DePaul University
College of Computing and Digital Media

Casey Bennett, PhD

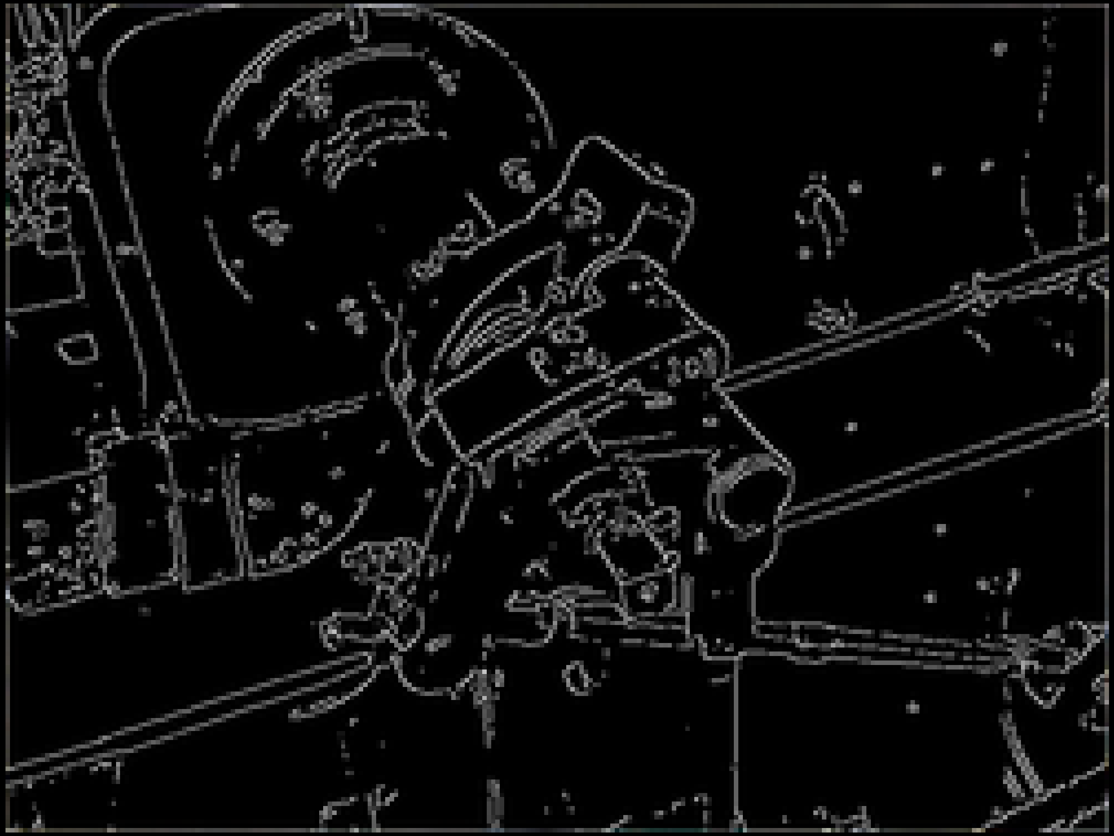
Feb. 4, 2019

This Week

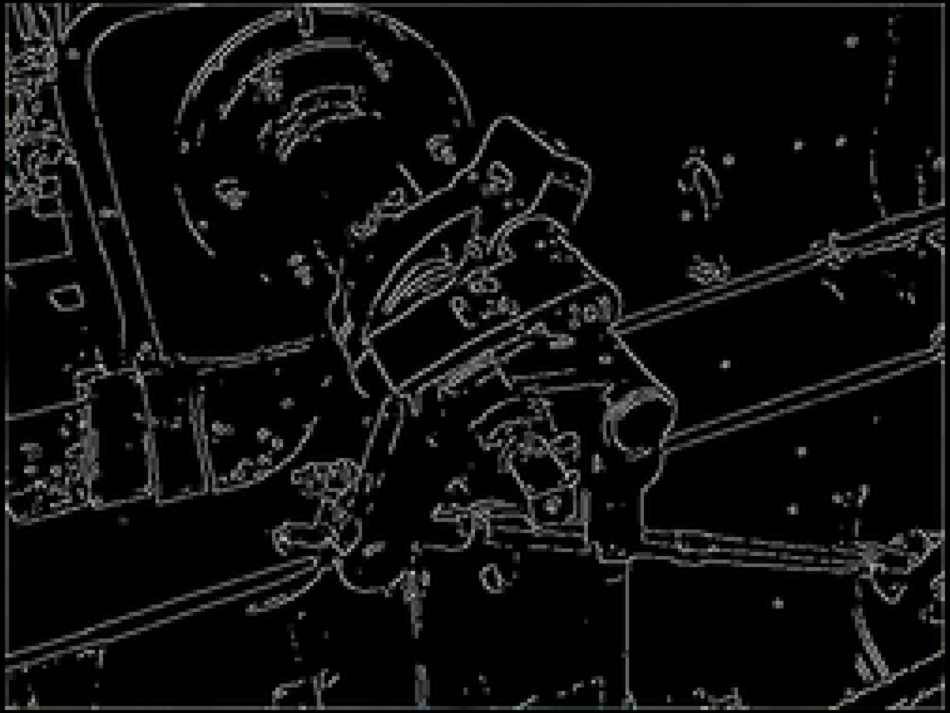
Neural Networks and Deep Learning

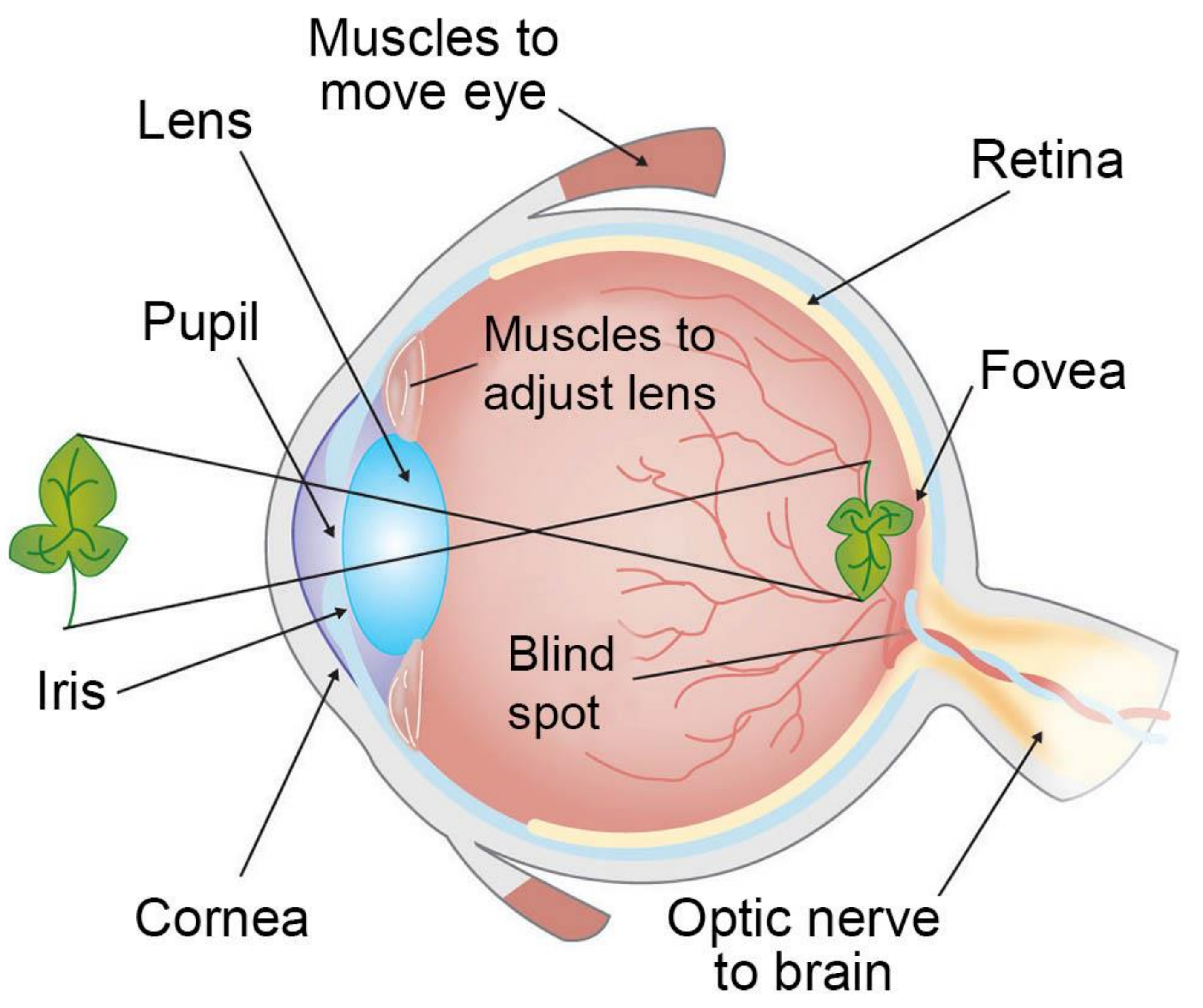
https://pollev.com/caseybennett801

or text “caseybennett801” to 37607

