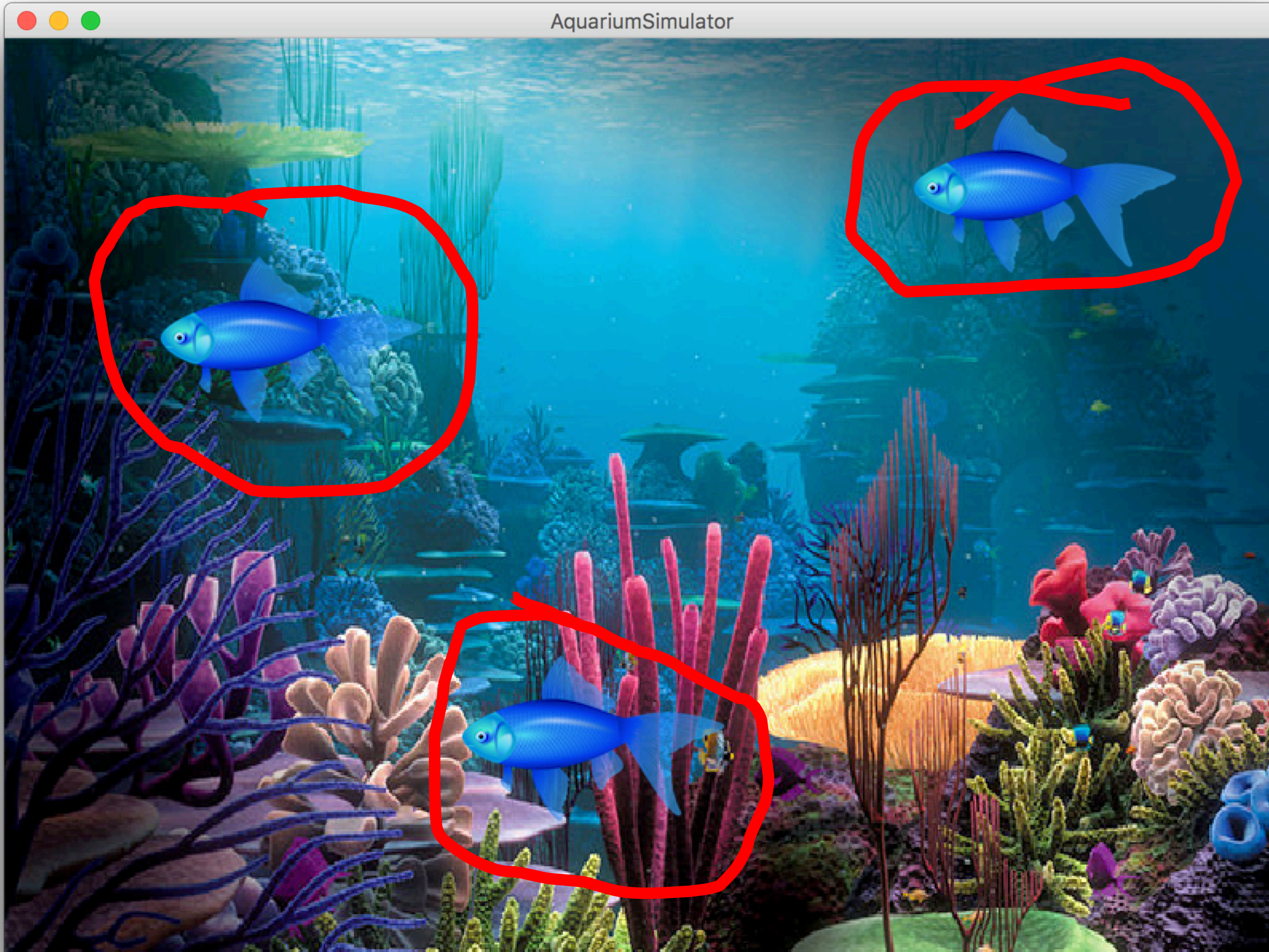
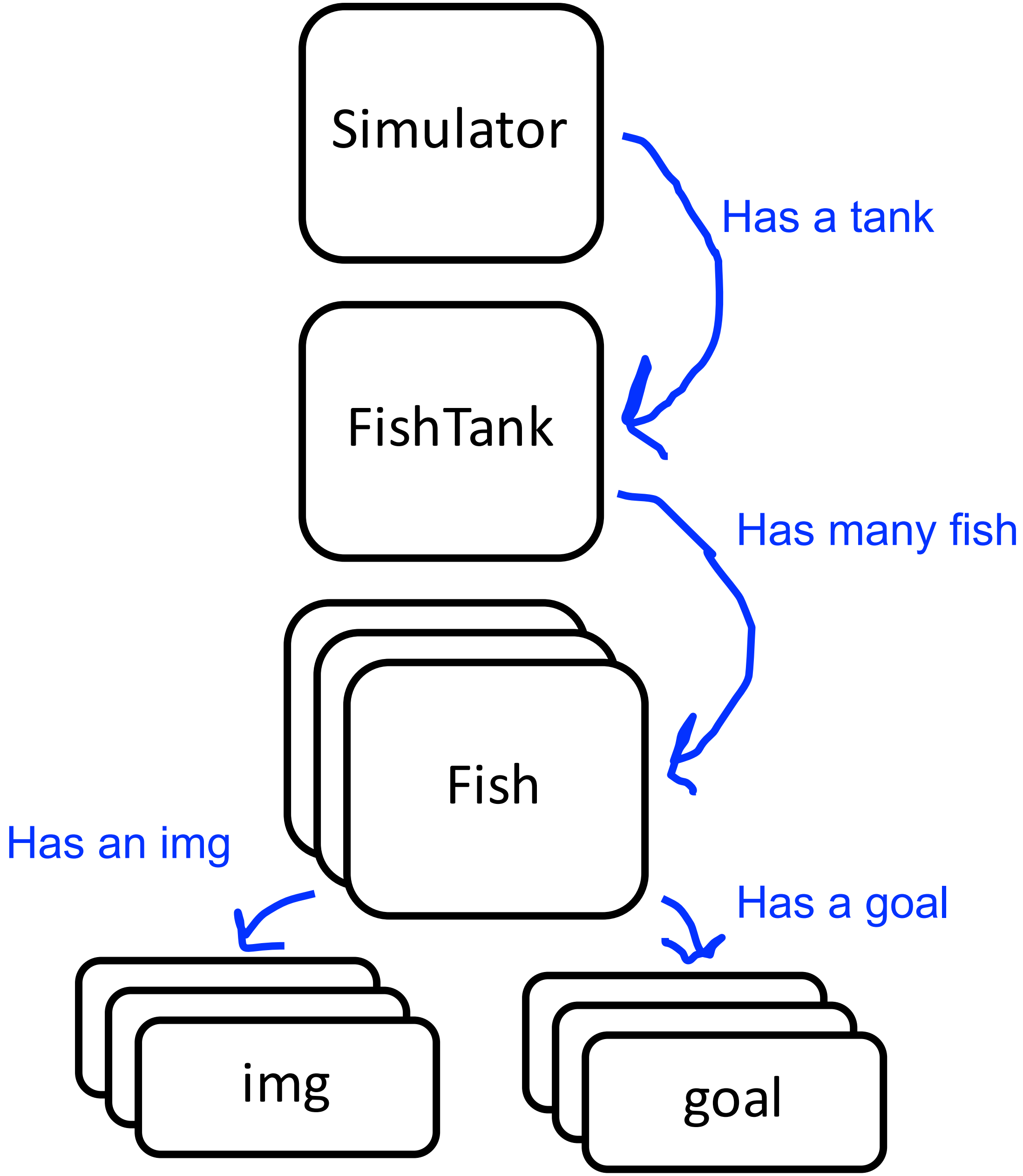
Machine Learning

