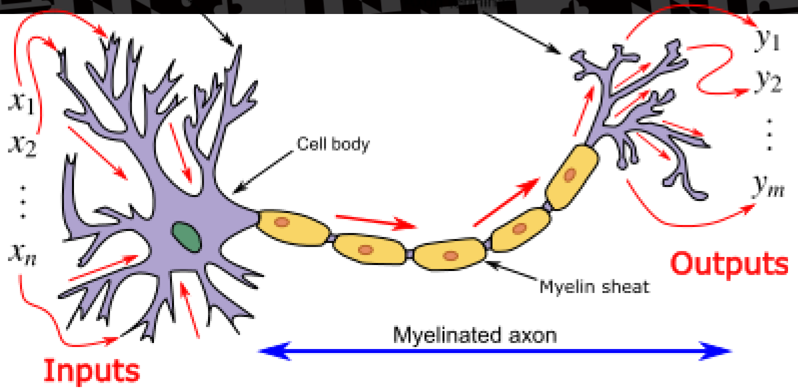
CMSC 471

ML: Brief Intro to Neural Network

How do animal brains work?



Neuron and myelinated axon, with signal flow from inputs at dendrites to outputs at axon terminals

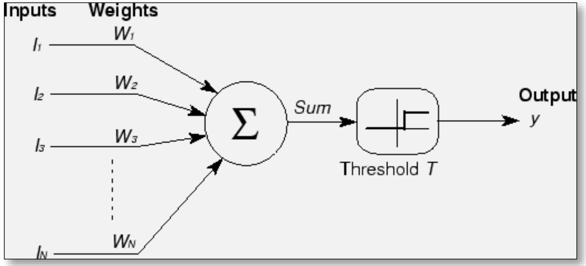
Neurons have body, axon and many dendrites

- In one of two states: firing and rest
- They fire if total incoming stimulus > threshold

Synapse: thin gap between axon of one neuron and dendrite of another

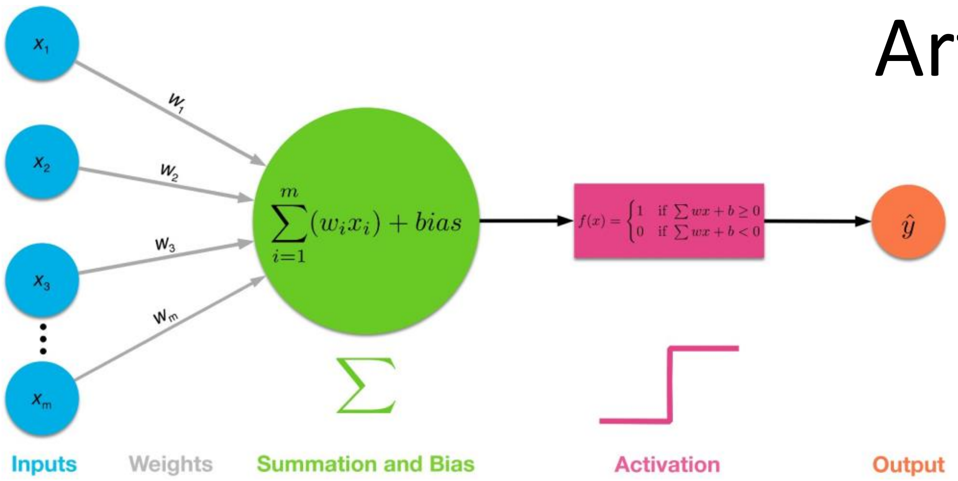
- Signal exchange

McCulloch & Pitts



- First mathematical model of biological neurons, **1943**
- All Boolean operations can be implemented by these neuron-like nodes
- Competitor to Von Neumann model for general purpose computing device
- Origin of automata theory