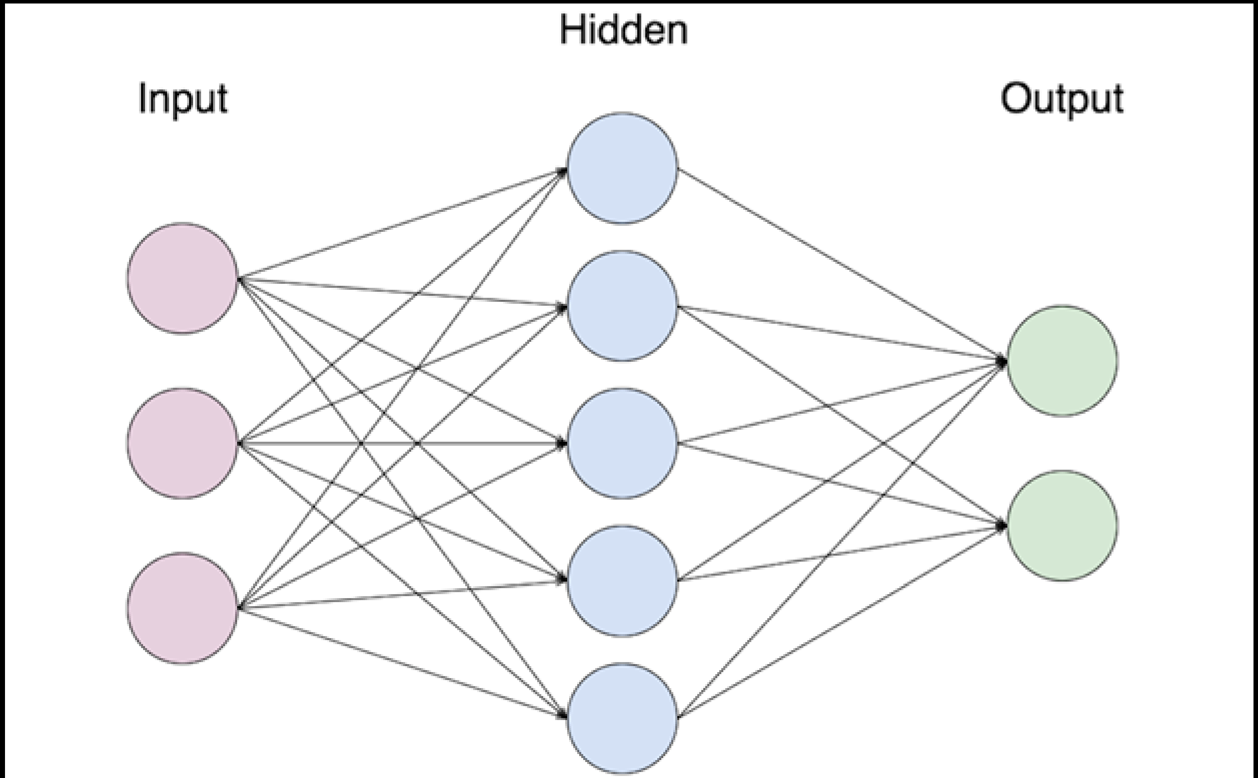


How could we go from seeing a line or a dot to understanding a whole scene?



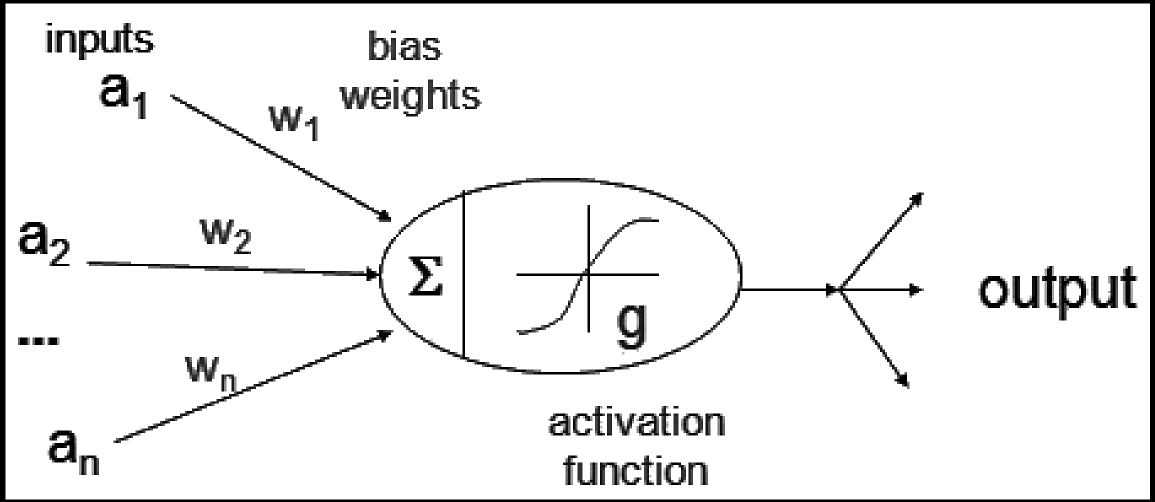

Muscles to
move eye
Lens
Retina
Pupil
Muscles to
adjust lens
Fovea
Iris
Blind
spot
Cornea
Optic nerve
to brain

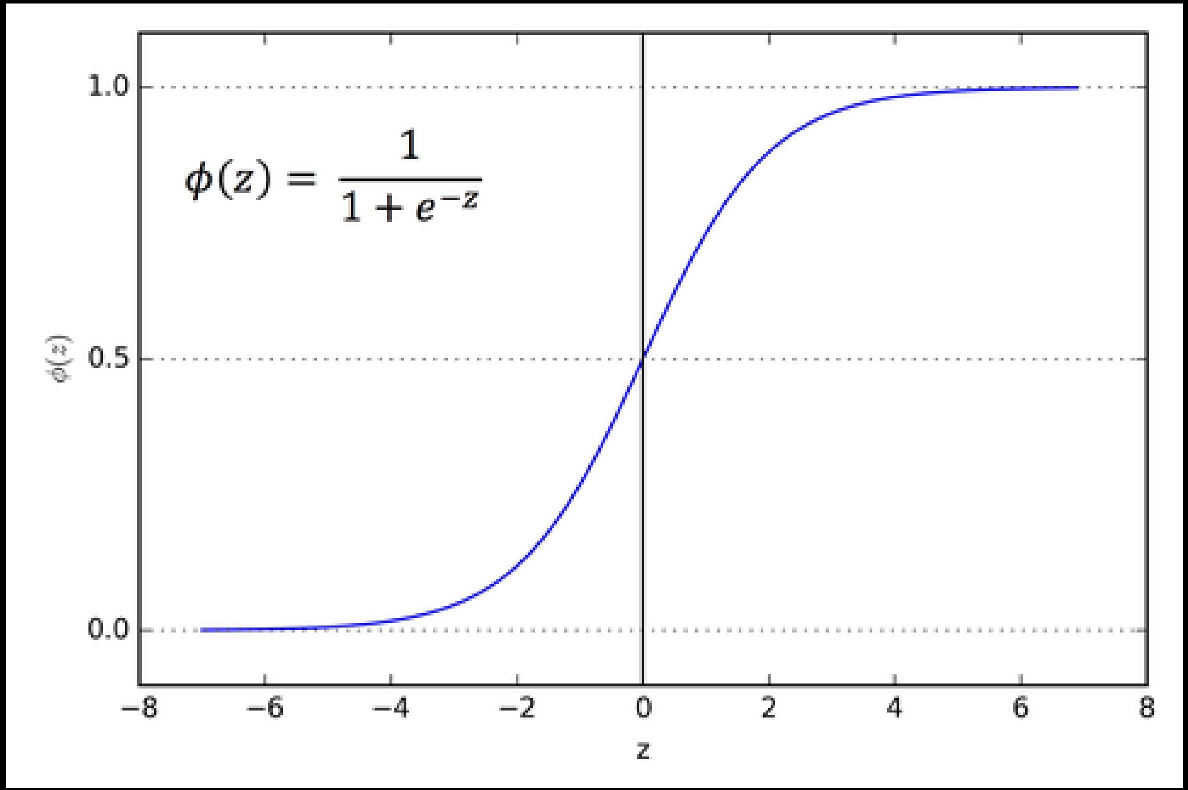
How could we make those neurons learn new things?

