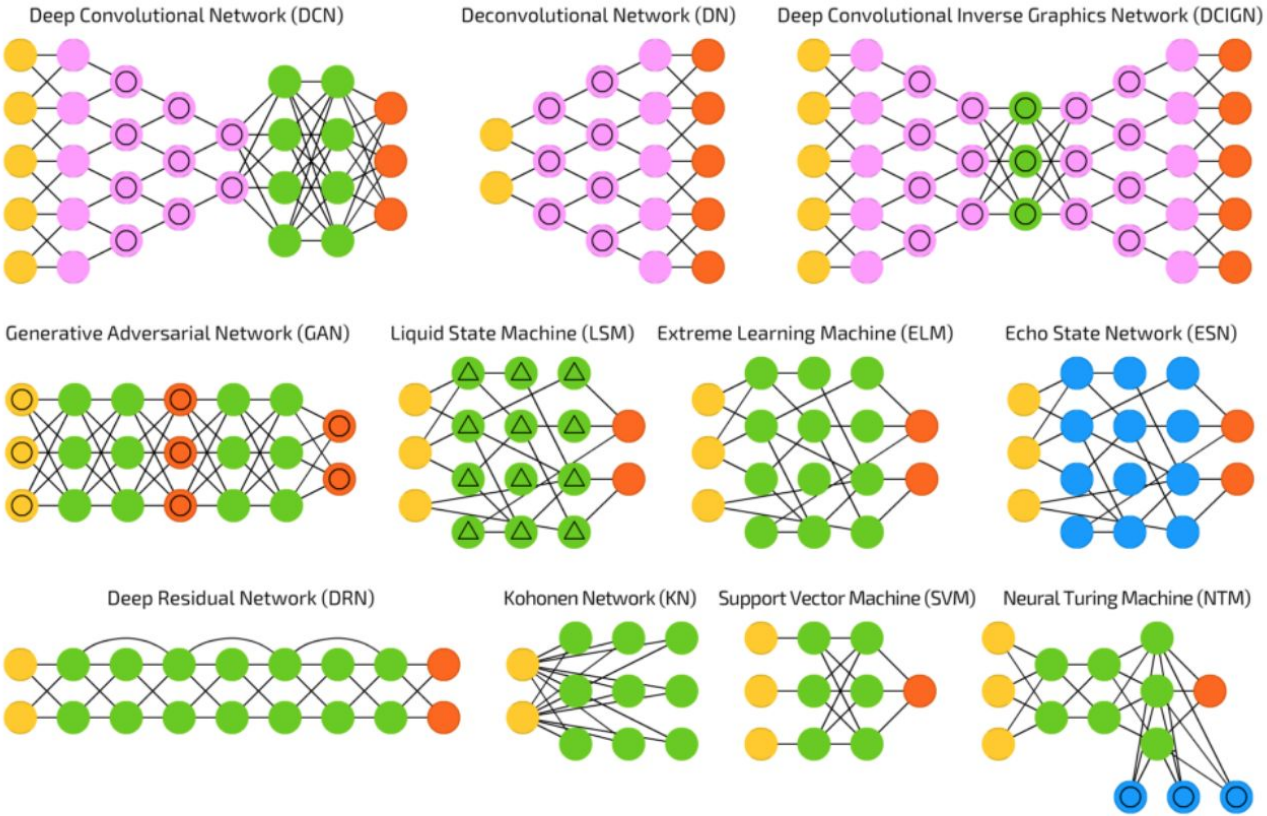
Different Types of Neural Networks

- Convolutional Neural Networks
- Recurrent Neural Networks
- Long Short Term Memory
- Generative Adversarial Neural Networks
- Autoencoders

Different Types of Neural Networks



Different Types of Neural Networks



Primary Tools for Deep Learning

- Tensorflow
- Torch/PyTorch
- Keras
- Caffe

Citations

- Andrew Ng, Deep Learning and Neural Networks