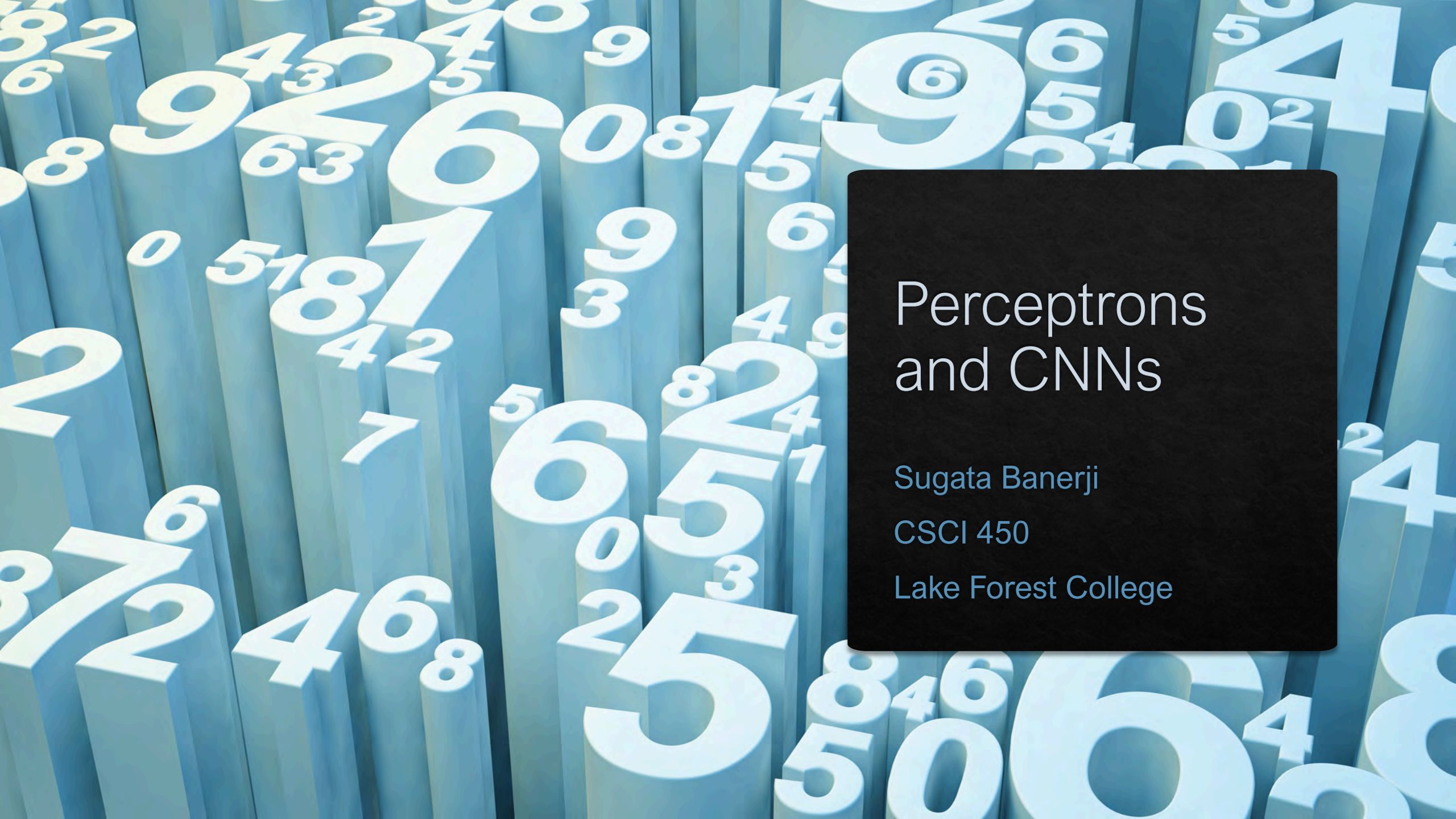


Perceptrons and CNNs

Sugata Banerji

CSCI 450

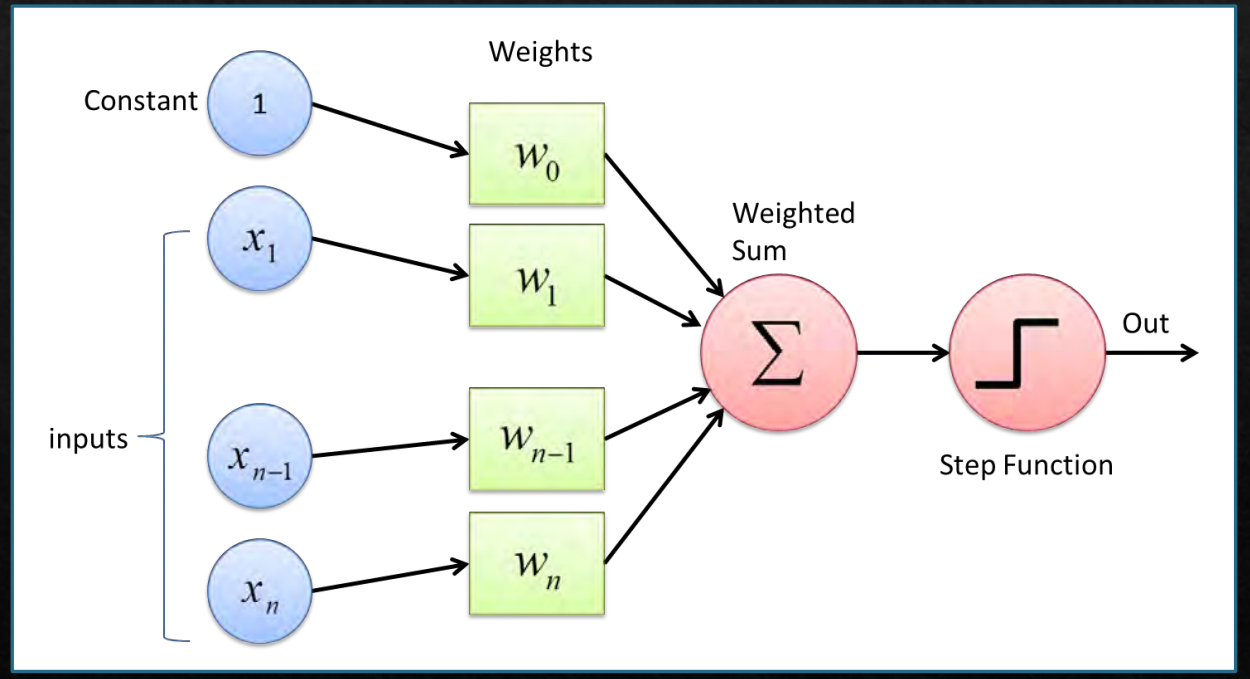
Lake Forest College

The Perceptron - History

- An algorithm for supervised learning of binary classifiers
- Invented in 1958 at the Cornell Aeronautical Laboratory by Frank Rosenblatt.
- Initially seemed promising, but it was quickly proved that they are only capable of learning linearly separable patterns.
  - This caused the field of neural network research to stagnate for many years
  - Later it was found that a multilayer network of perceptrons could solve many problems
- Neural network research experienced a resurgence in the 1980s.