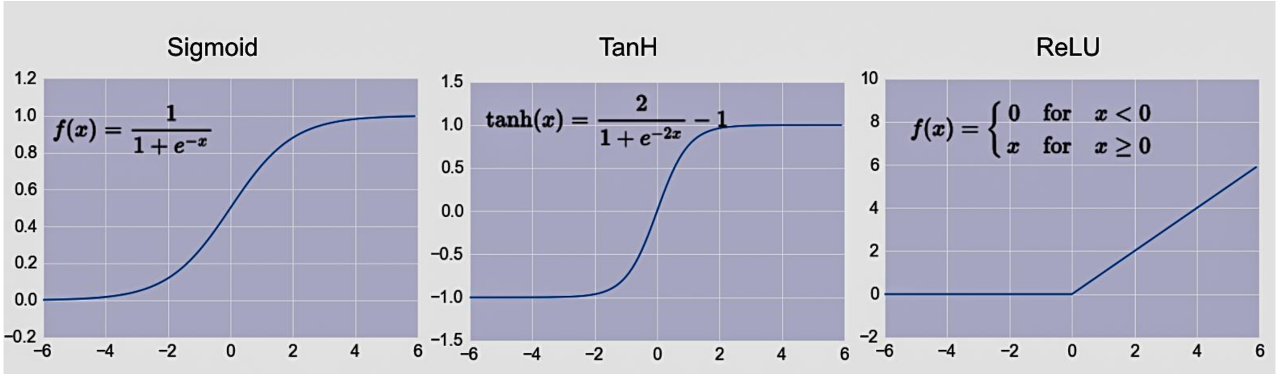
Artificial neural network



- Model still used today!
- Set of **nodes** with inputs and outputs
- Node performs computation via an **activation function**
- **Weighted connections** between nodes
- Connectivity gives network architecture
- NN computations depend on connections, weights, and activation function

Common Activation Functions

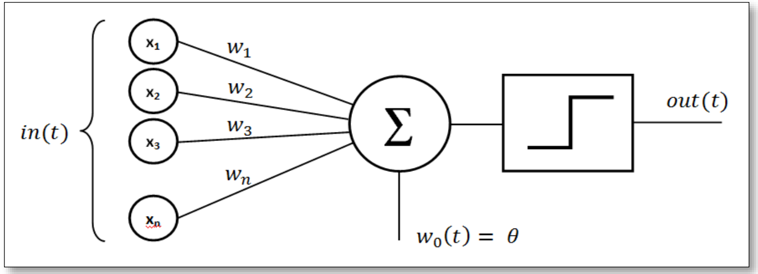
defines the output of that node given an input



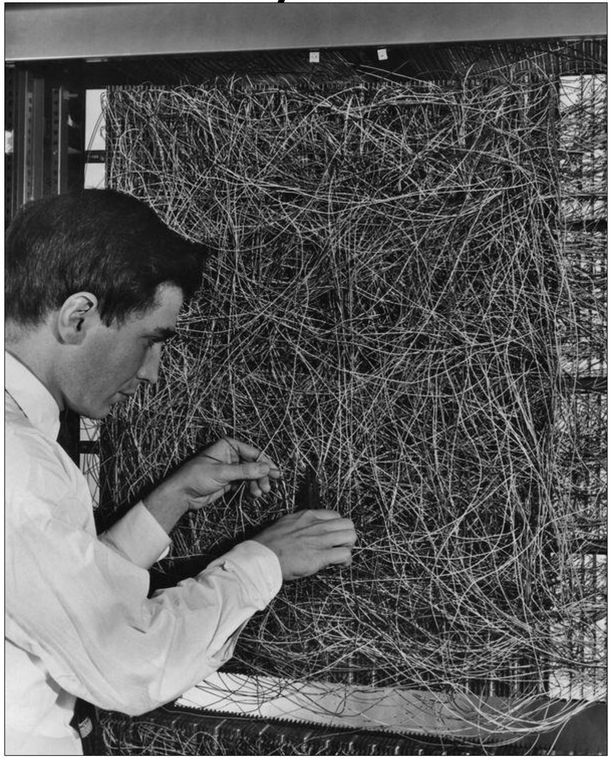
Choice of activation function depends on problem and available computational power