Chris Piech
CS106A, Stanford University

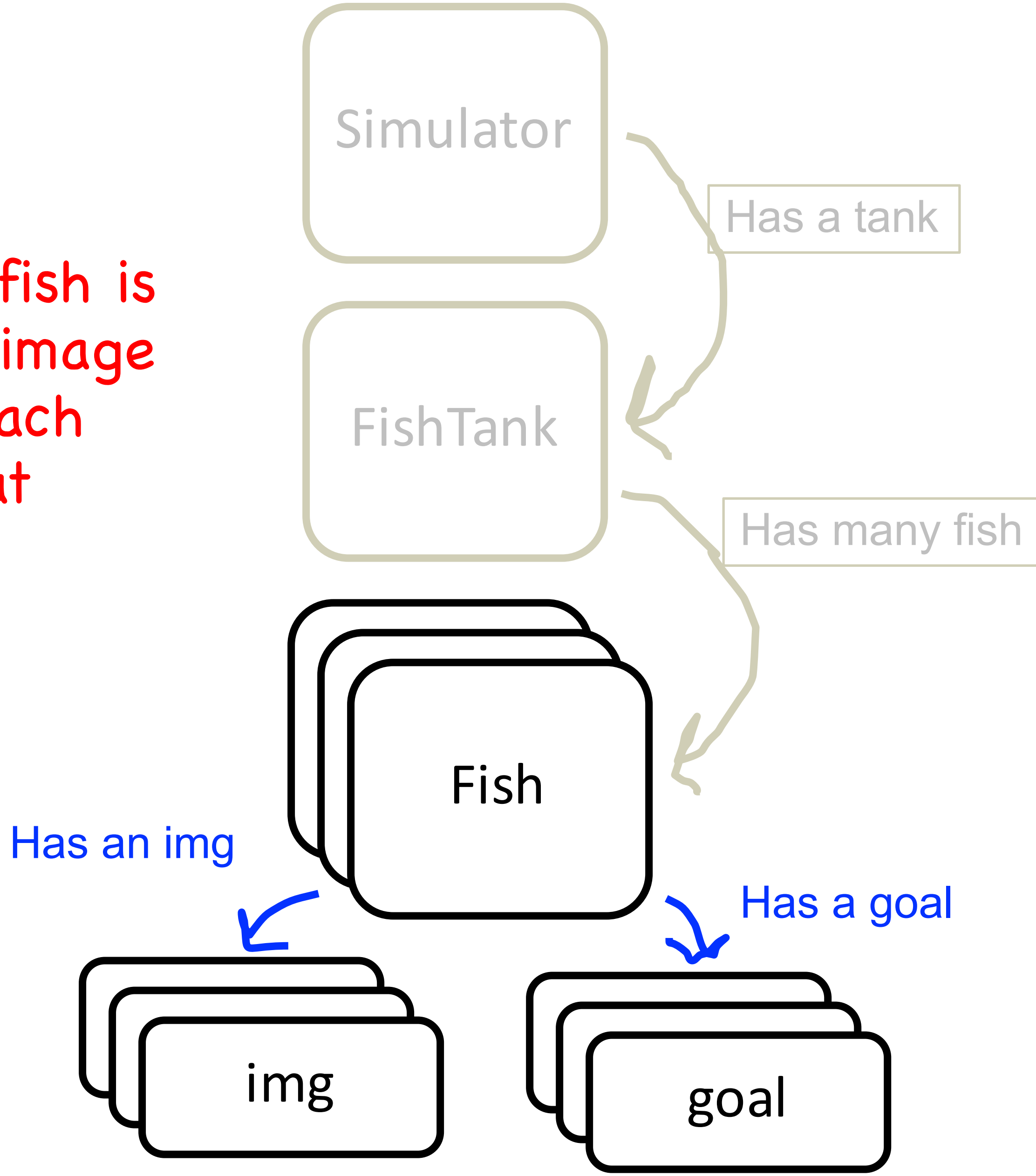

AquariumSimulator

Architecture



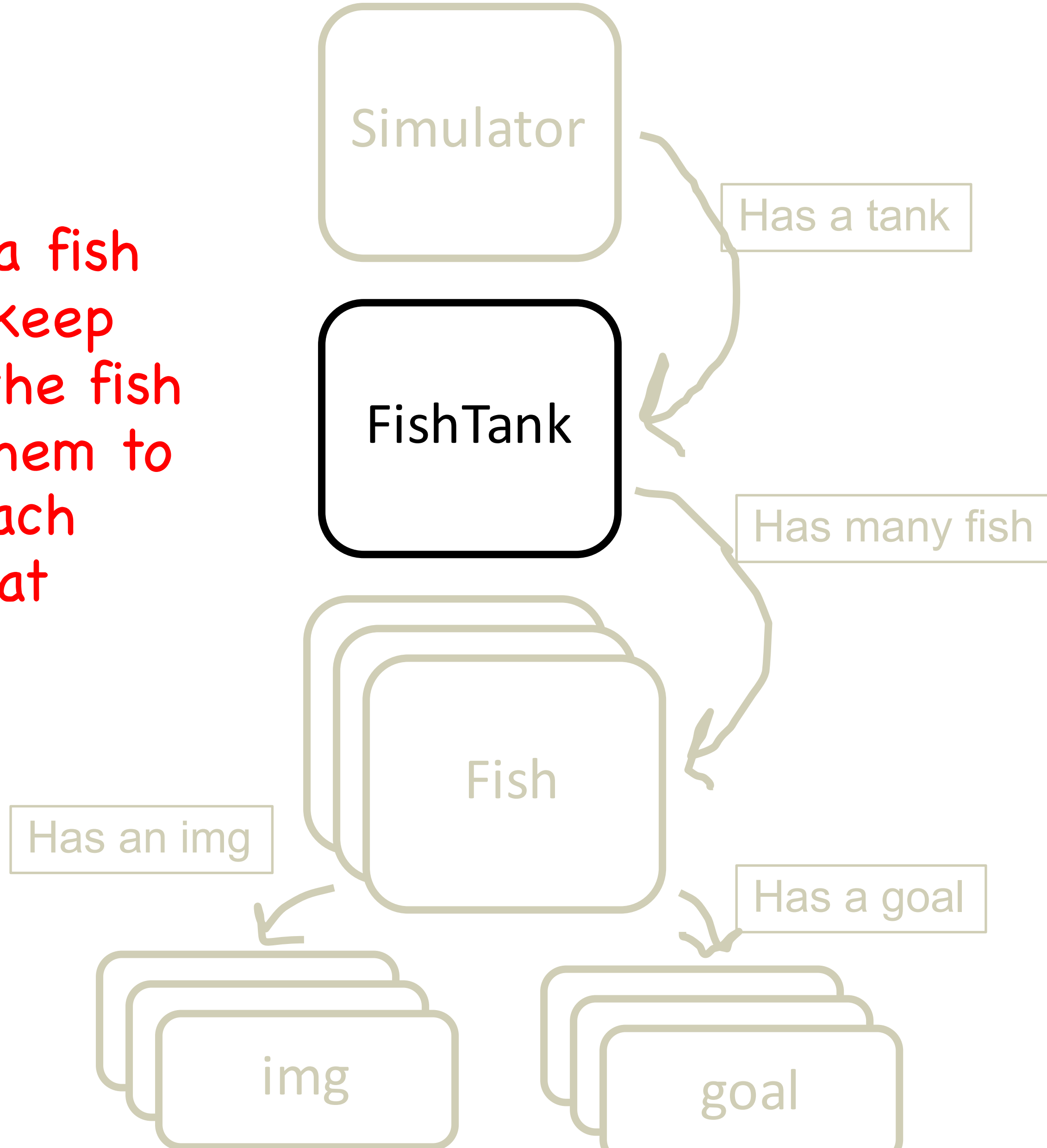
Architecture

The job of a fish is to update its image and goal each heartbeat



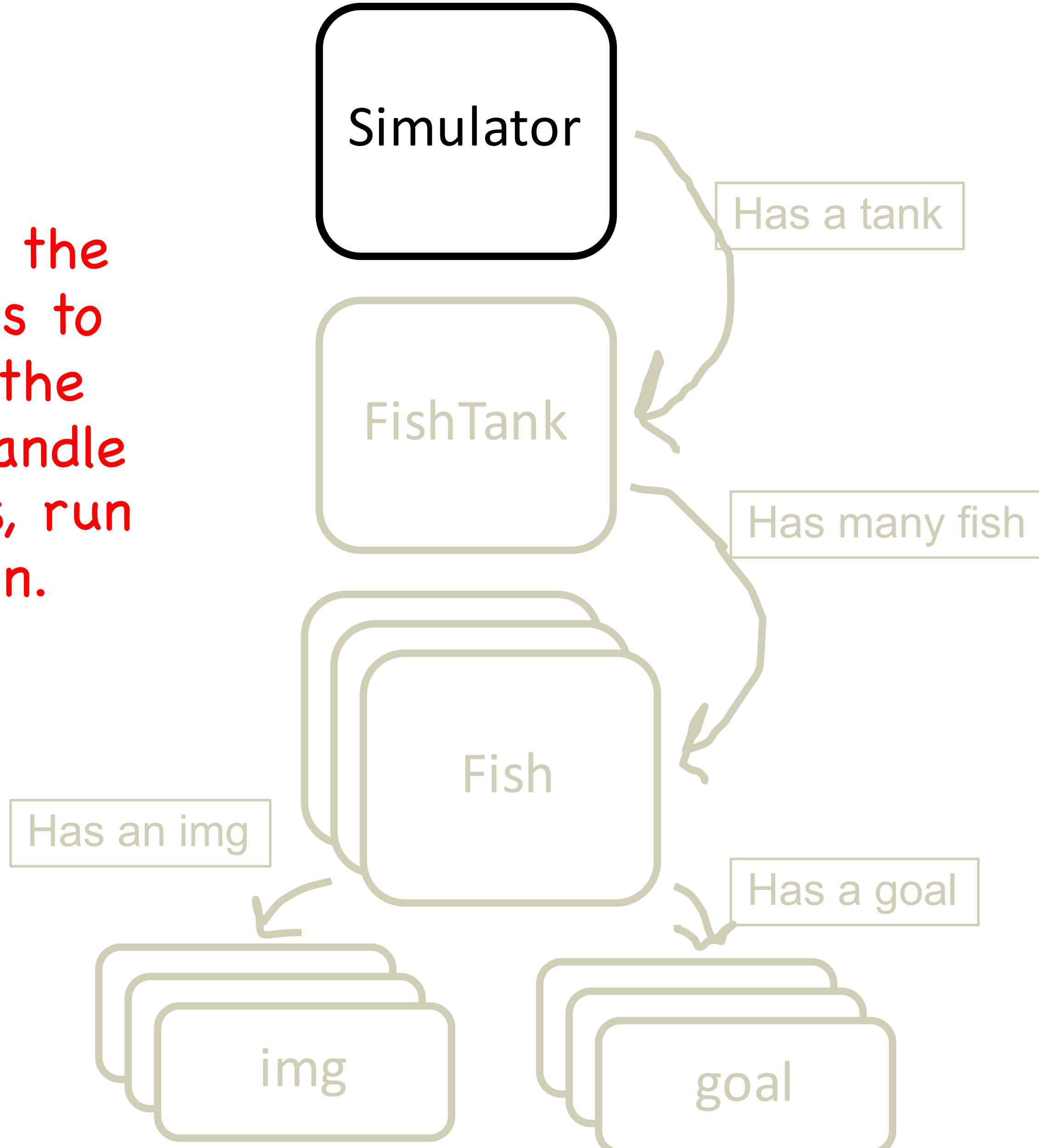
Architecture

The job of a fish tank is to keep track of all the fish and to tell them to update each heartbeat



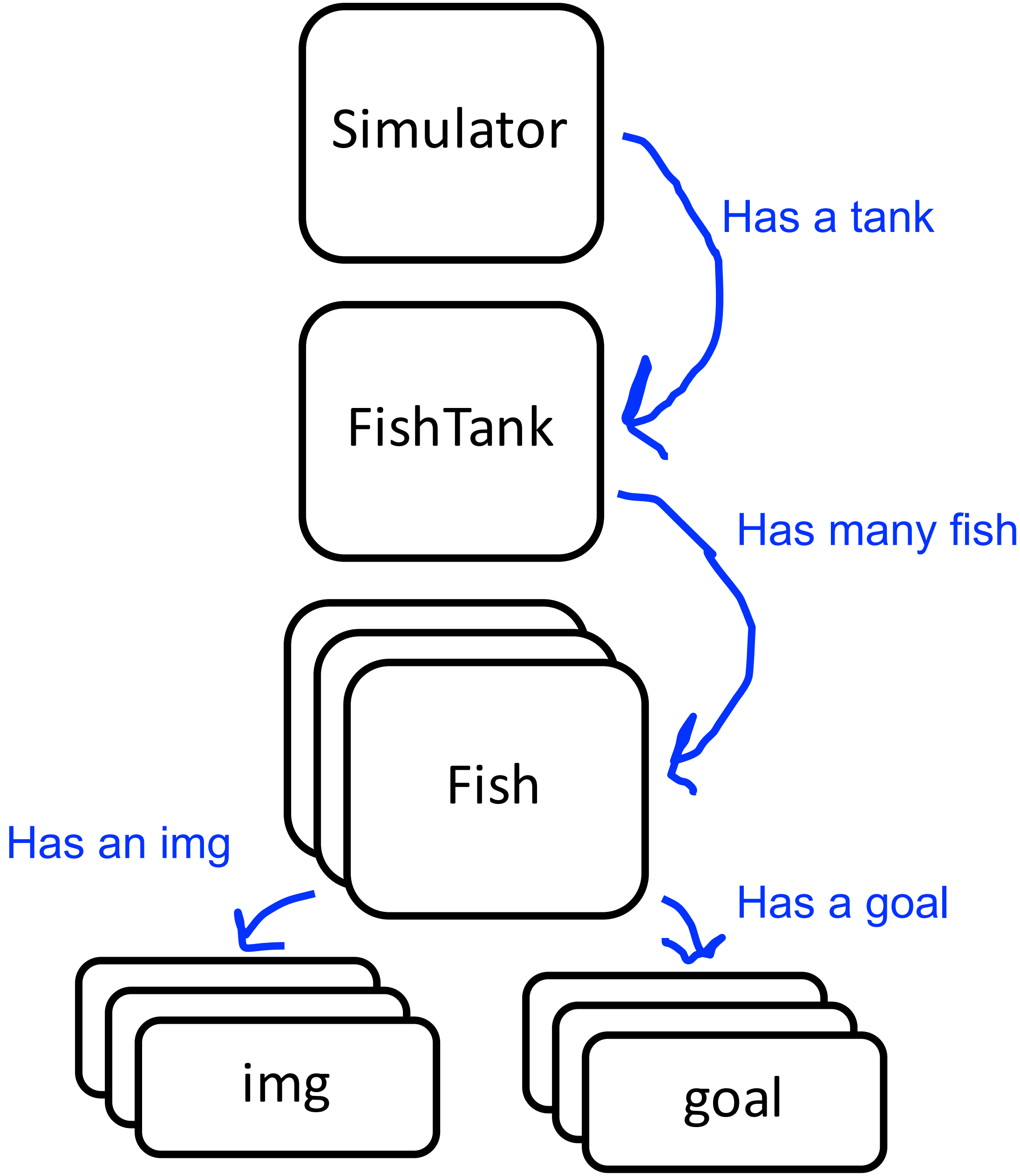
Architecture

The job of the simulator is to "control" the program. Handle user events, run animation.