The Perceptron - Structure

- The perceptron is a simplified model of a biological neuron.
- It takes an *n*-dimensional input
- Internally it has *n* weights and a *bias*
- It outputs *0* or *1*
- The weights and bias start as random and get proper values during training
- Training is done by the perceptron algorithm (feedforward)



The Perceptron Algorithm (Feedforward)

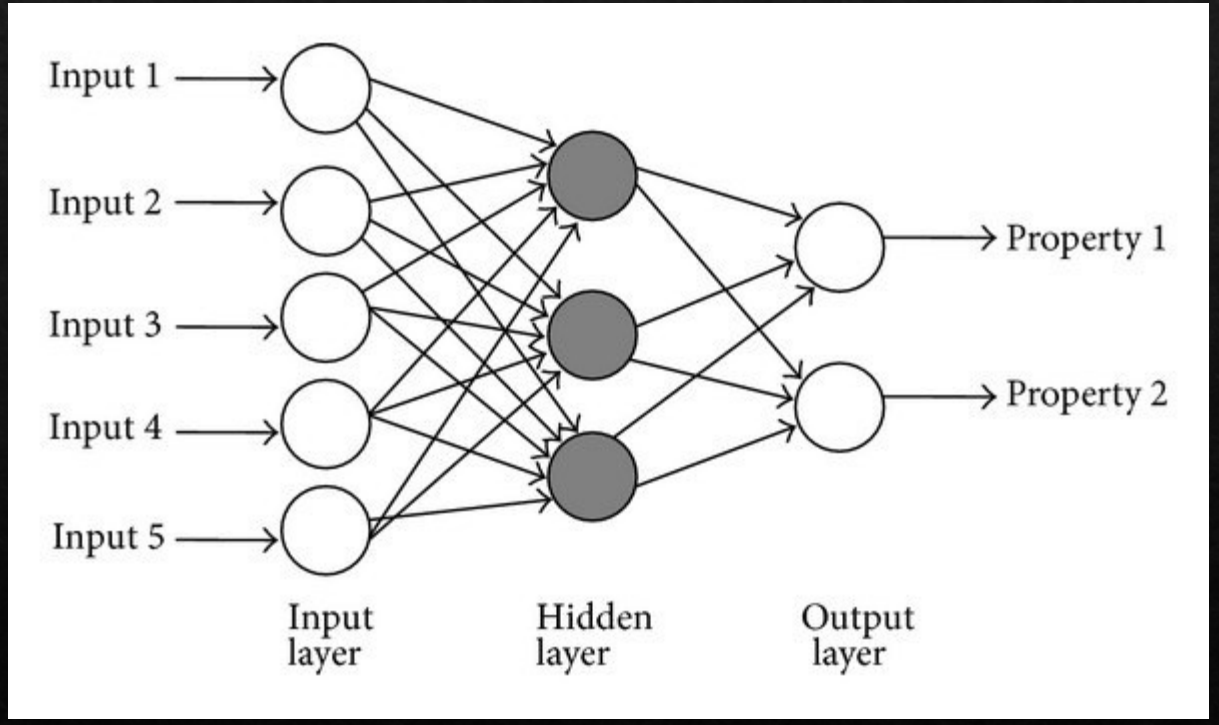
$$f(x) = \begin{cases} 1 & \text{if } w.x + b > 0 \\ 0 & \text{otherwise} \end{cases}$$

Where *w.x* = dot product of *w* and *x*, and *b* is the *bias*

- $w_i(t+1) = w_i(\mathrm{t}) + \mathrm{r}.\left[d_j - y_j(\mathrm{t})\right]x_{j,i}$
  - j signifies the j-th training sample
  - i signifies the i-th dimension
  - r is the learning rate
  - d is the desired output
  - y is the actual output
  - t is the iteration