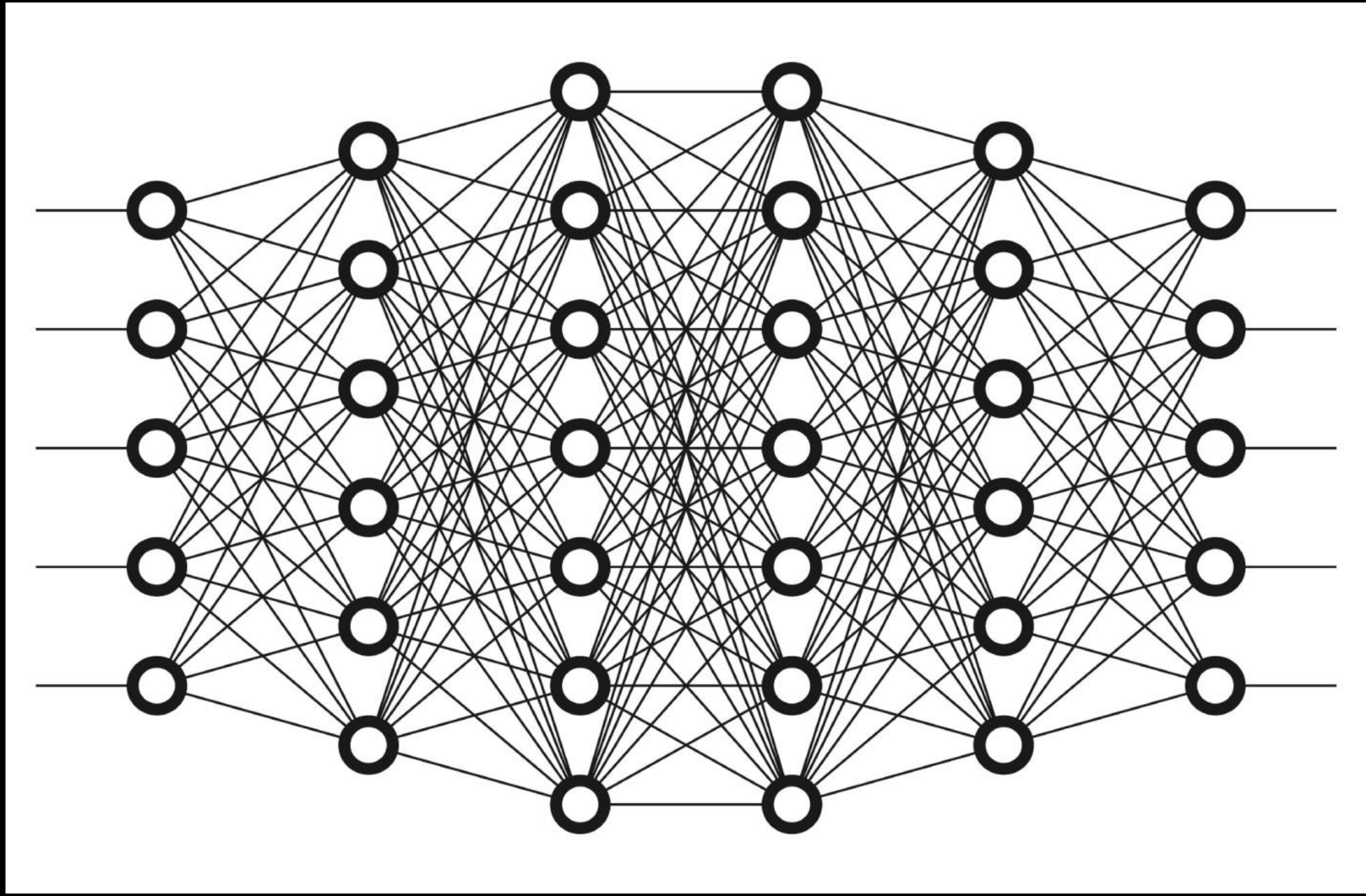


- We can connect these nodes together into *layers*, and teach them to do things by showing them their own errors to adjust connection *weights*
- Neural Network