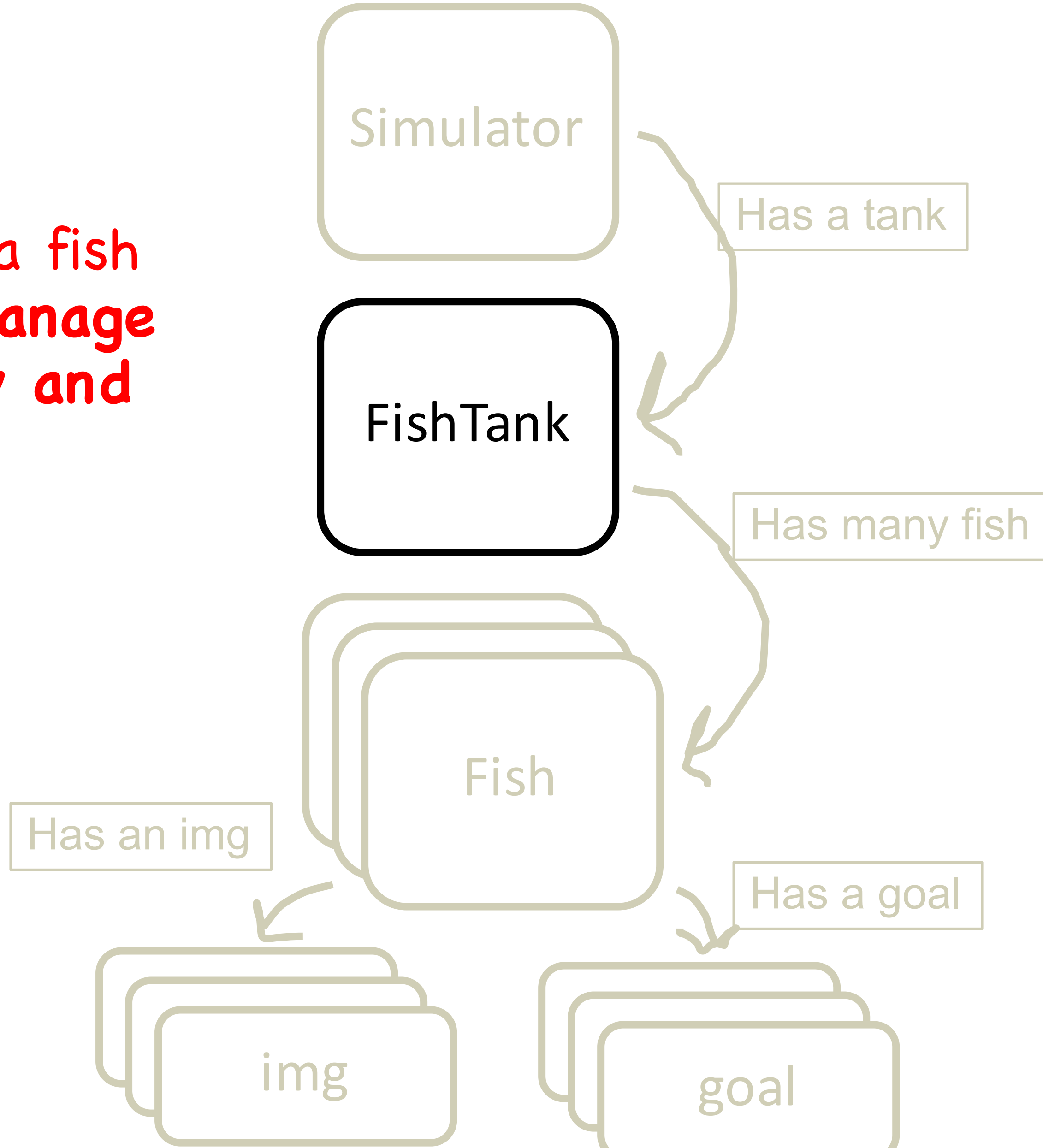
Whose job is it to put the imges on the screen?

Architecture



Architecture

The job of a fish tank is to **manage the display and state.**



extends

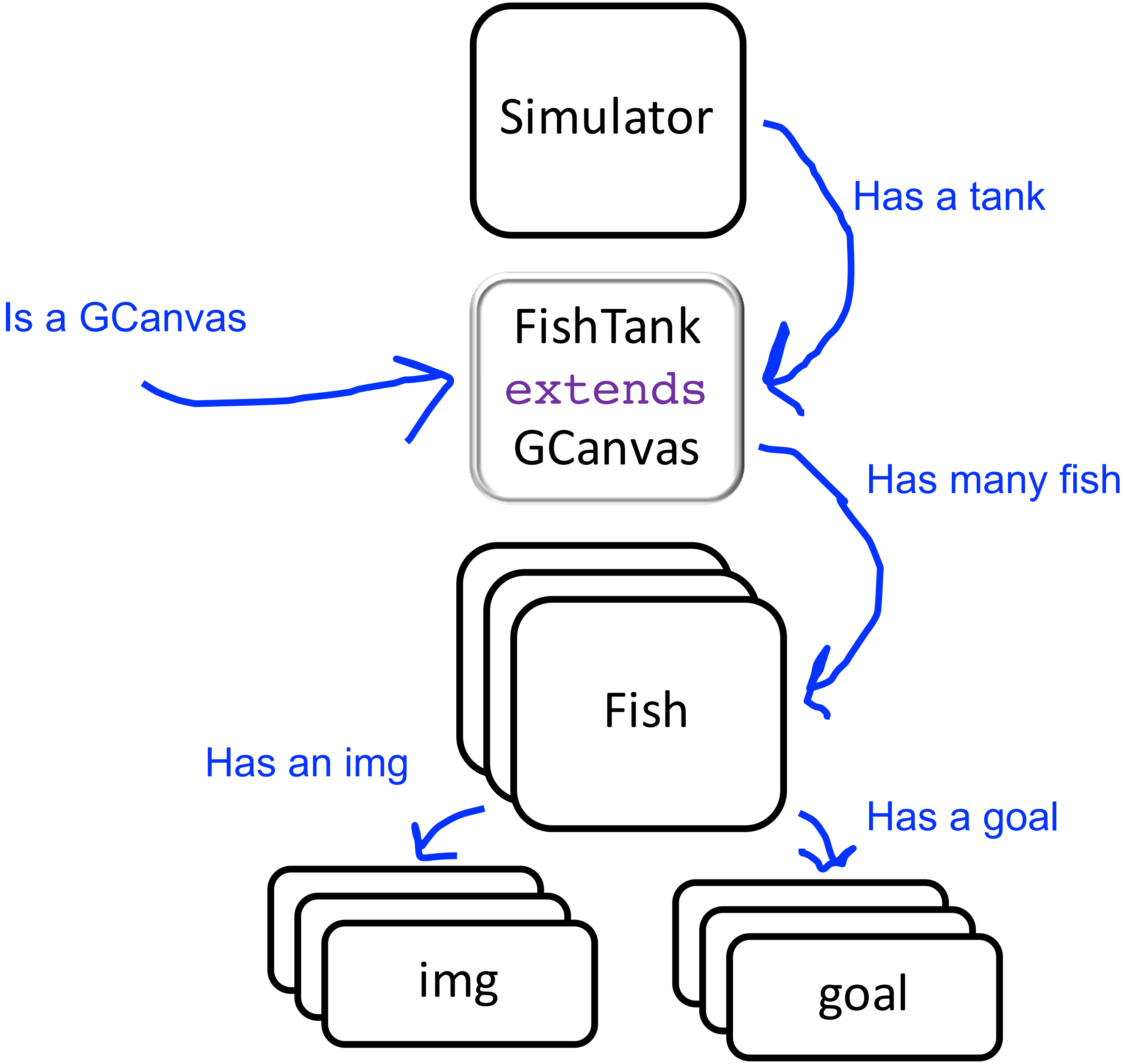
Make a class inherit all the
instance variables and methods
of another class

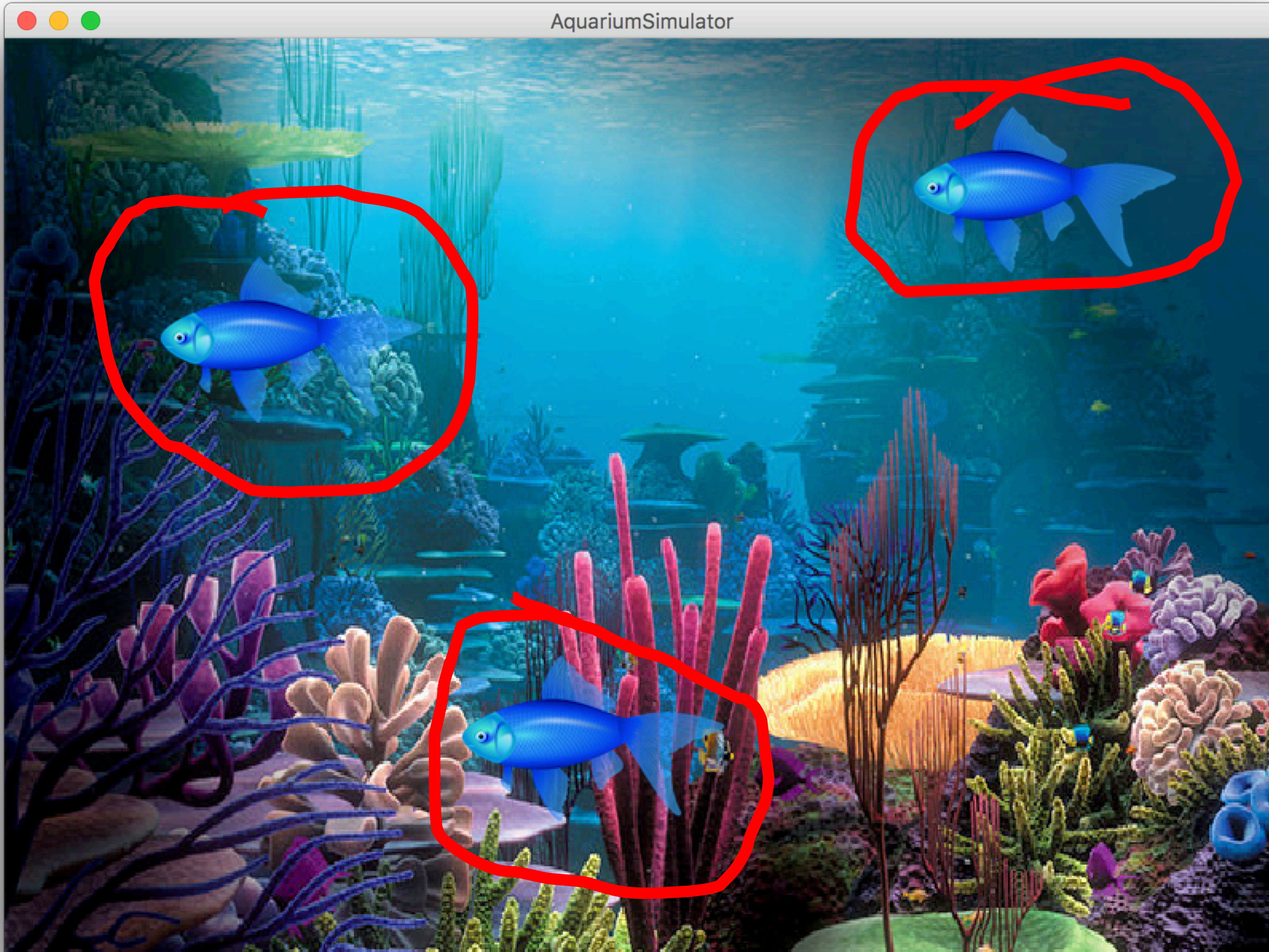
```
public class Simulator extends GraphicsProgram {
    // class definition
}
```

```java
public class NameSurferGraph extends GCanvas {
   // class definition
}
```

```
public class FishTank extends GCanvas {
   // class definition
}
```

Architecture




AquariumSimulator

implements

I promise that this class will define a few given methods

```java
public class NameSurferGraph extends GCanvas,
   implements ComponentListener {
    // class definition
}
```

Also a cheeky way to share constants between classes

`implements`

I promise that this class will define
a few given methods

Machine Learning

Machine Learning
or, How we learned to decompose

There is something going on
in the world of AI

Something big (for us)…

[suspense]

How can we develop intelligent **agents?**

Volunteer

Computer programs

How can we develop intelligent **agents?**

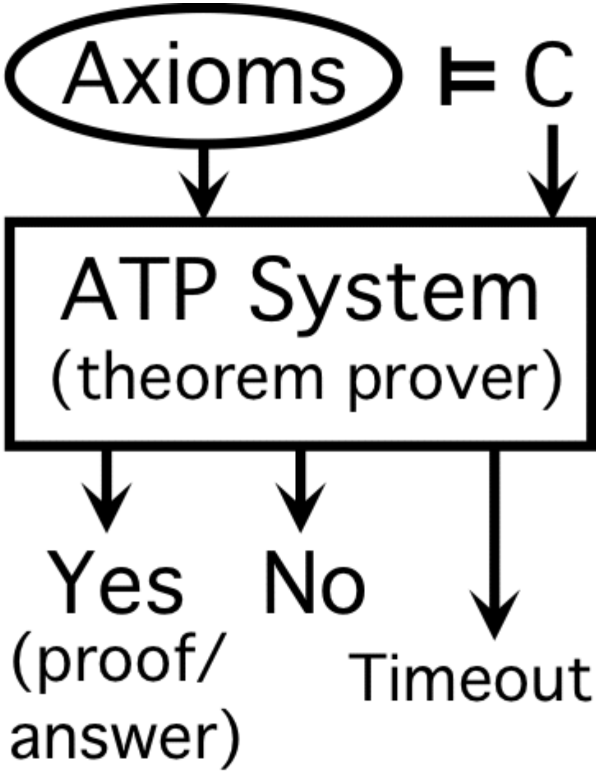
Better than chance

As well as humans

Early Optimism 1950

1952

1955



Early Optimism 1950

"Machines will be capable, within twenty years, of doing any work a man can do."
–Herbert Simon, 1952

Underwhelming Results 1950s to 1980s

The spirit is willing but the flesh is weak.