Rosenblatt's perceptron (1958-60)

- Single layer network of nodes
- Real valued weights +/-
- Supervised learning using a simple learning rule



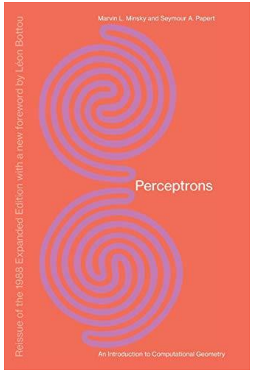
- Essentially a linear classifier
- Widrow & Hoff (1960-62) added better learning rule using gradient descent



Mark 1 perceptron computer, Cornell Aeronautical Lab, 1960

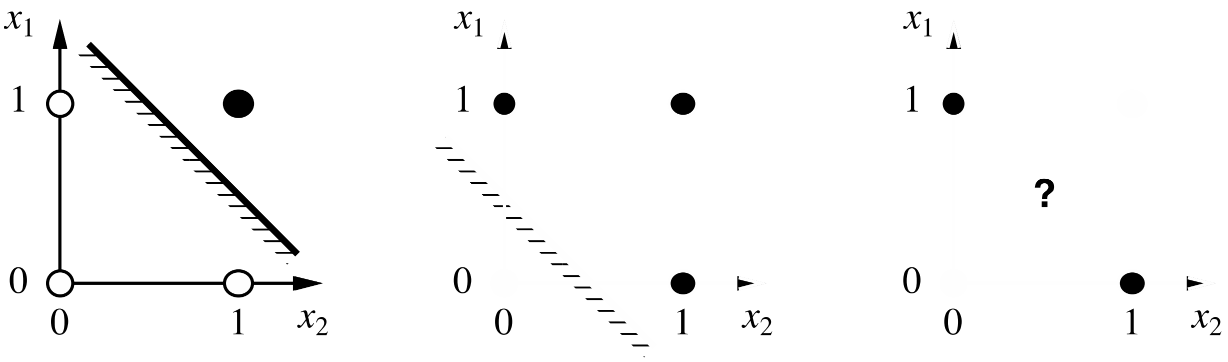
Problem with SLPs

- Perceptrons, Minsky and Papert, 1969
- Described serious problems with perceptron model
  - Single-layer perceptrons cannot represent (learn) simple functions that are not linearly separable, such as XOR
  - Multi-layers of non-linear units may have greater power but there is no learning rule for such nets
  - Scaling problem: connection weights may grow infinitely
  - First two problems overcame by latter effort in 80s, but scaling problem persists