Activation Function

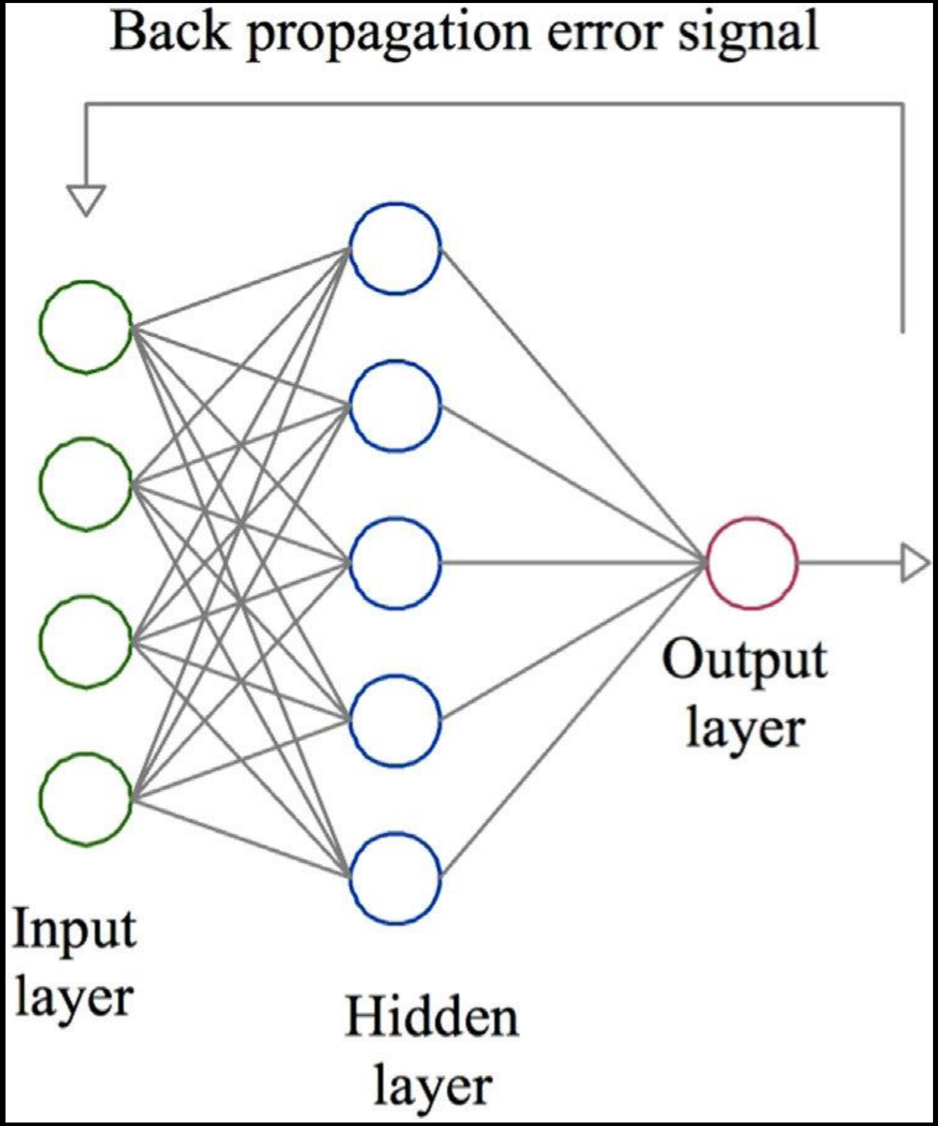




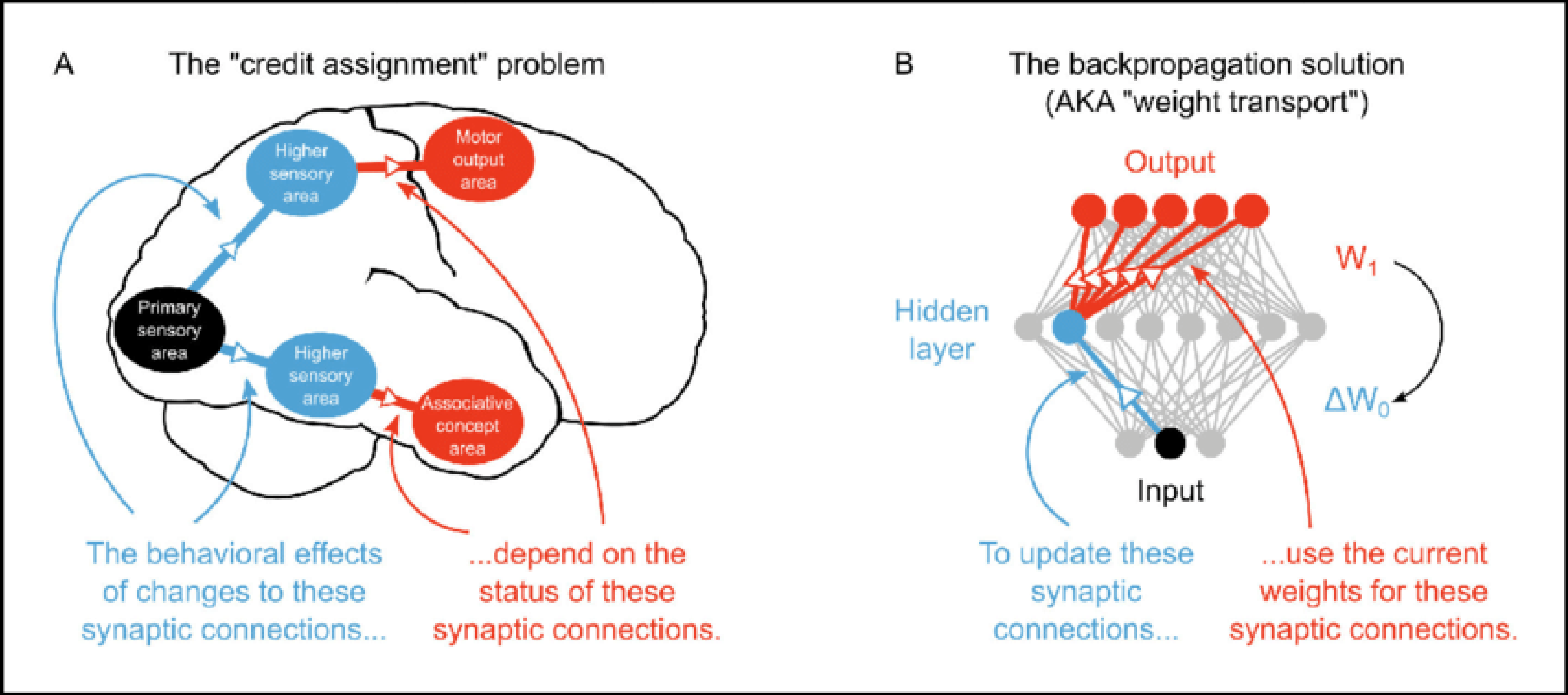


- Many hidden layers called Deep Learning
- Tends to be used for pattern recognition problems, such as computer vision or speech recognition

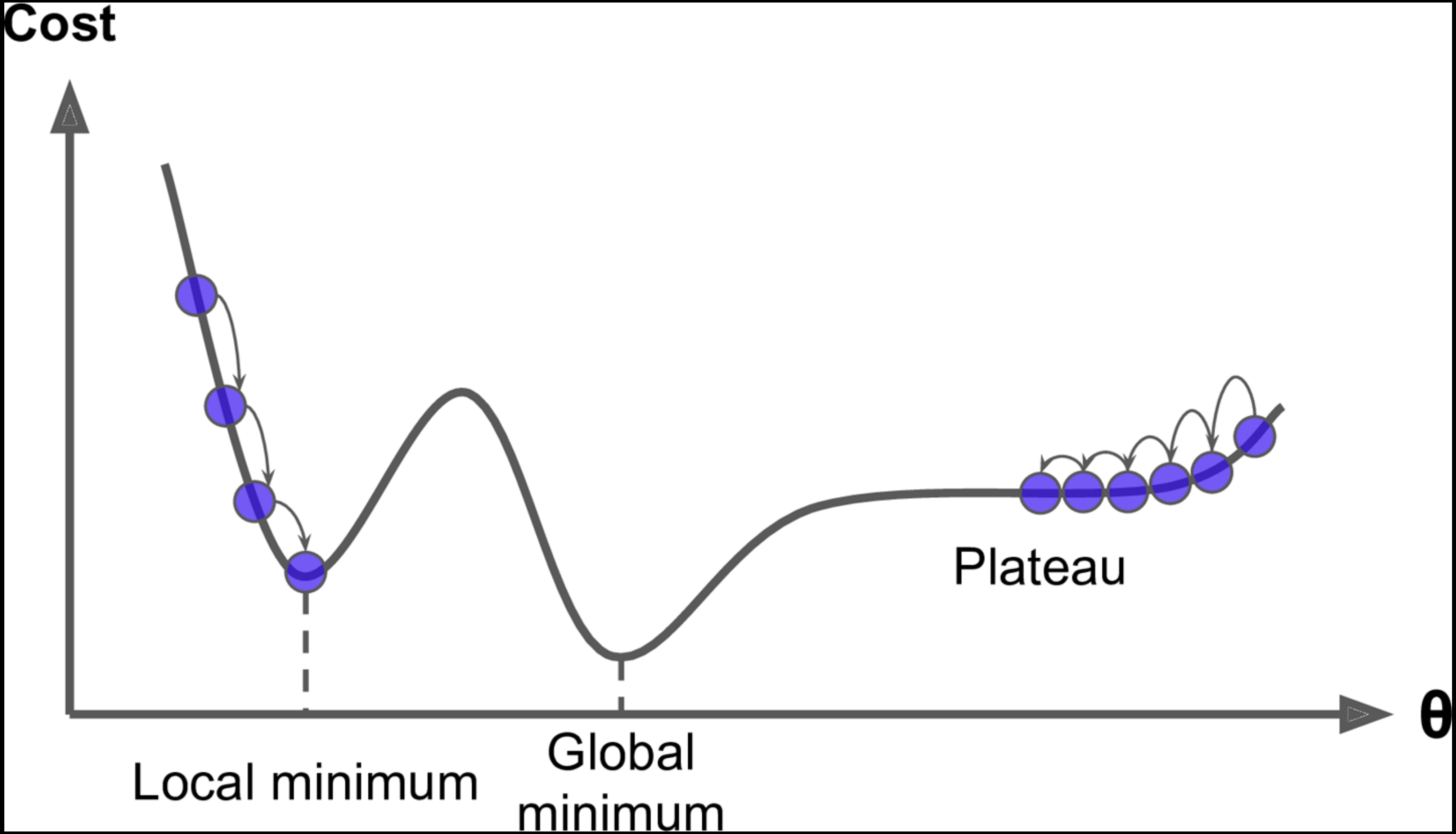
Backpropagation



Credit Assignment Problem



Training the Model: Gradient Descent


Cost
θ
Local minimum
Global
minimum
Plateau

How do we make sure we don't just find a local optima?