↓

(Russian)

↓

The vodka is good but the meat is rotten.

The world is too complex


BRACE YOURSELVES
WINTER IS COMING

Machine Learning: Learn From Experience

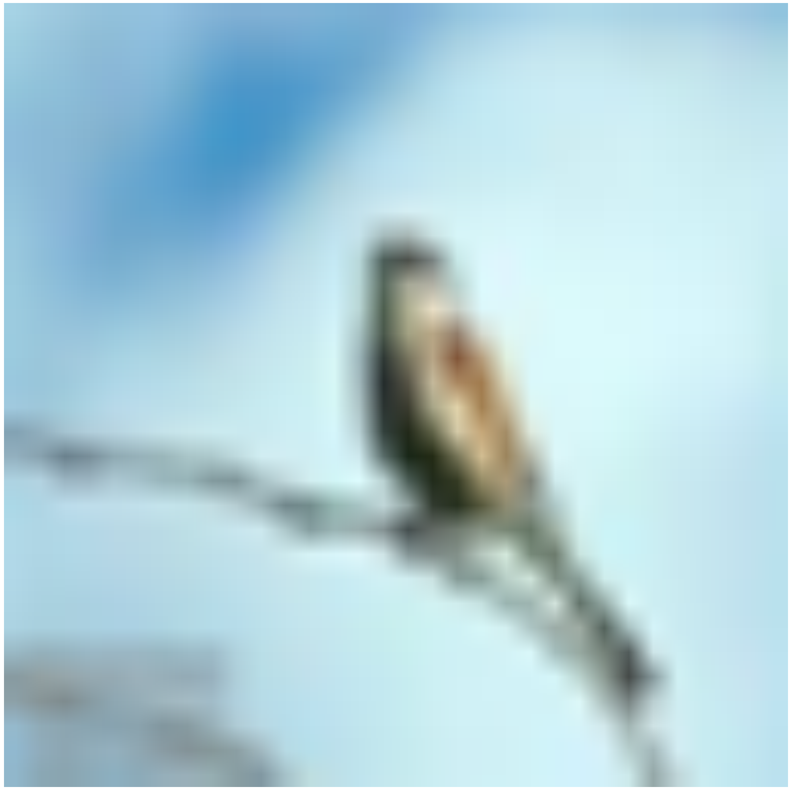
Some success

Hard problems seemed impossible.

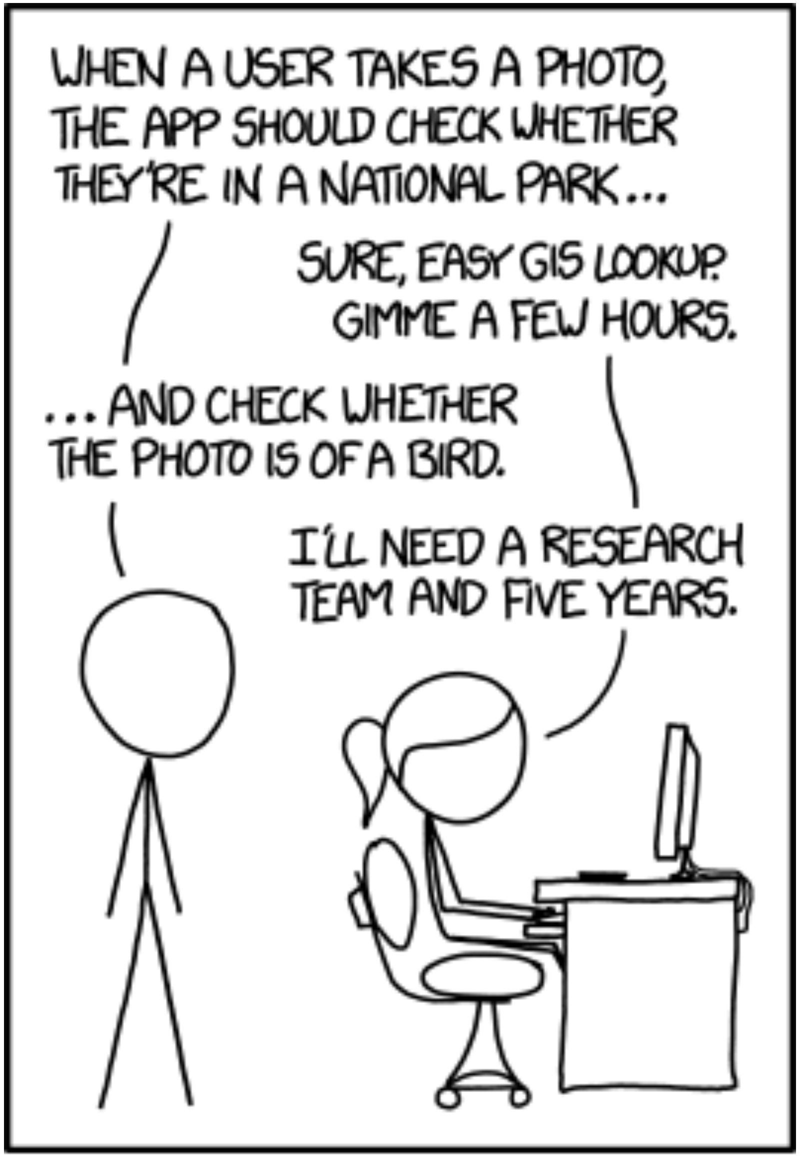

kirkh on deviantart
COMPUR

Can we predict hand written digits?

Can we predict birds vs planes?



Vision is Hard



Vision is Hard

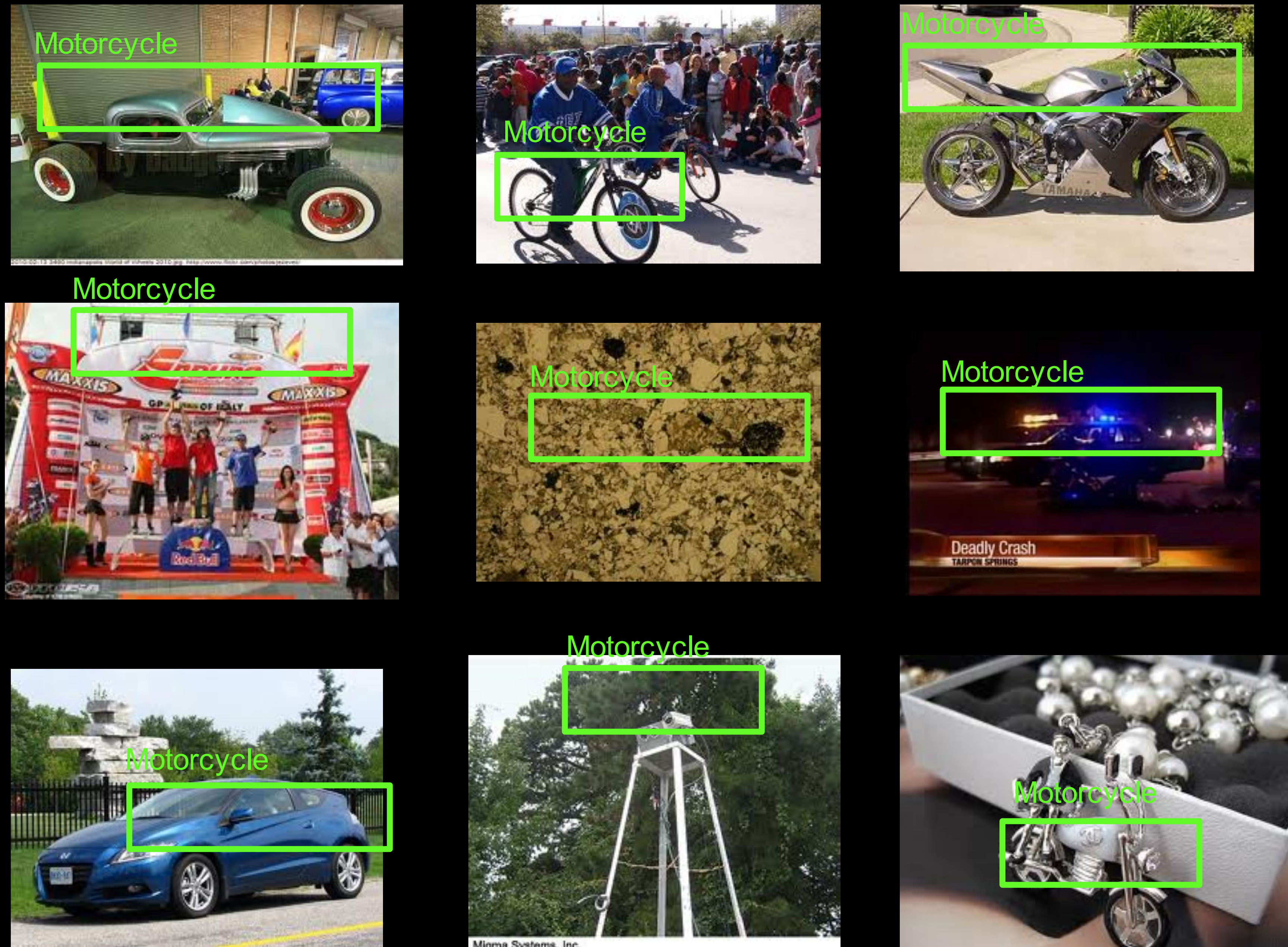
You see this:

But the camera sees this:

194	210	201	212	199	213	215	195	178	158	182	209
180	189	190	221	209	205	191	167	147	115	129	163
114	126	140	188	176	165	152	140	170	106	78	88
87	103	115	154	143	142	149	153	173	101	57	57
102	112	106	131	122	138	152	147	128	84	58	66
94	95	79	104	105	124	129	113	107	87	69	67
68	71	69	98	89	92	98	95	89	88	76	67
41	56	68	99	63	45	60	82	58	76	75	65
20	43	69	75	56	41	51	73	55	70	63	44
50	50	57	69	75	75	73	74	53	68	59	37
72	59	53	66	84	92	84	74	57	72	63	42
67	61	58	65	75	78	76	73	59	75	69	50