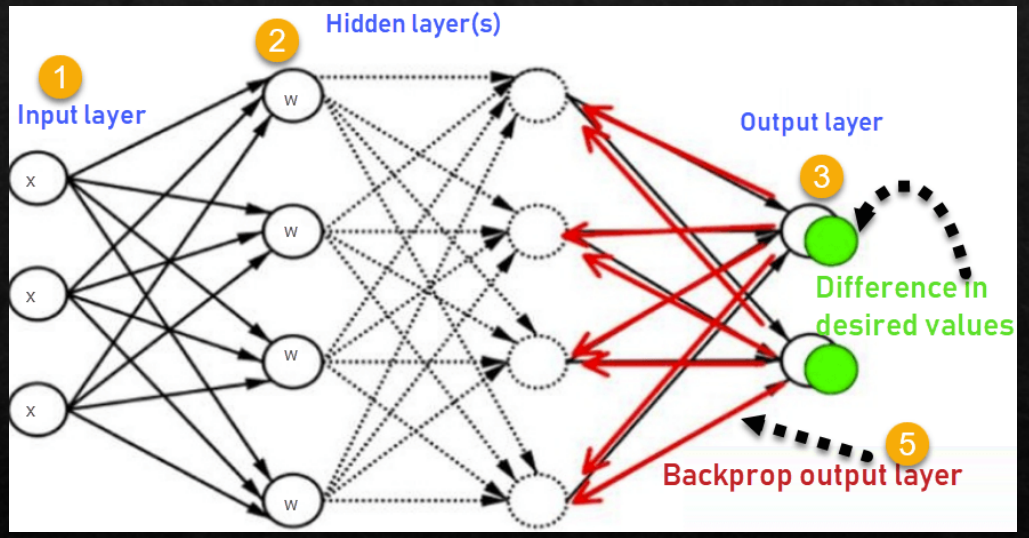
Multilayer Perceptrons

- Has more than one layer
- Outputs of one layer become inputs of the next
- Layers between input and output layers are called hidden layers
- Can detect more complex patterns
- May need *backpropagation* and sigmoid functions