Gradient Descent

$$\mathbf{a}_{n+1} = \mathbf{a}_n - \gamma \nabla F(\mathbf{a}_n)$$

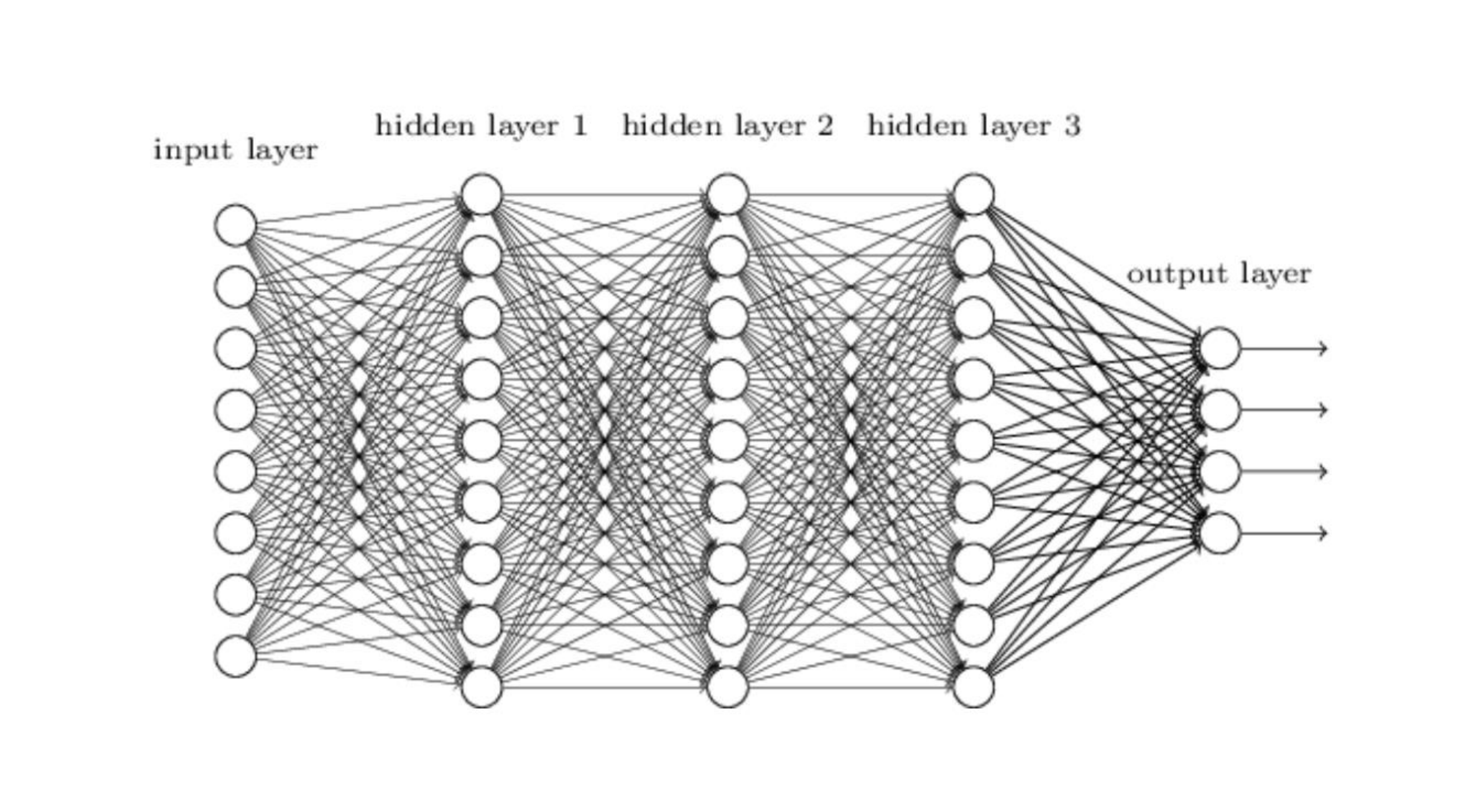
- Simply says if we have a point, “a”, and a Function “F”, we want to find the gradient and move in the negative direction so we can find the lowest point

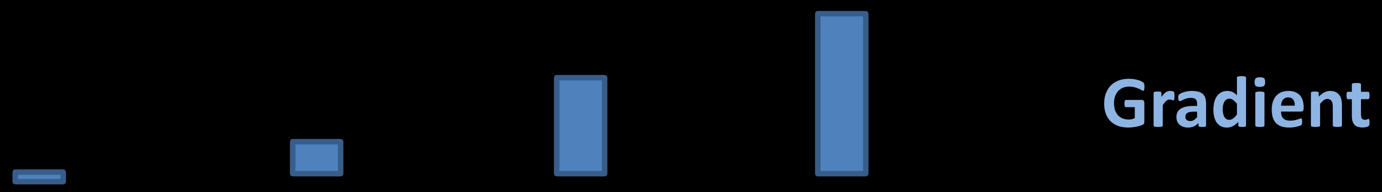
- Gamma here is the learning rate or step size

GD Psuedocode

```
# Find local minima of f(x) = x^4 + 3x^3 + 2
cur_x = 6                                   # The algorithm starts at x=6
gamma = 0.01                                # step size multiplier
precision = 0.00001
previous_step_size = 1
max_iters = 10000                           # maximum number of iterations
iters = 0                                   # iteration counter

df = lambda x: 4 * x**3 - 9 * x**2

while previous_step_size > precision and iters < max_iters:
    prev_x = cur_x
    cur_x -= gamma * df(prev_x)
    previous_step_size = abs(cur_x - prev_x)
    iters+=1

print("The local minimum occurs at", cur_x)
#The output for the above will be: ('The local minimum occurs at', 2.2499646074278457)
```

Vanishing Gradient Problem





Code Implementation

#SciKit Neural Network

```
MLPClassifier(solver='adam', hidden_layer_sizes=(10), activation='relu', alpha=0.0001,
              batch_size='auto', learning_rate='constant', learning_rate_init=0.001, power_t=0.5,
              max_iter=200, shuffle=True, random_state=None, tol=0.0001,momentum=0.9,
              early_stopping=False, validation_fraction=0.1, beta_1=0.9, beta_2=0.999,
              n_iter_no_change=10)
```

#Spark Neural Network

```
layers = [init_lyr_size, 10, bin_cnt]      #First layer must be set to Feature cnt final layer to target bins
clf = MultilayerPerceptronClassifier(labelCol="idxLabel", featuresCol="idxFeatures",
                                     maxIter=100, layers=layers, blockSize=128, seed=1234,
                                     tol=1e-06, stepSize=0.03, solver='l-bfgs')
```

- There are also *Regressor* versions for Neural Network in both Scikit and Spark, for when you have a continuous categorical or binary target variable you are trying to predict