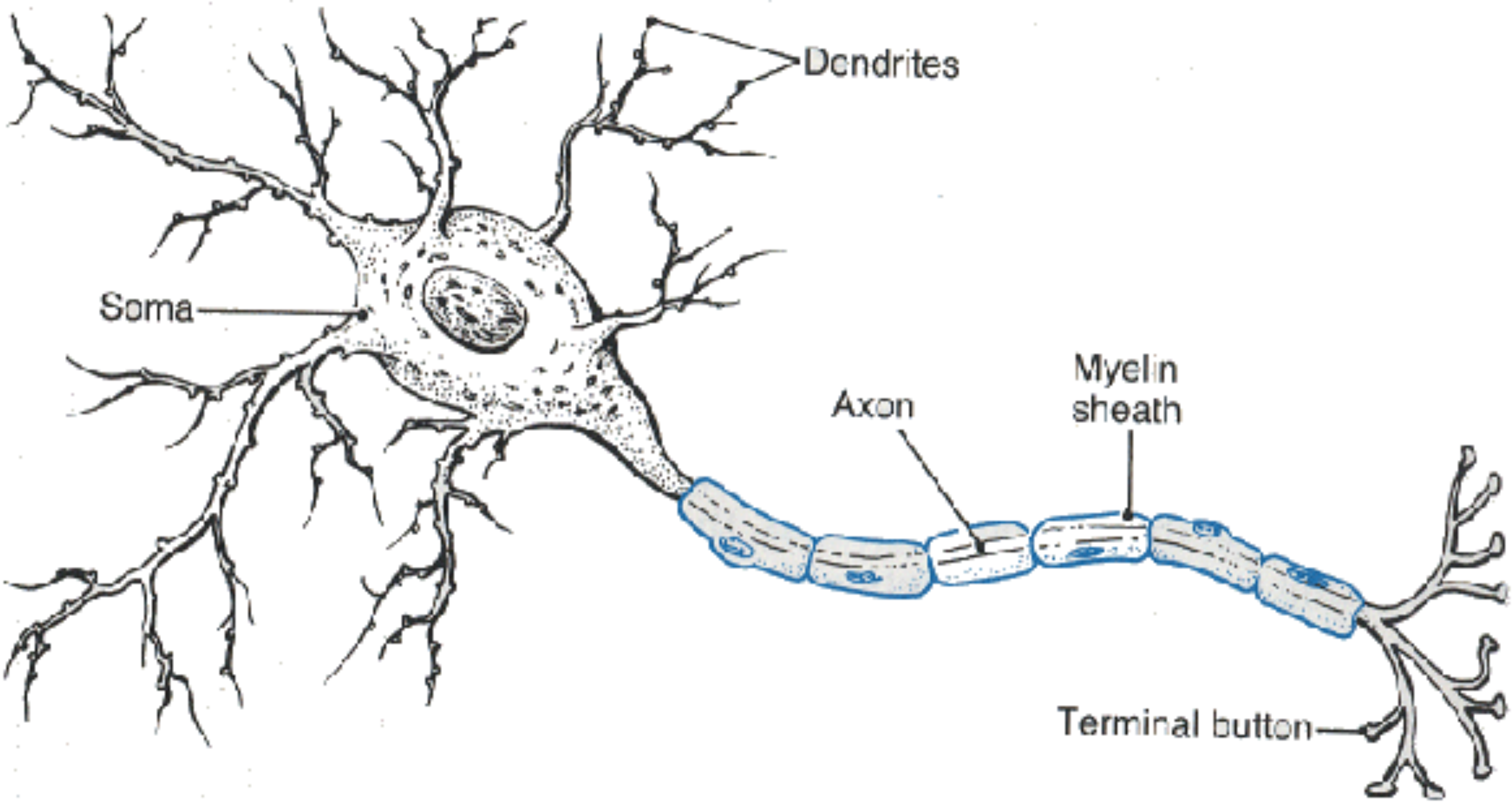
[Andrew Ng]

Not Perfect...

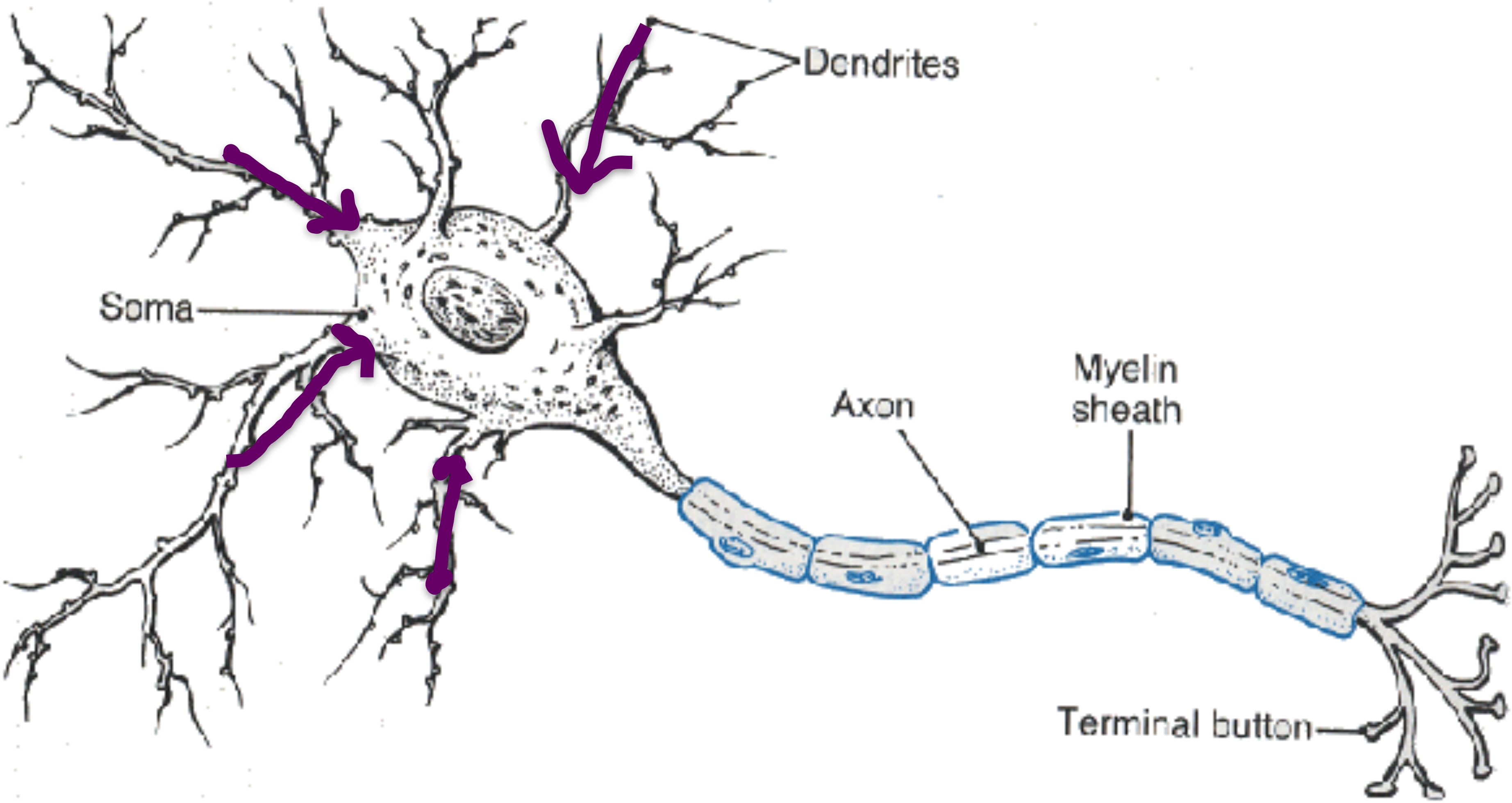

Motorcycle
Motorcycle
Motorcycle
Motorcycle
MAXXIS
MAXXIS
Red Bull
Motorcycle
Motorcycle
Deadly Crash
TARPON SPRINGS
Motorcycle
Motorcycle
Migma Systems, Inc.
Motorcycle