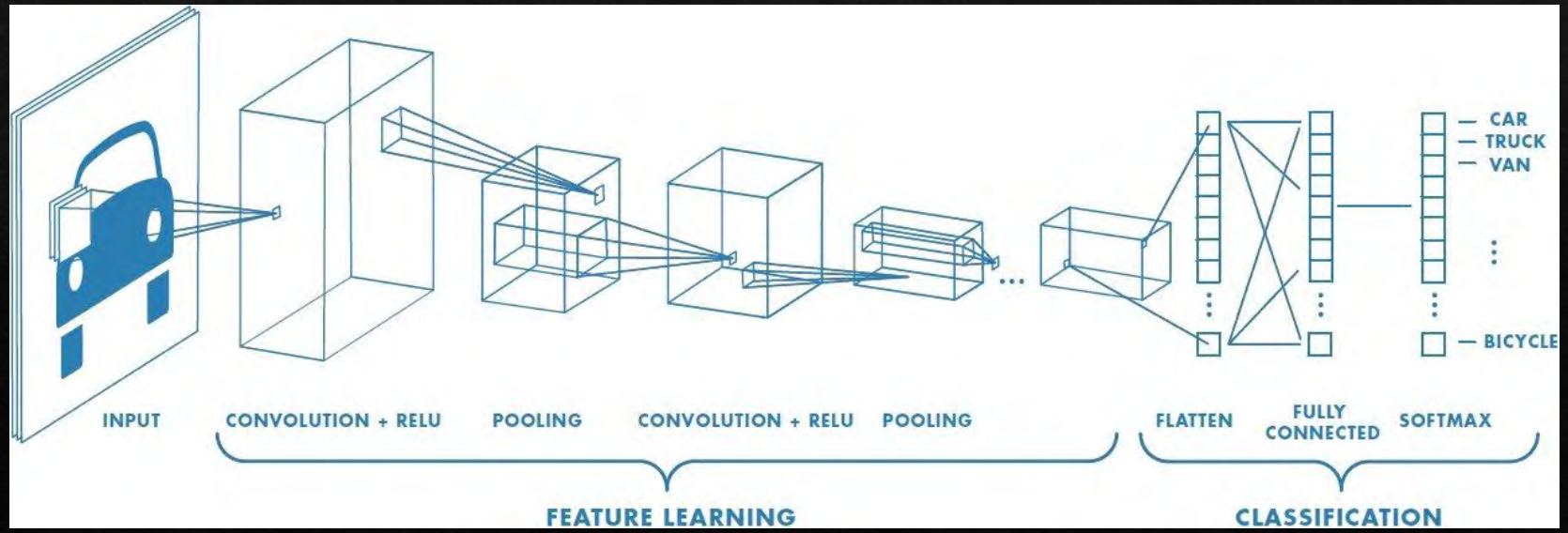
Backpropagation

- An algorithm to train multilayer neural networks (perceptrons)
- The error causes the weights to change backwards from the output to the input, hence the name
- The gradient of the error function with respect to the neural network's weights is calculated
- Fuch faster than forward-propagation
- Further Reading: https://brilliant.org/wiki/backpropagation/



Convolutional Neural Networks (CNN)

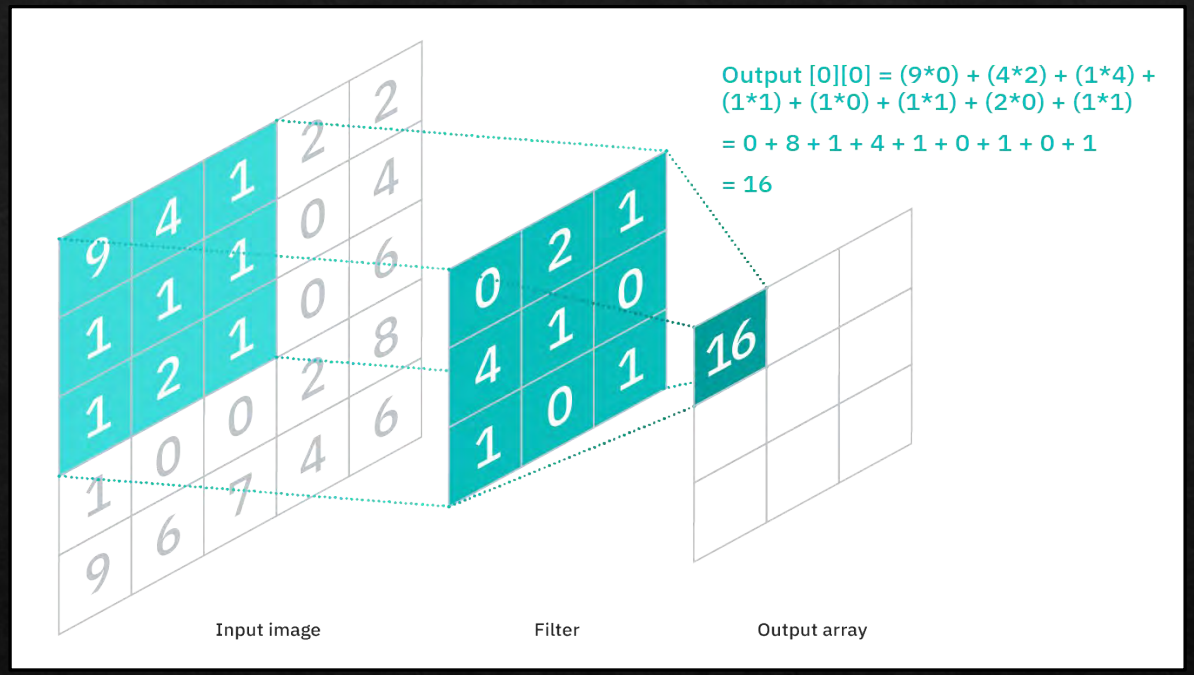
- A deep network (more then one hidden layer)
- Takes 2-D (or 3D) input
- Components
  - Convolution
  - Pooling
  - Dropout
  - ReLU
  - Fully Connected
  - Softmax
  - Sigmoid
  - Etc.