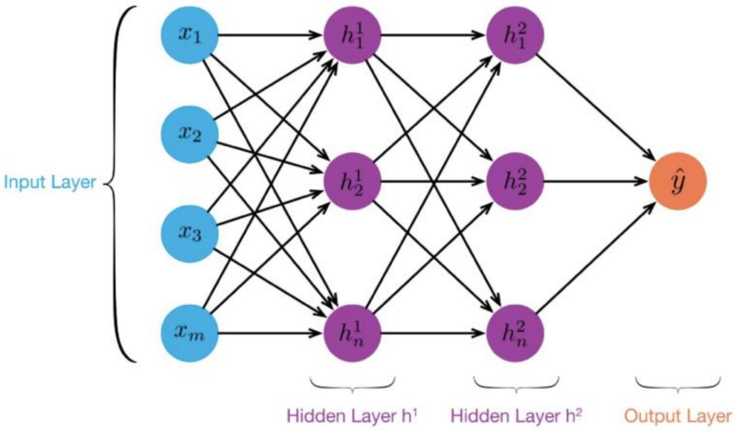
Not with a perceptron ☹

Consider Boolean operators (and, or, xor) with four possible inputs: 00 01 10 11



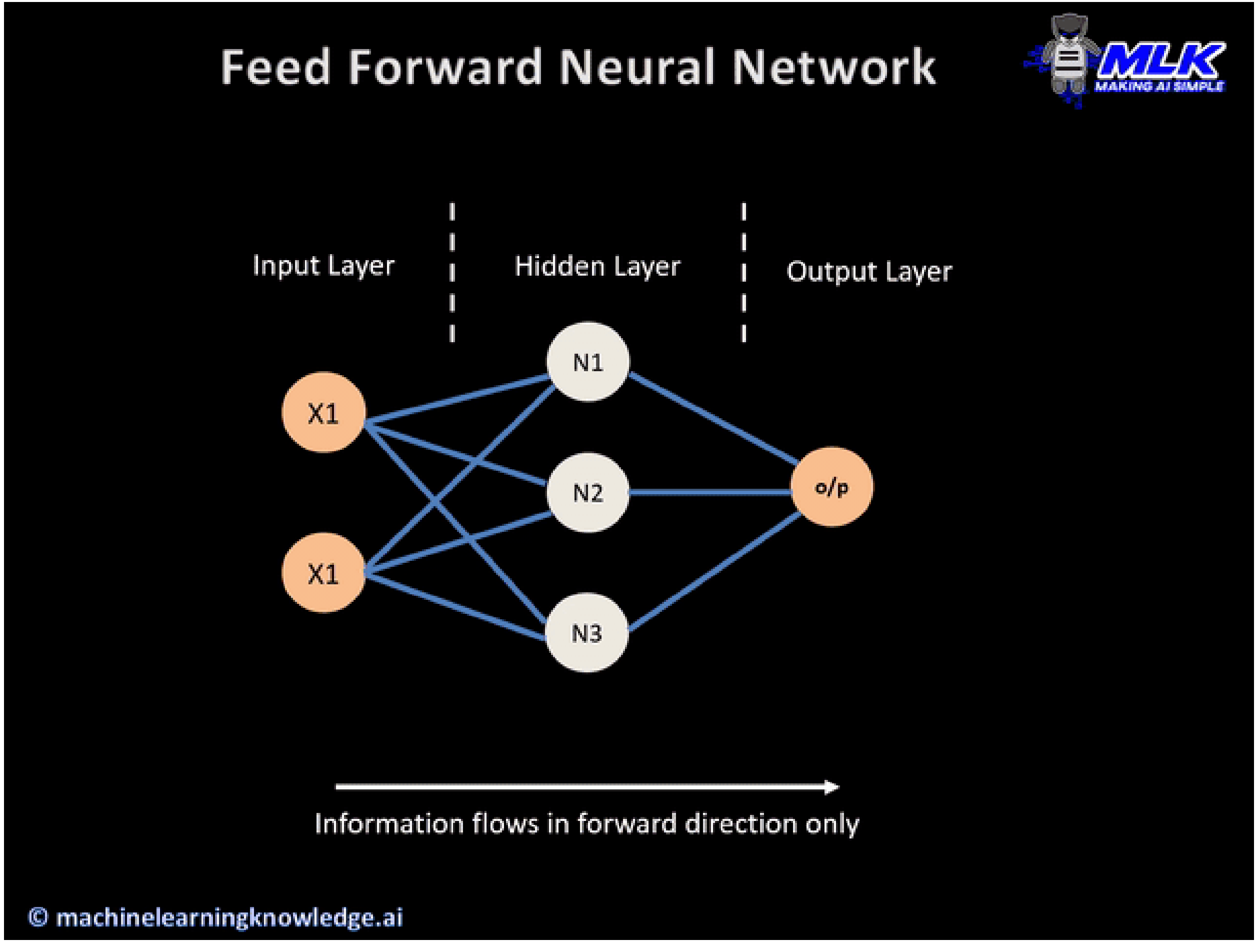
(a) x_1 **and** x_2 (b) x_1 **or** x_2 (c) x_1 **xor** x_2