Great idea inspired by biology

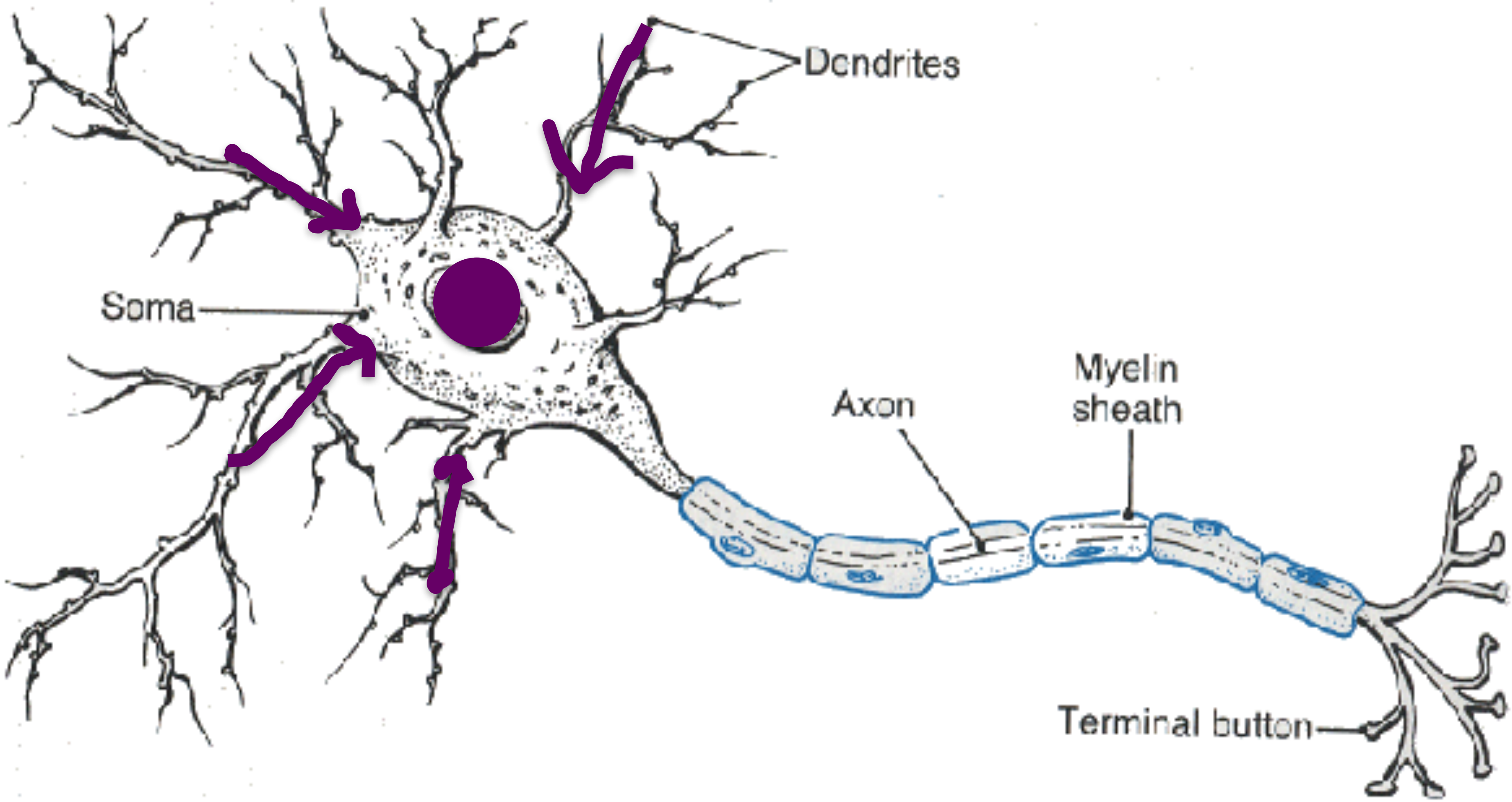
Neuron



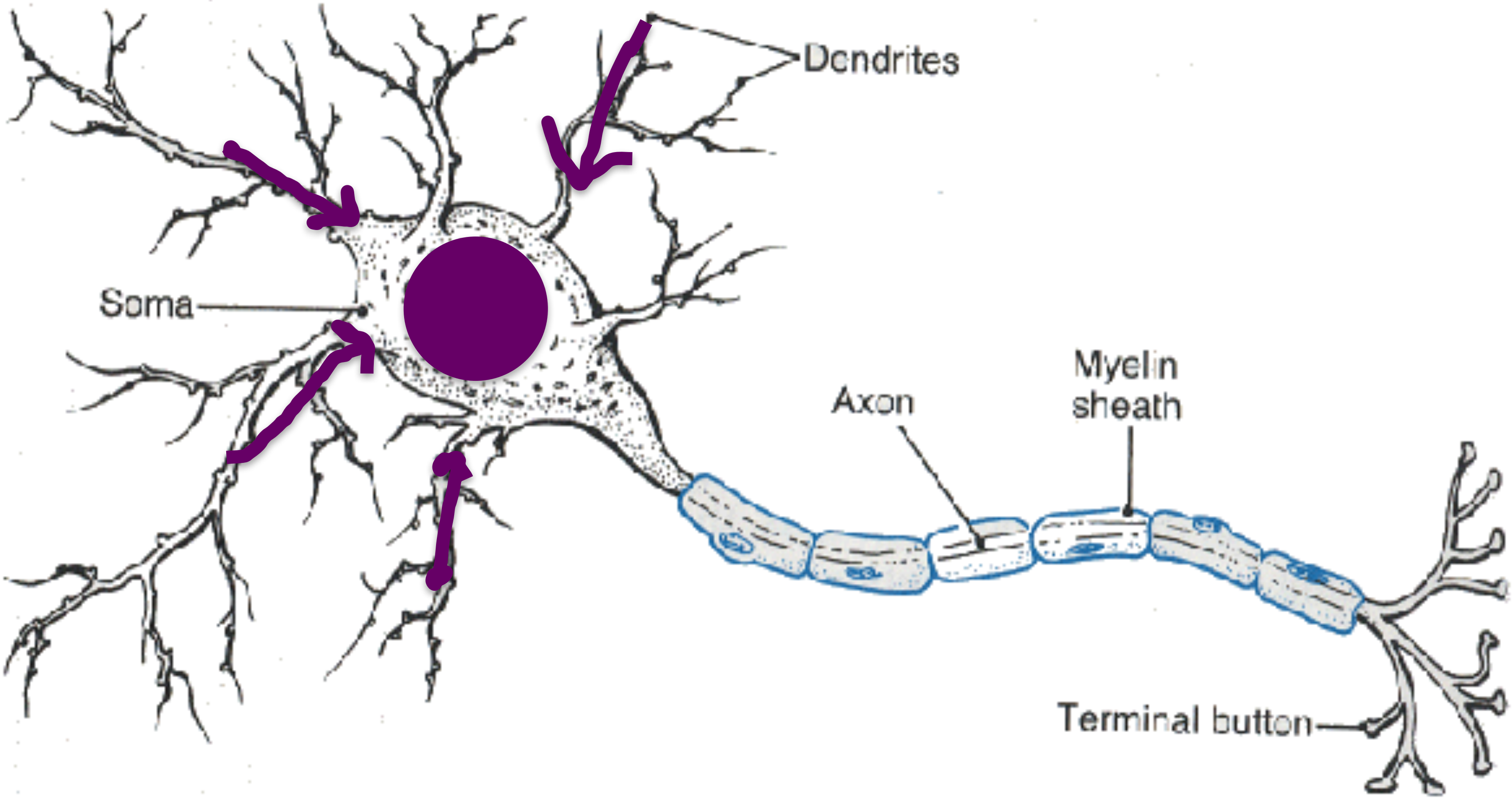
Neuron



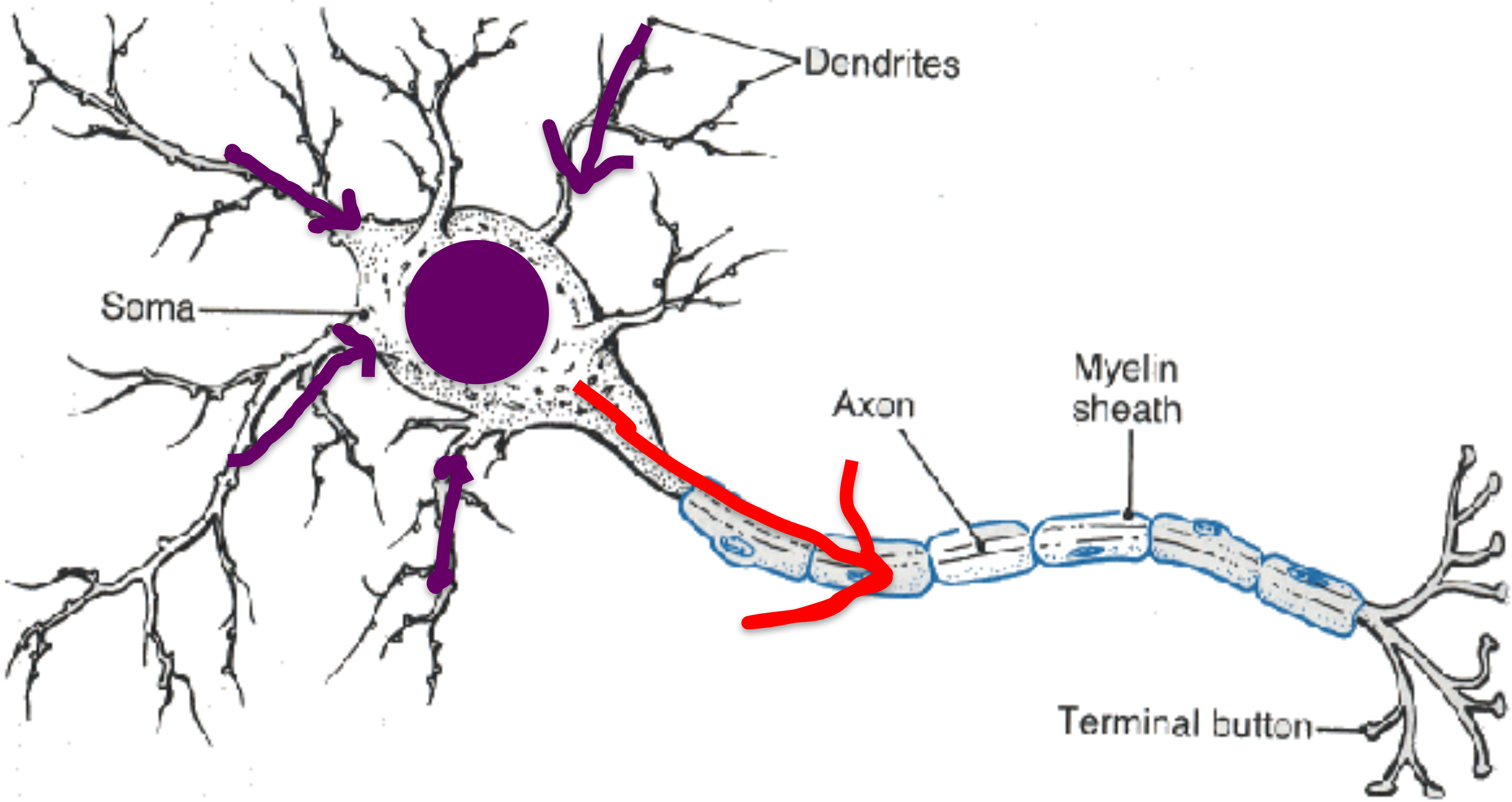
Neuron



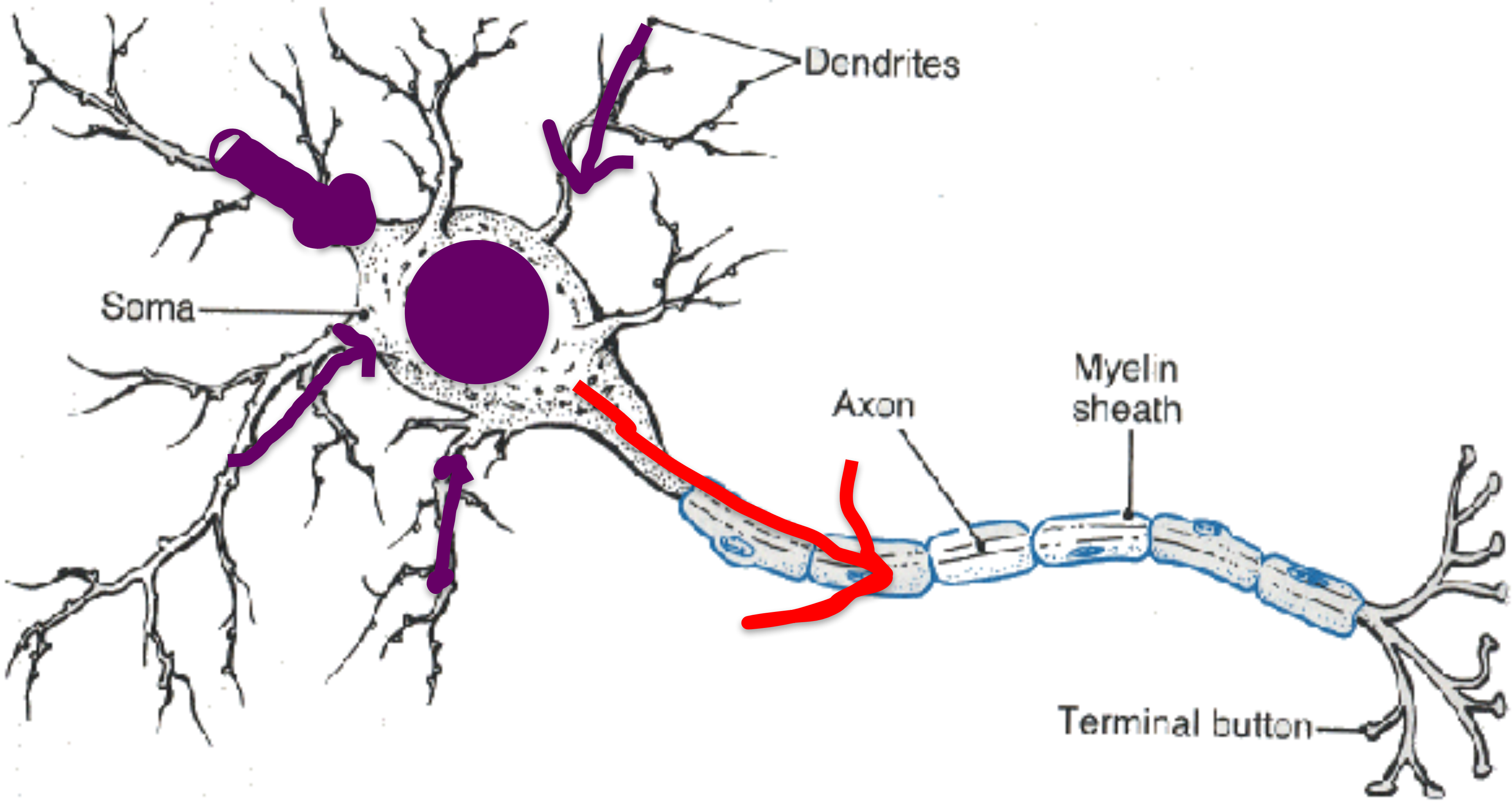
Neuron



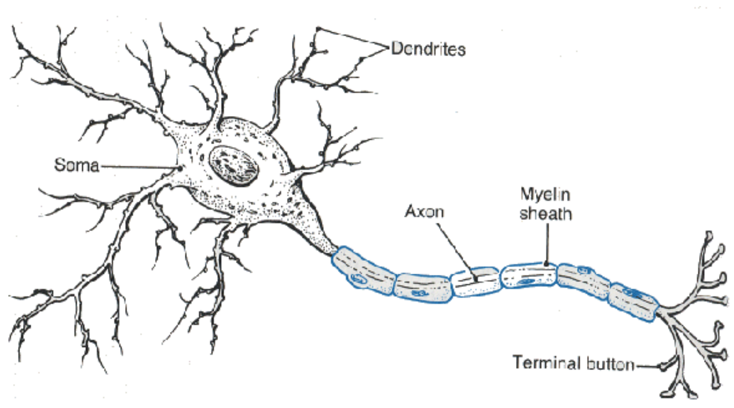
Neuron

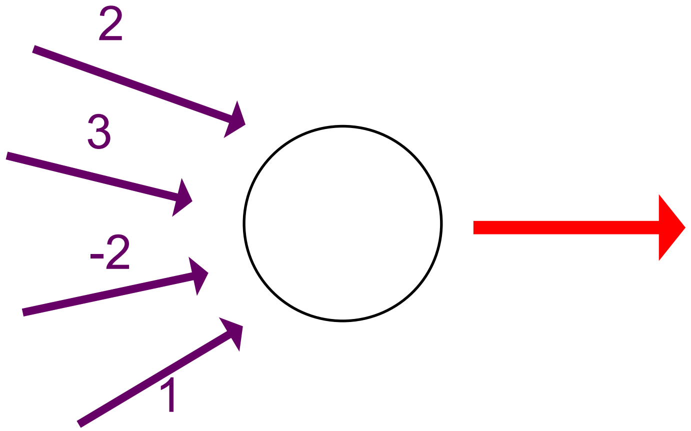


Some Inputs are More Important

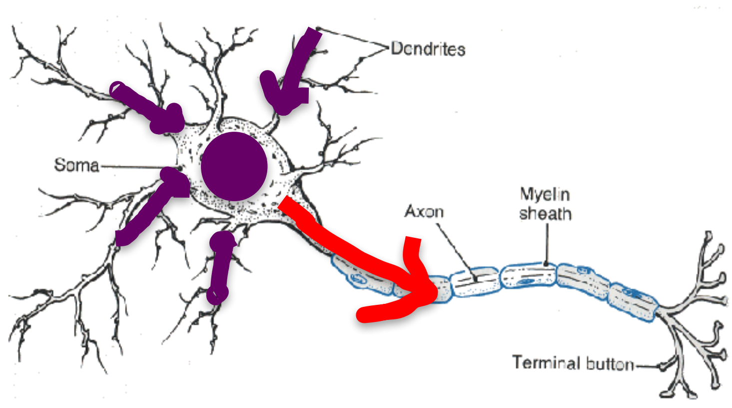


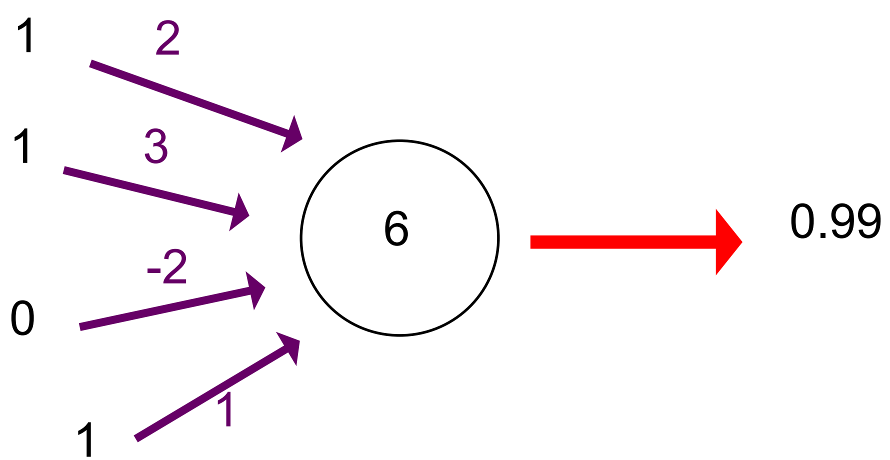