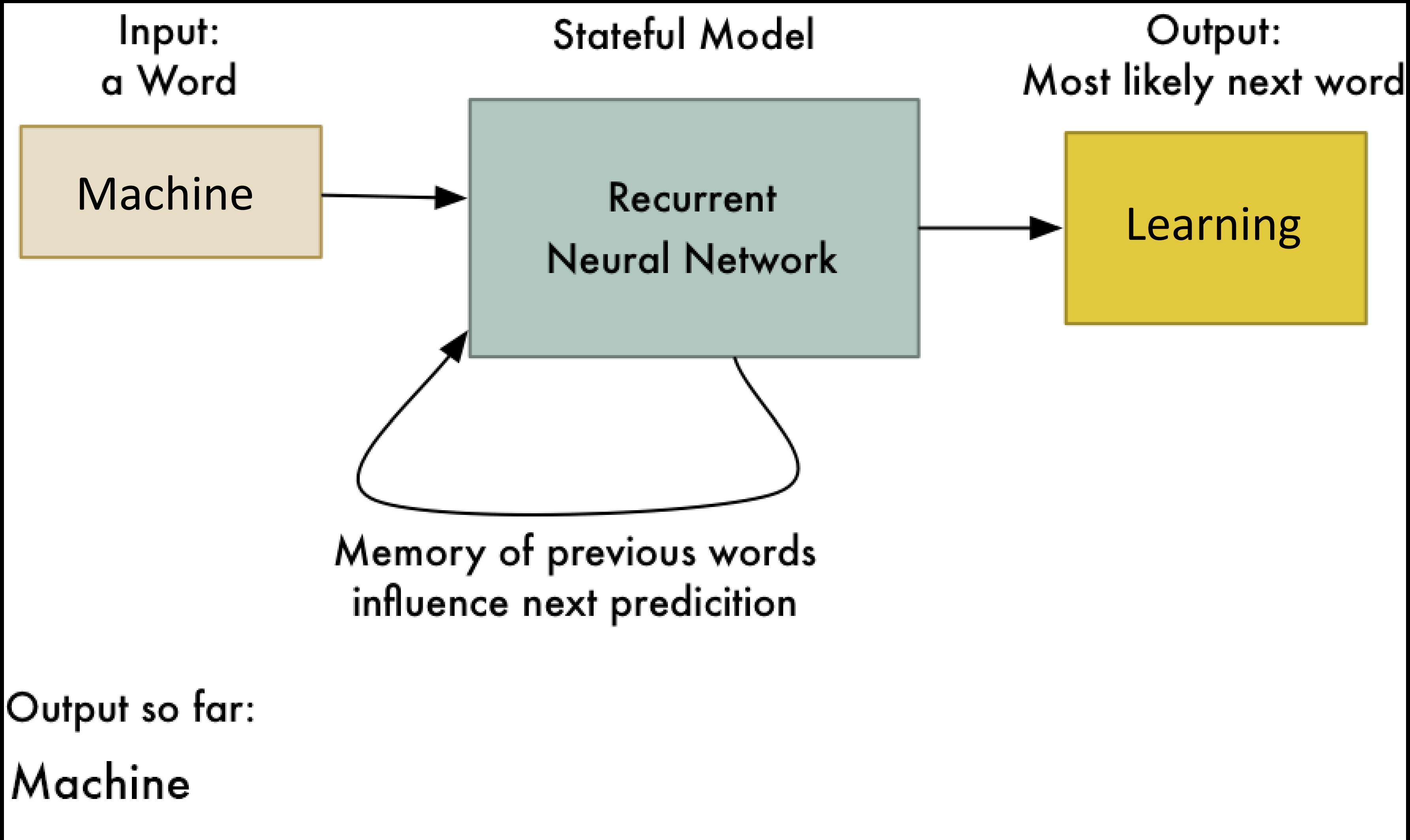
Code Implementation

- Pay careful attention to a few parameters:
  - Number of iterations (max iter)
  - Learning rate (aka step size)
  - Activation function
  - Hidden Layer setup

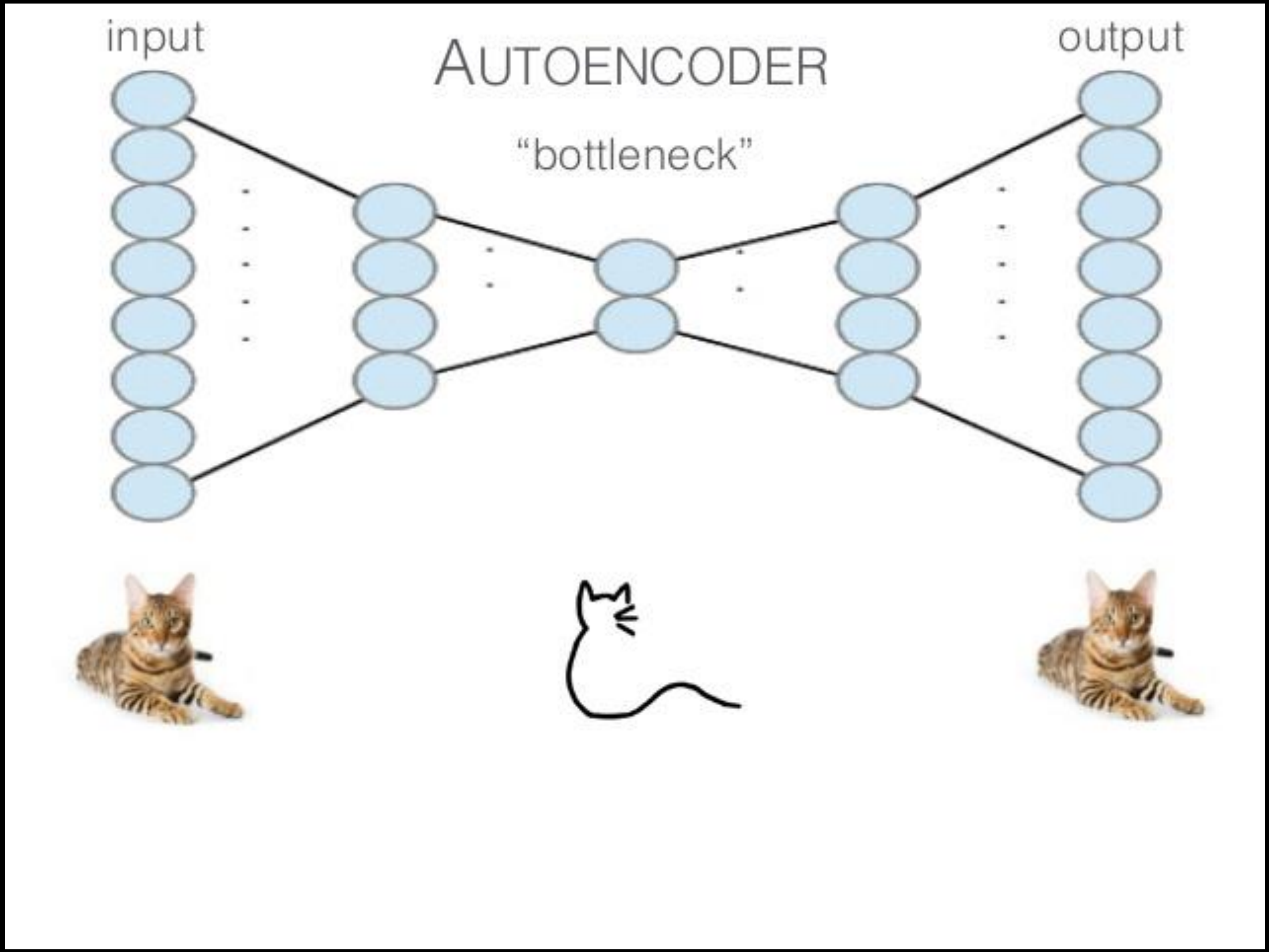
Neural Networks come in many colors

The problem with learning new things, is that we sometimes forget old things. How can we prevent that?