CNN Layers

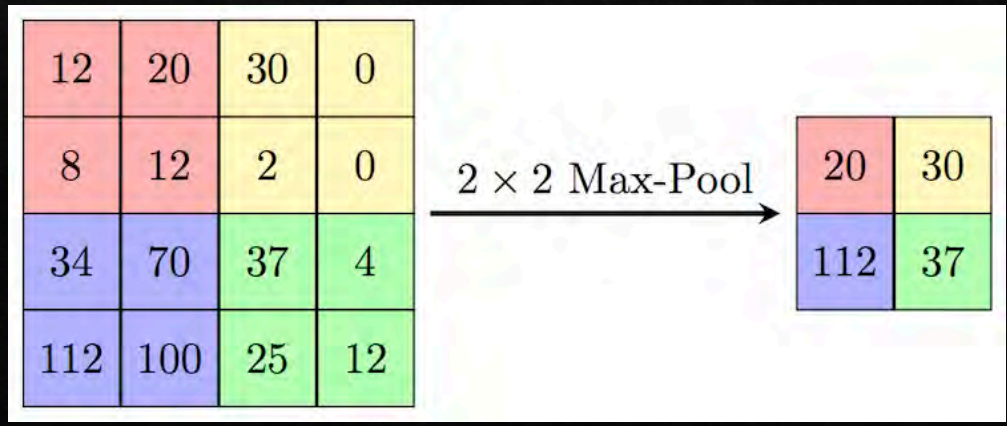
Convolution

- Filters to detect basic patterns in the image



Pooling

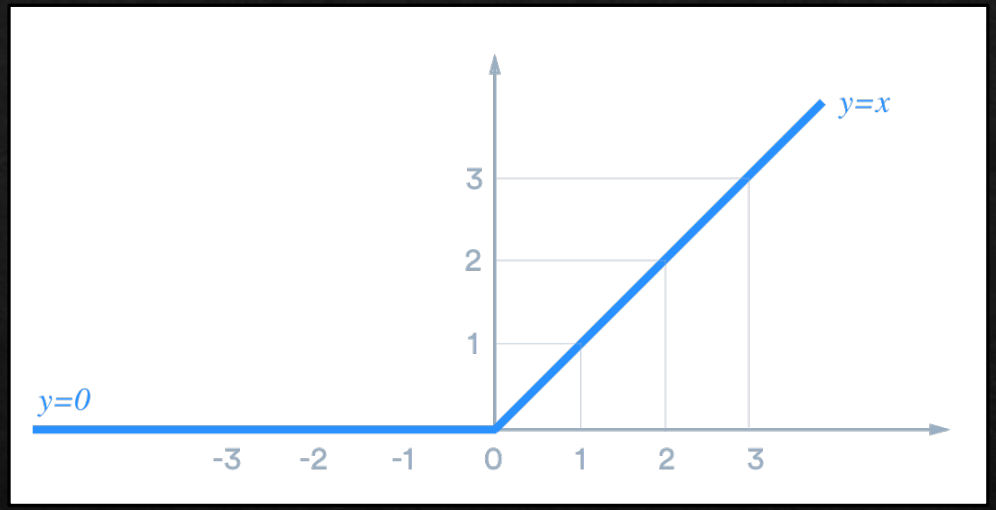
- Downscales the image while preserving approximate location of highest responses