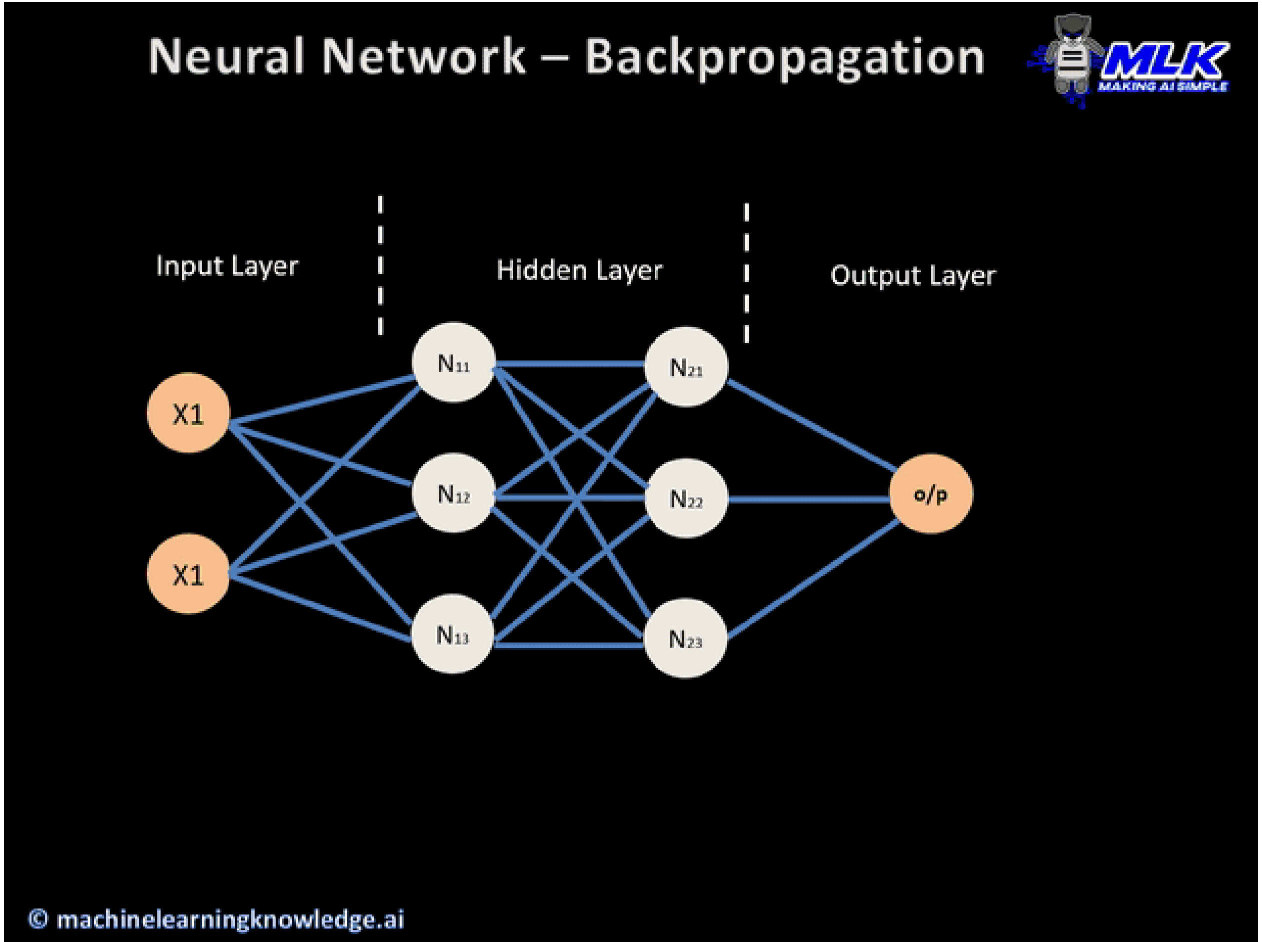
Training examples are not linearly separable for one case: *sum=1 iff x1 xor x2*