Recurrent Neural Networks and LTSMs



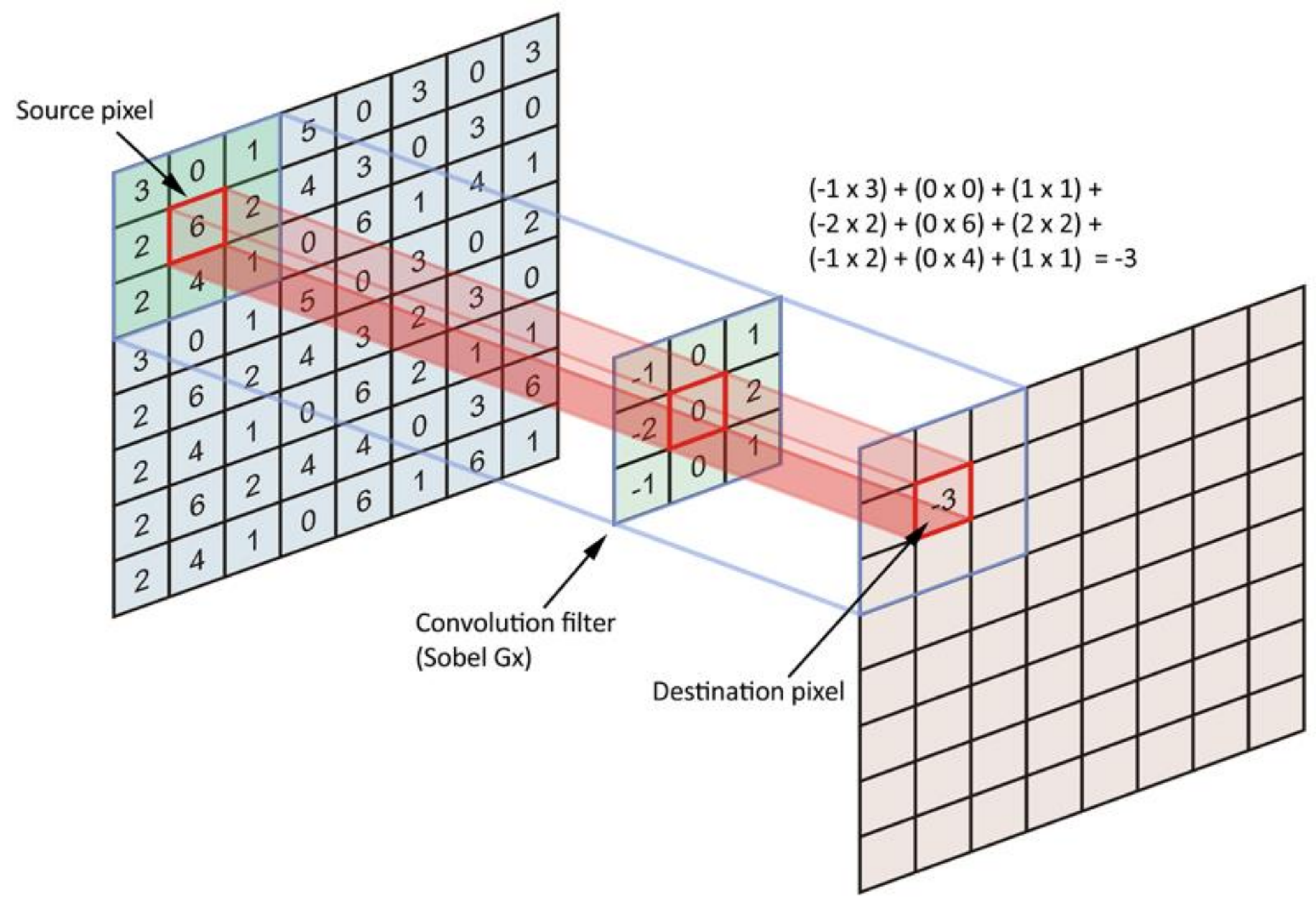
RBMs and Autoencoders

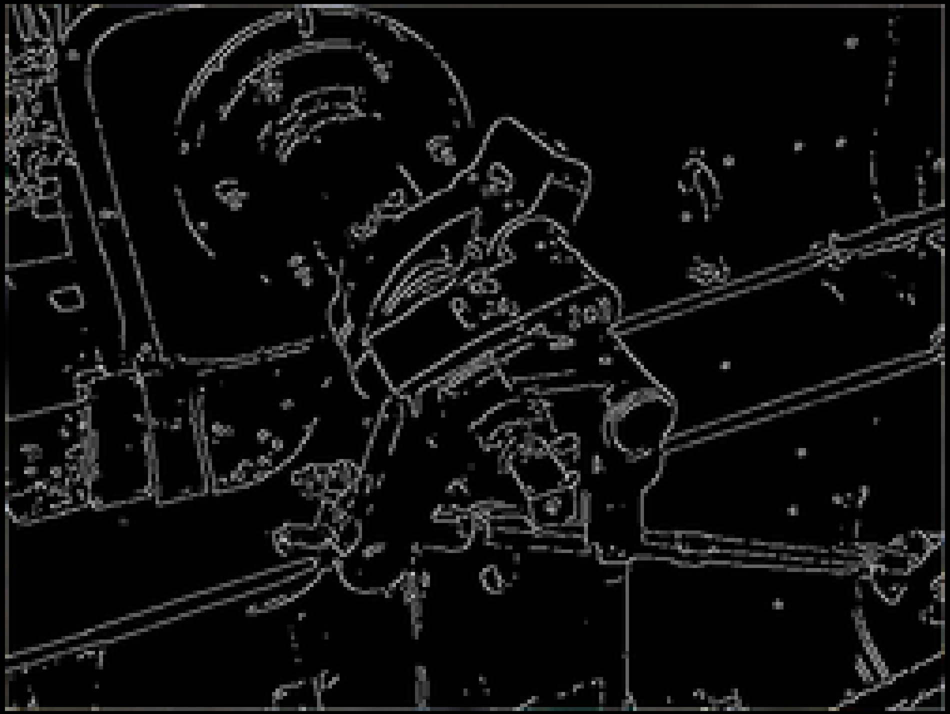


- RBMs and Autoencoders often used for *compression* … to create more relevant info
- RBMs and Autoencoders are useful when there is *ambiguity* in the features
- RBMs use stochastic activation, gibbs sampling
- You can kind of think of the process of training an RBM like a variant of the EM algorithm

How does the camera on your phone sharpen or blur images?

Convolution Filter





Convolutional Neural Network

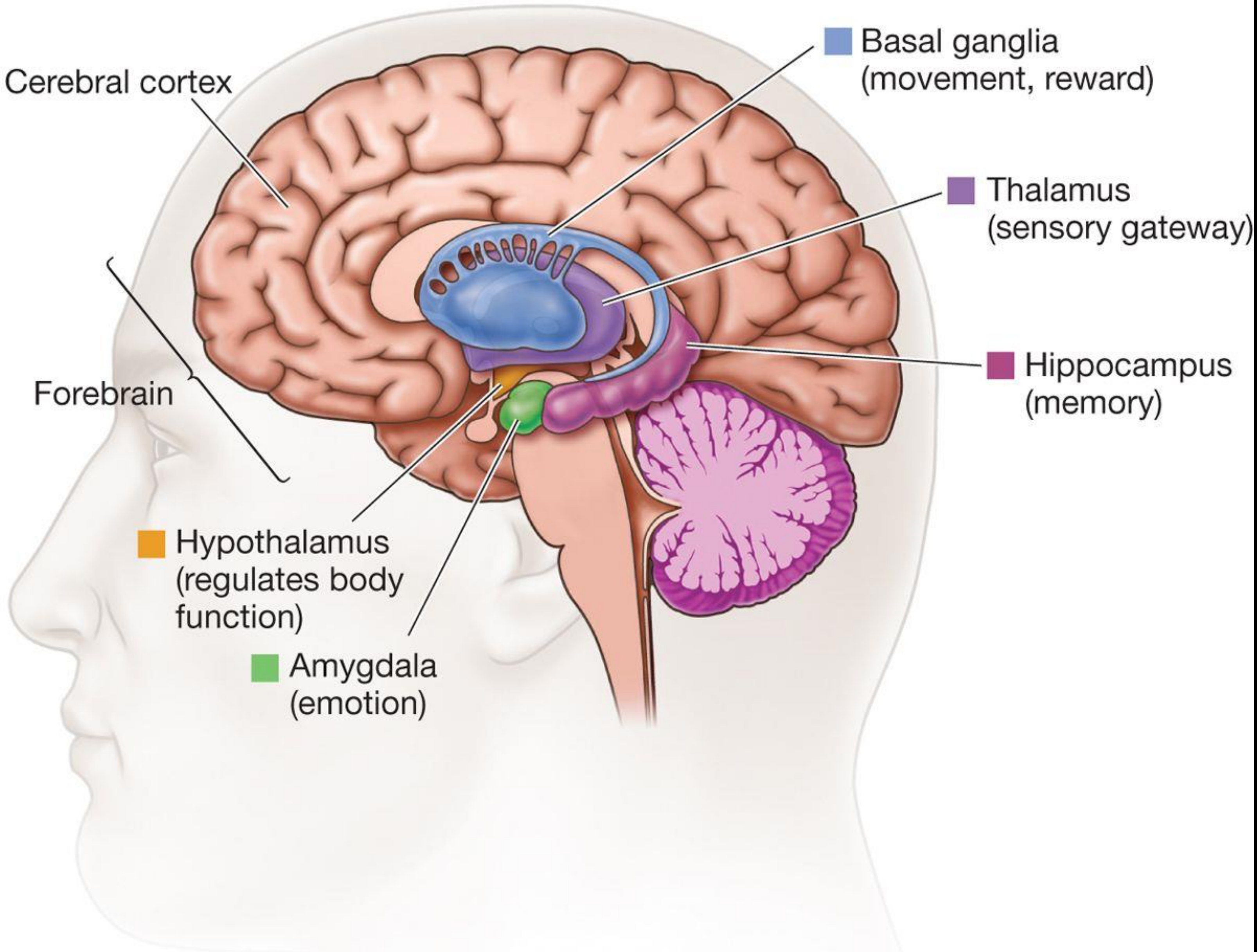
- Convolution NNs are very useful when dealing with image data