CNN Layers

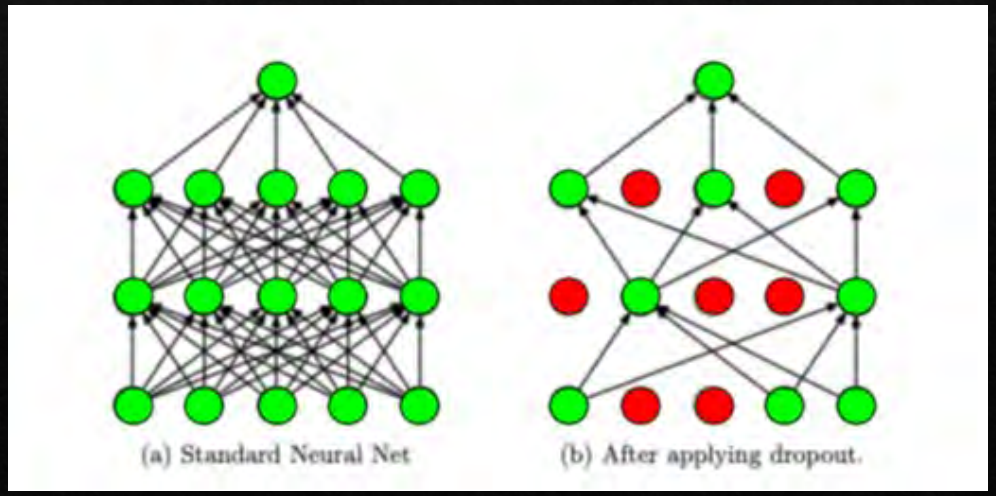
ReLU

- Rectified Linear Units or ReLU layers keep positive outputs intact while making negative outputs zero



Dropout

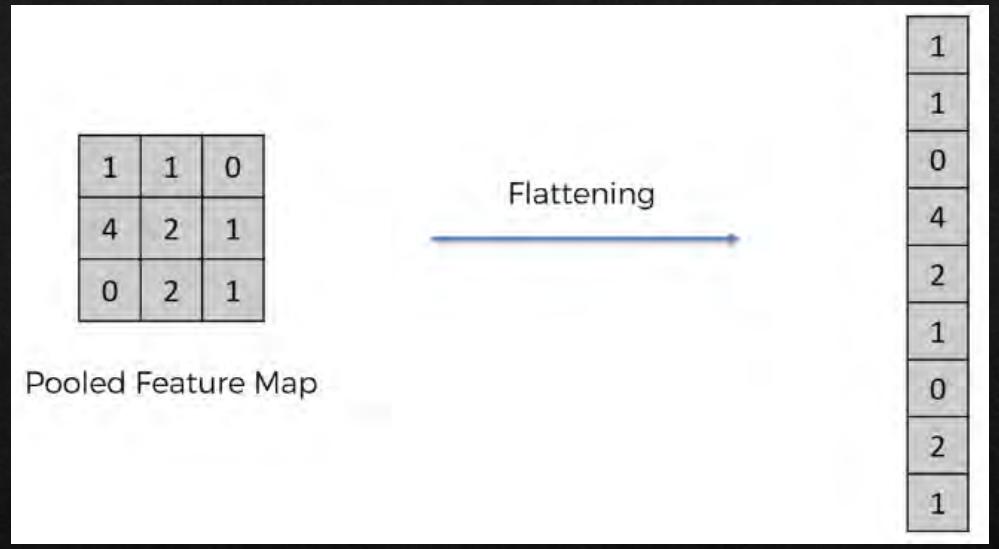
- Dropout layers “kill” some of the outputs from the previous layer to make the training more robust