MLP: Multilayer Perceptron

- ≥ 1 "hidden layers" between inputs & output
- Can compute **non-linear functions**
- Training: adjust weights slightly to reduce error between output **y** and target value **t;** repeat
- Introduced in 1980s, still used today


Feed Forward Neural Network
MLK
MAKING AI SIMPLE
Input Layer
Hidden Layer
Output Layer
X1
X1
N1
N2
N3
o/p
Information flows in forward direction only
© machinelearningknowledge.ai



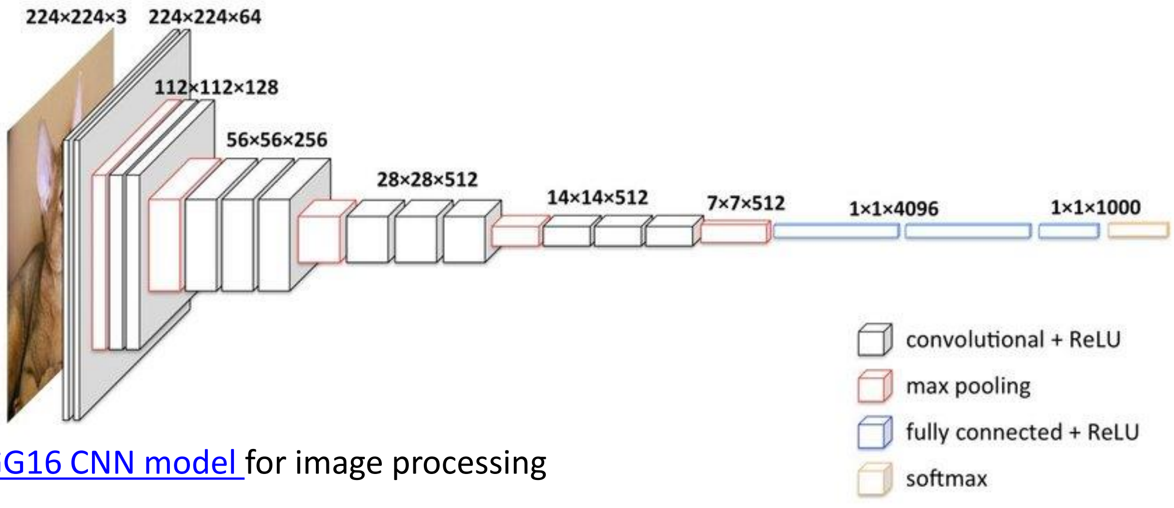
Neural Network – Backpropagation
MLK
MAKING AI SIMPLE
Input Layer
Hidden Layer
Output Layer
X1
X1
N11
N12
N13
N21
N22
N23
o/p
© machinelearningknowledge.ai

But problems remain…

- It’s often the case that solving a problem just reveals a new one that needs solving
- For a large MLPs, backpropagation takes forever to converge!
- Two issues:
  - Not enough compute power to train the model
  - Not enough labeled data to train the neural net
- SVMs dominate, since they converge to global optimum in O(n^2)

Deep Learning

- Deep learning refers to models going beyond simple feed-forward multi-level perceptron
  - Though it was used in a ML context as early as 1986
- "deep" refers to the models having many layers (e.g., 10-20) that do different things