Artificial Neuron



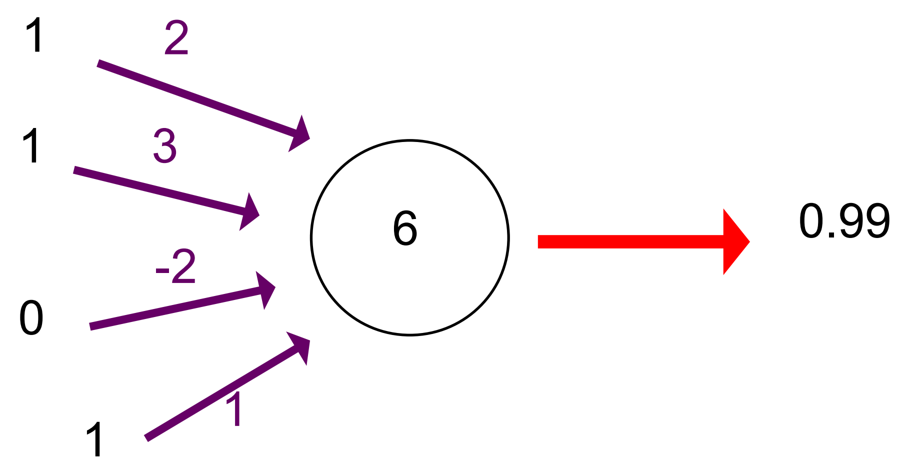


Artificial Neuron



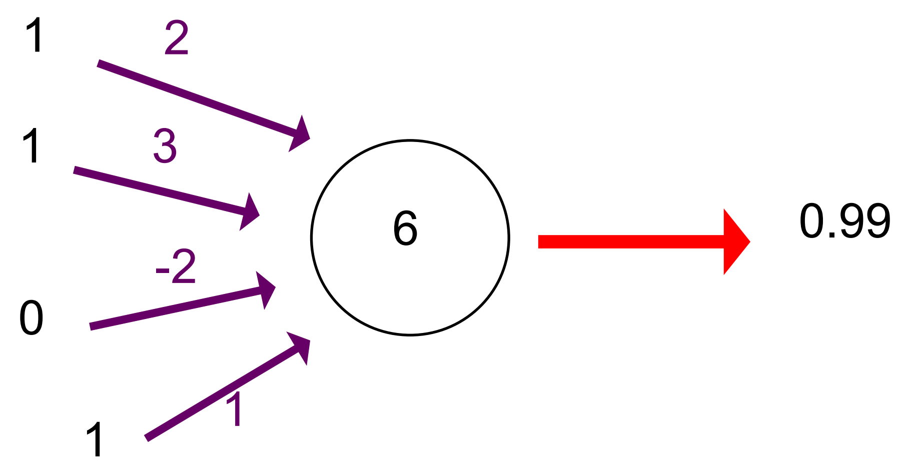


Artificial Neuron



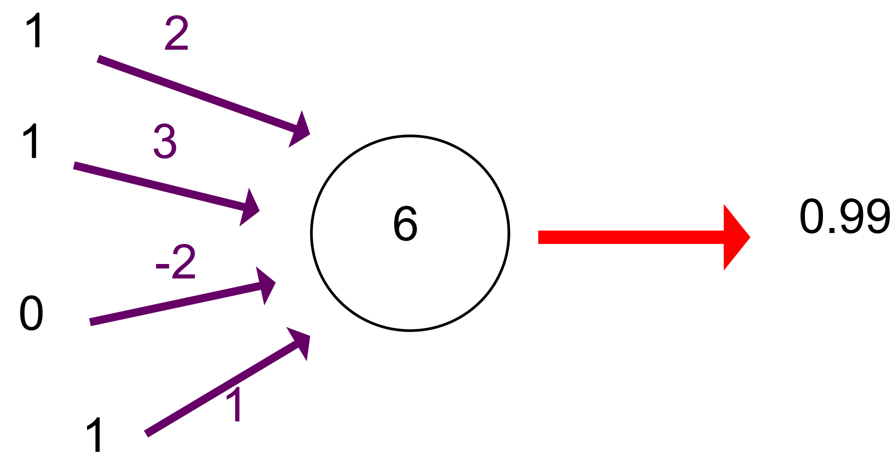

```
buildup = input1 * weight1 +
          input2 * weight2 +
          input3 * weight3 +
          input4 * weight4
```

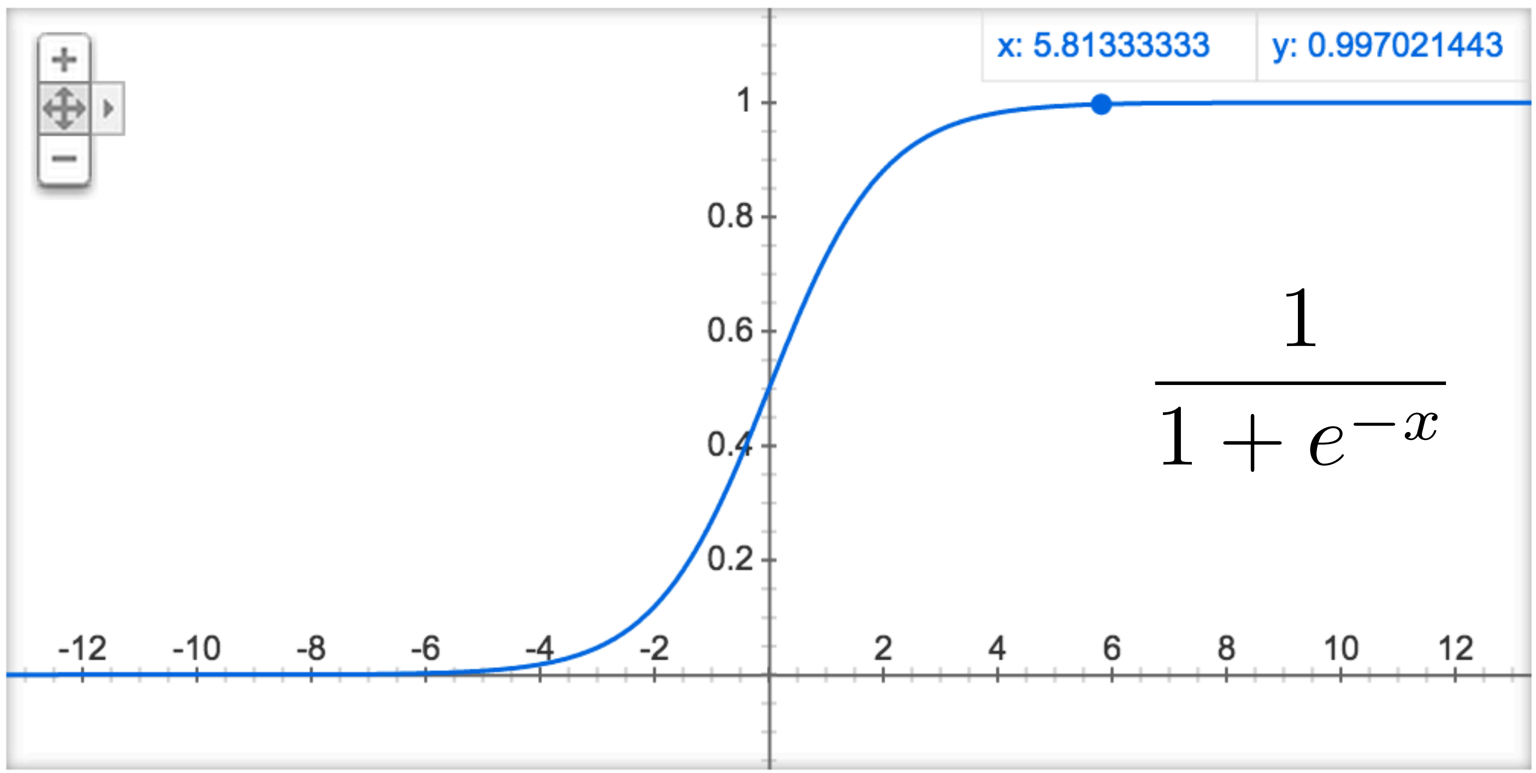
Artificial Neuron




```
buildup = 1 * 2 +
          1 * 3 +
          0 * -2 +
          1 * 1
```