CNN Layers

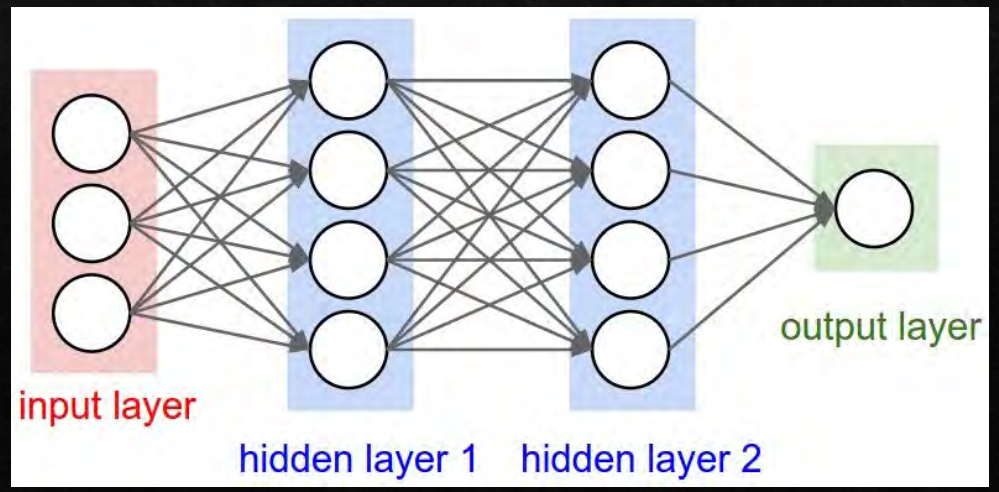
Flattening

- Changes the n-dimensional output from a layer into a 1-dimensional vector before the FC layers



Fully Connected

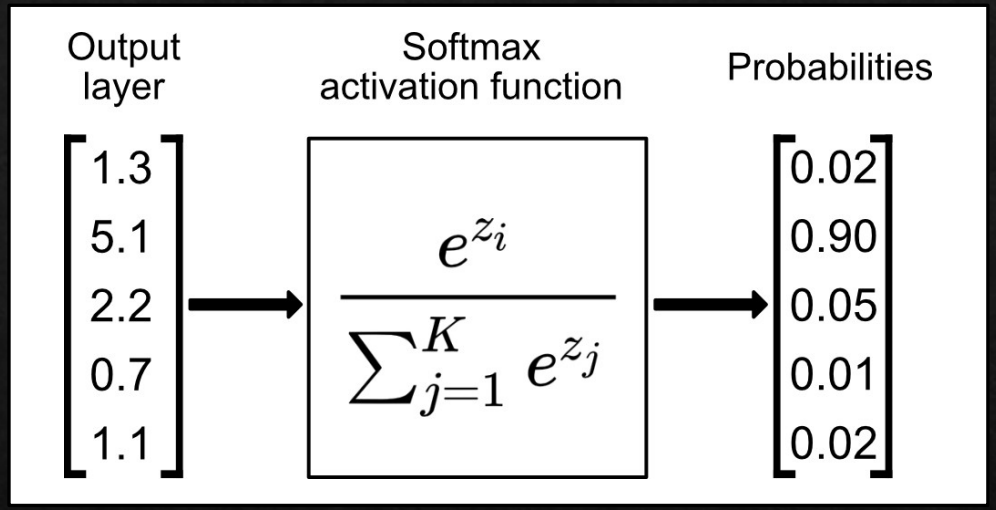
- A standard network of perceptrons all connected to each other that take 1-D inputs



CNN Layers

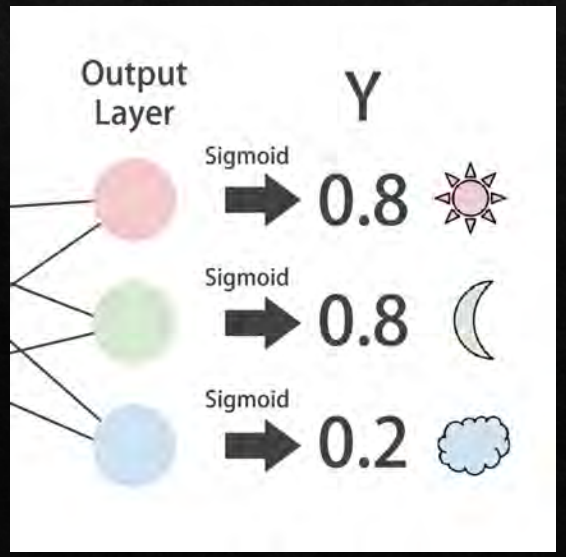
Softmax

- Last layer in single-label classification networks to make the output scores sum to 1



Sigmoid

- Used in output layer of multi-label classification networks where the scores don't sum to 1



First Layer filters

First Layer filters learnt by AlexNet by Krizhevsky et al (https://papers.nips.cc/paper/2012/file/c399862d3b9d6b76c8436e924a68c45b-Paper.pdf)