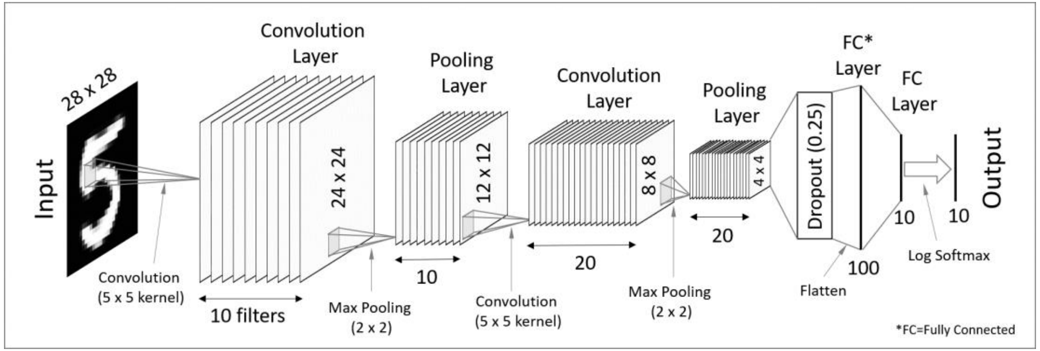
The VGG16 CNN model for image processing

Neural Network Architectures

Current focus on large networks with different “architectures” suited for different kinds of tasks

- Feedforward Neural Network
- CNN: Convolutional Neural Network
- RNN: Recurrent Neural Network
- LSTM: Long Short Term Memory
- GAN: Generative Adversarial Network
- Transformers

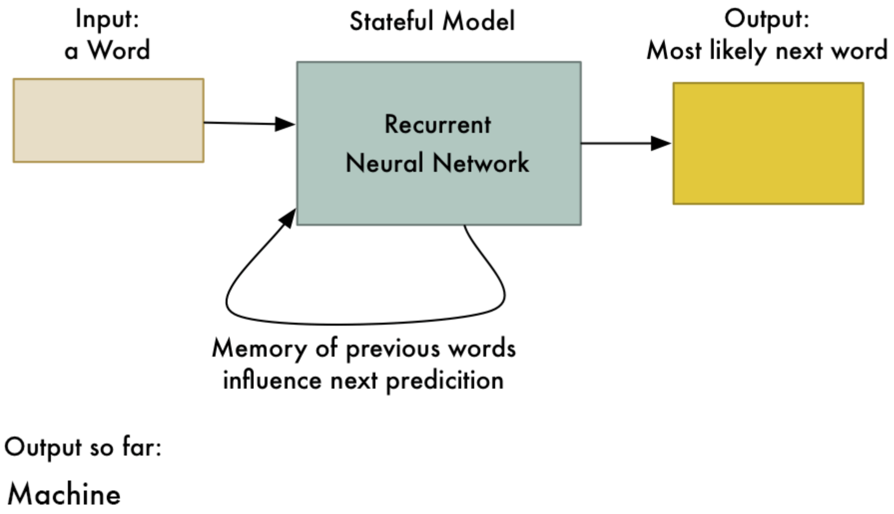
CNN: Convolutional Neural Network



- Good for image processing: classification, object recognition, automobile lane tracking, etc.
- Classic demo: learn to recognize hand-written digits from MNIST data with 70K examples

RNN: Recurrent Neural Networks

- Good for learning over sequences of data, e.g., a sentence orf words
- LSTM (Long Short Term Memory) a popular architecture



gif from Adam Geitgey

Keras

- “Deep learning for humans”
- Keras npw (v2.4) only supports TensorFLow
- Supports CNNs and RNNs and common utility layerslike dropout, batch normalization and pooling
- Coding neural networks used to be a LOT harder: Keras makes it easy and accessible!
- Documentation: https://keras.io/

Conclusions

- Quick intro to neural networks & deep learning
- Learn more by
  - Take UMBC's CMSC 478 machine learning class
  - Try scikit-learn's neural network models
  - Explore Keras as : https://keras.io/
  - Explore Google's Machine Learning Crash Course
  - Try Miner/Kasch tutorial on applied deep learning
  - Work through examples
- and then try your own project idea