- Their main advantage is that the convolution filter itself is *learned* from the data itself, rather than be hand-engineered by a human

Special Topic: Biologically-Inspired Computing

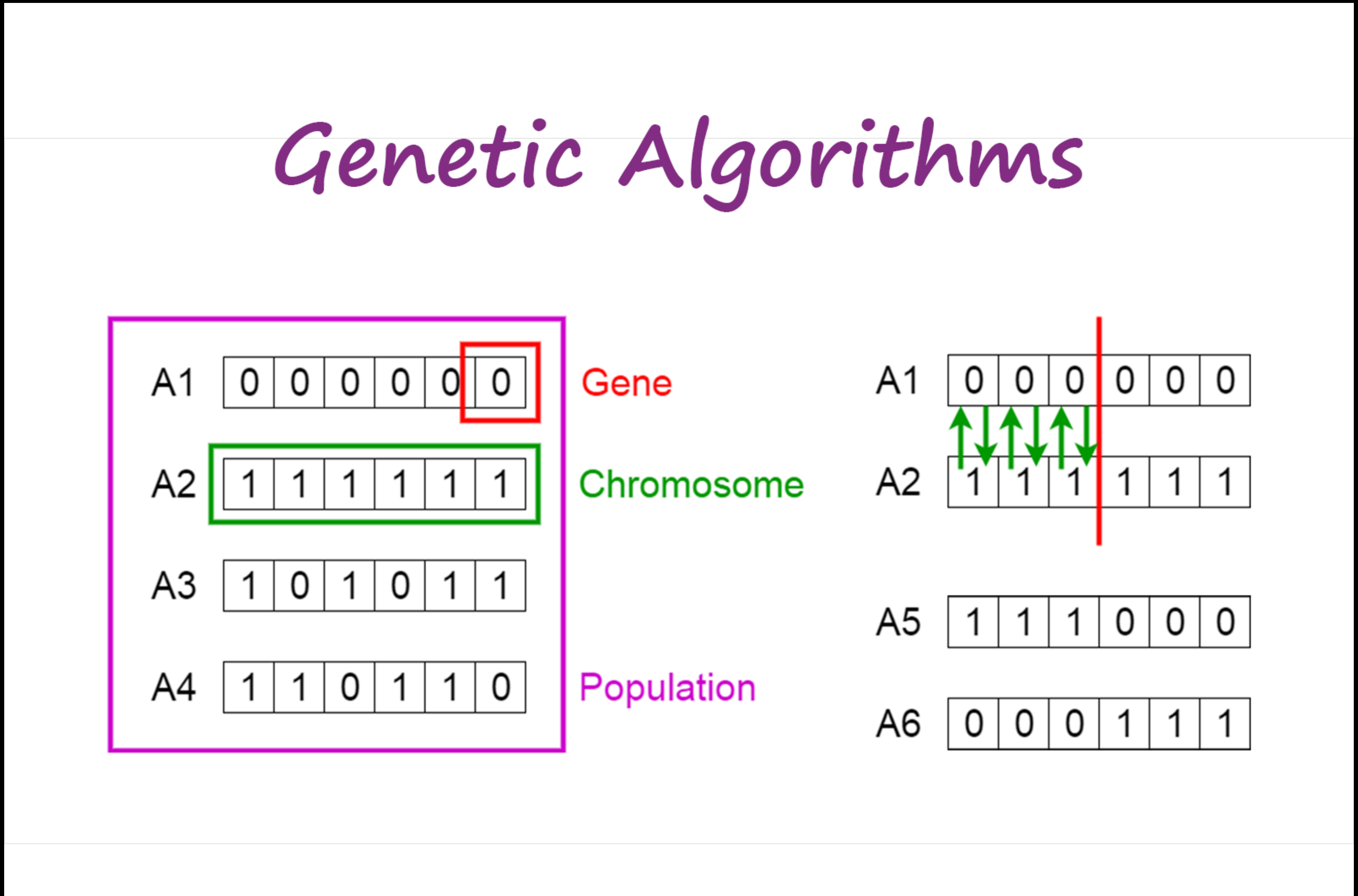
Deep Learning advocates claim that it emulates how the human brain functions. What is one fundamental problem with that argument?

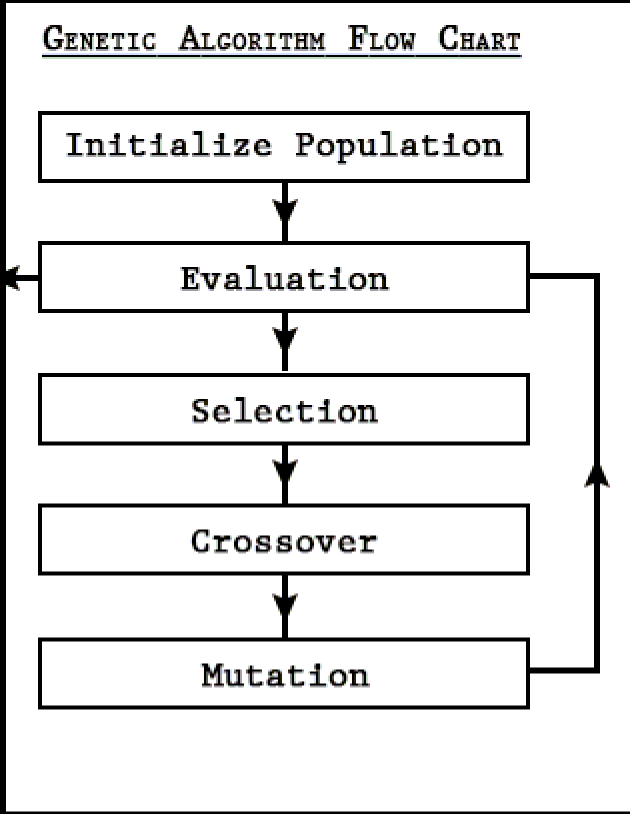
Human Brain



Cerebral cortex
Basal ganglia
(movement, reward)
Thalamus
(sensory gateway)
Forebrain
Hippocampus
(memory)
Hypothalamus
(regulates body
function)
Amygdala
(emotion)

Genetic Algorithms





[Fitness Function]

Genetic Algorithms

- Often used for optimization problems
- Also can be used as a smart method to search for best feature set in feature selection

Other Natured-Inspired Algorithms

- Particle Filtering (also Kalman filtering)
- Swarm Optimization (or particle swarm)
- Ant Colony Optimization
- Simulated Annealing

For next week

1) Project Proposals Due
2) Coding templates will post this week (Scikit, Spark)
3) HW3 releases tonight