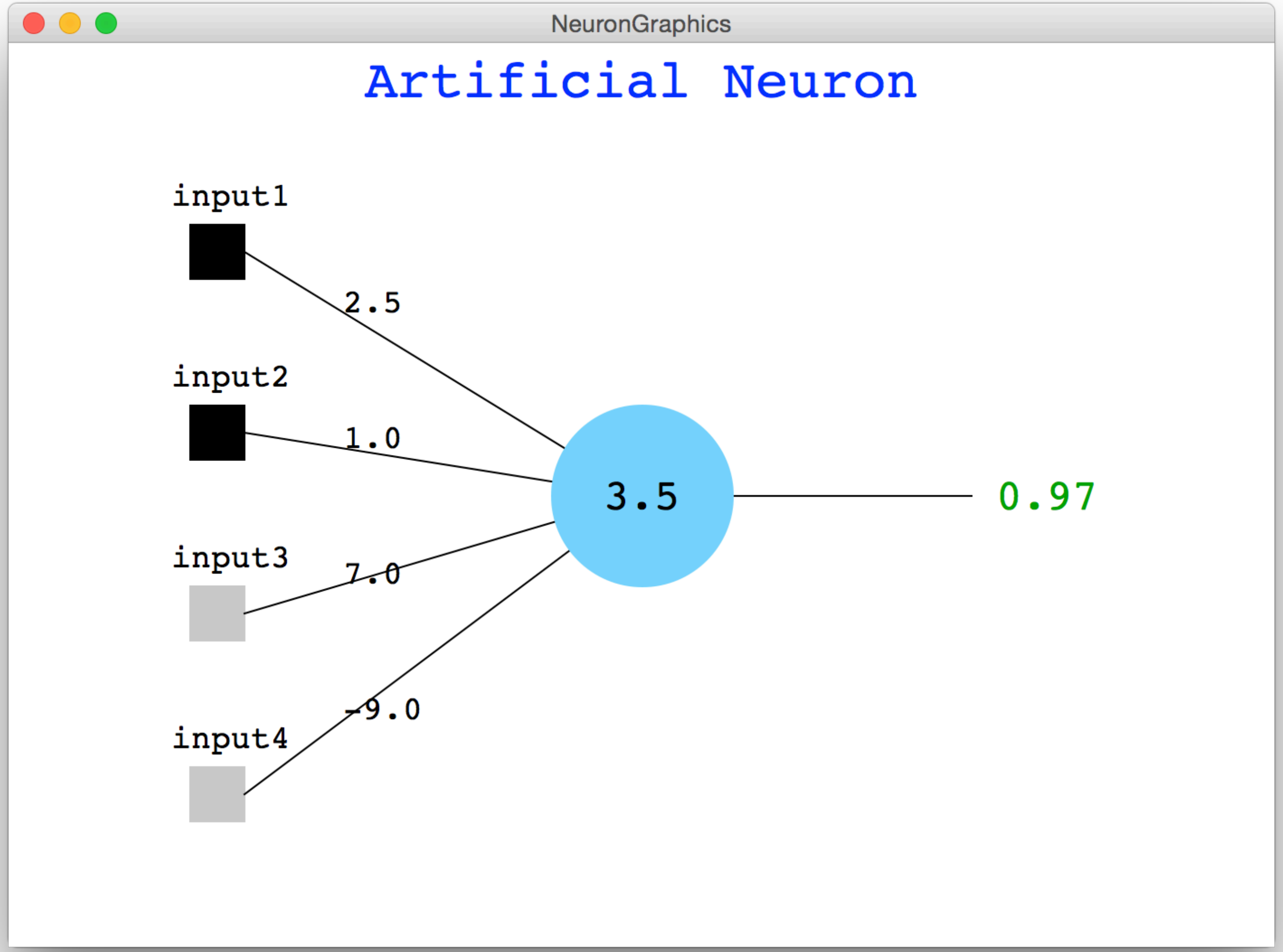
Sigmoid Function





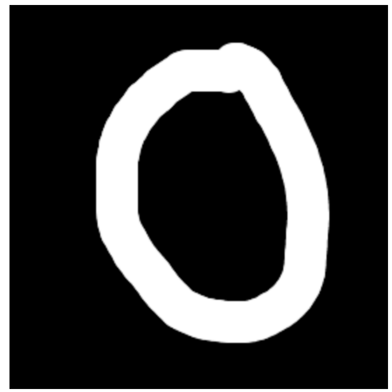
$$\frac{1}{1+e^{-x}}$$

Java Demo


NeuronGraphics
Artificial Neuron
input1
input2
input3
input4
2.5
1.0
7.0
-9.0
3.5
0.97

Digit Recognition Example

Let's make feature vectors from pictures of numbers



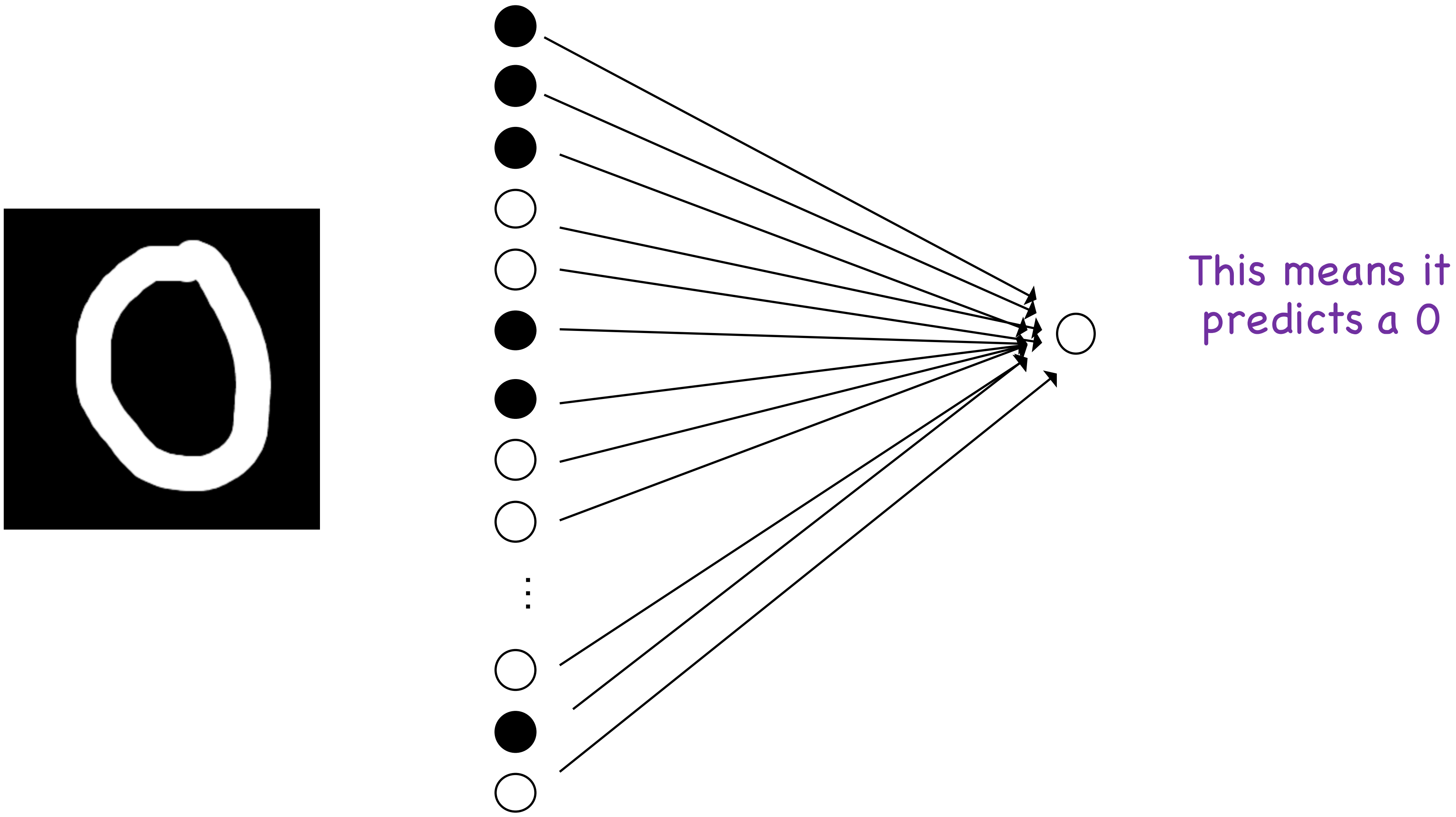
input $= [0, 0, 0, 0, \ldots, 1, 0, 0, 1, \ldots 0, 0, 1, 0]$

label $= 0$

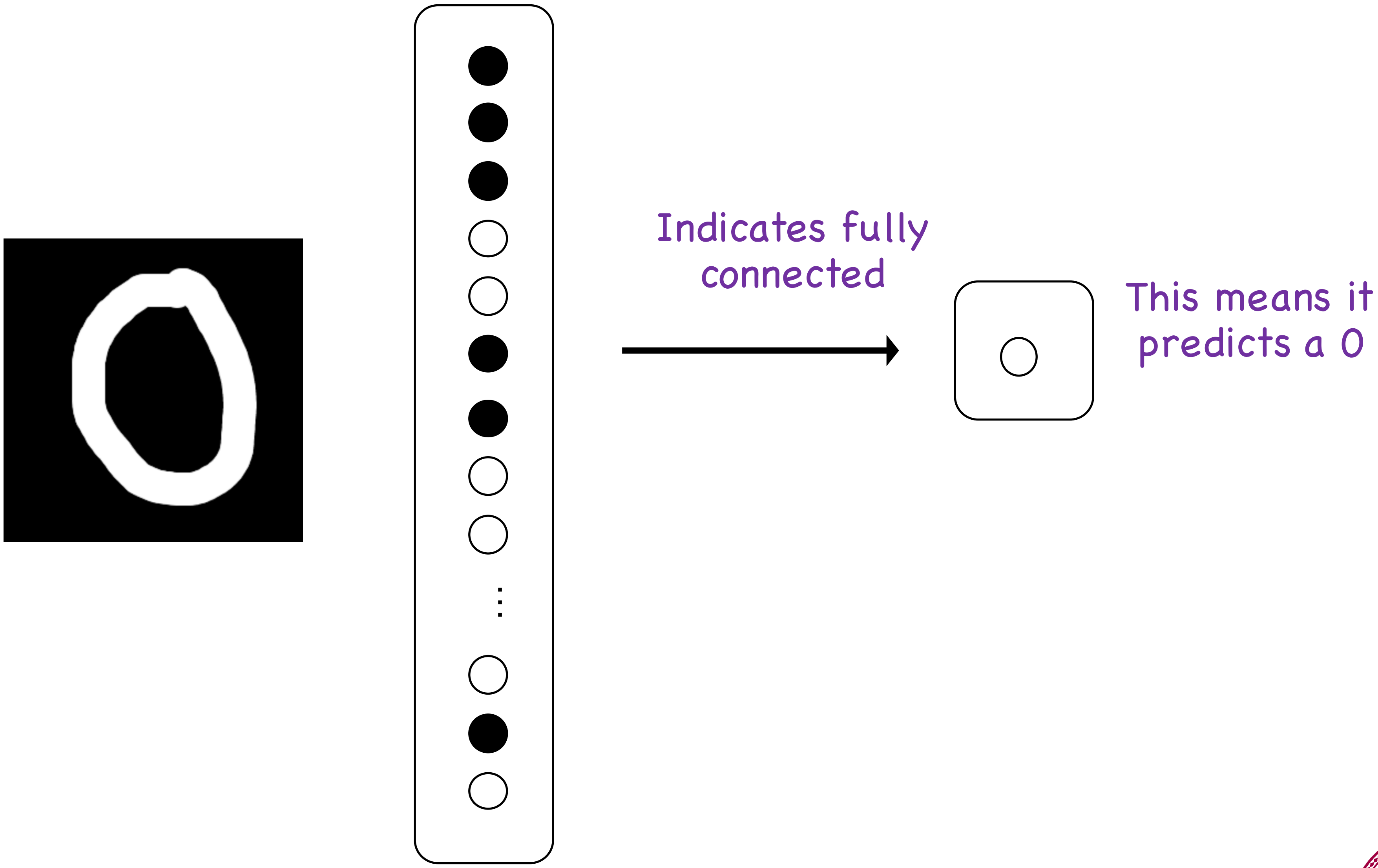
input $= [0, 0, 1, 1, \ldots, 0, 1, 1, 0, \ldots 0, 1, 0, 0]$

label $= 1$

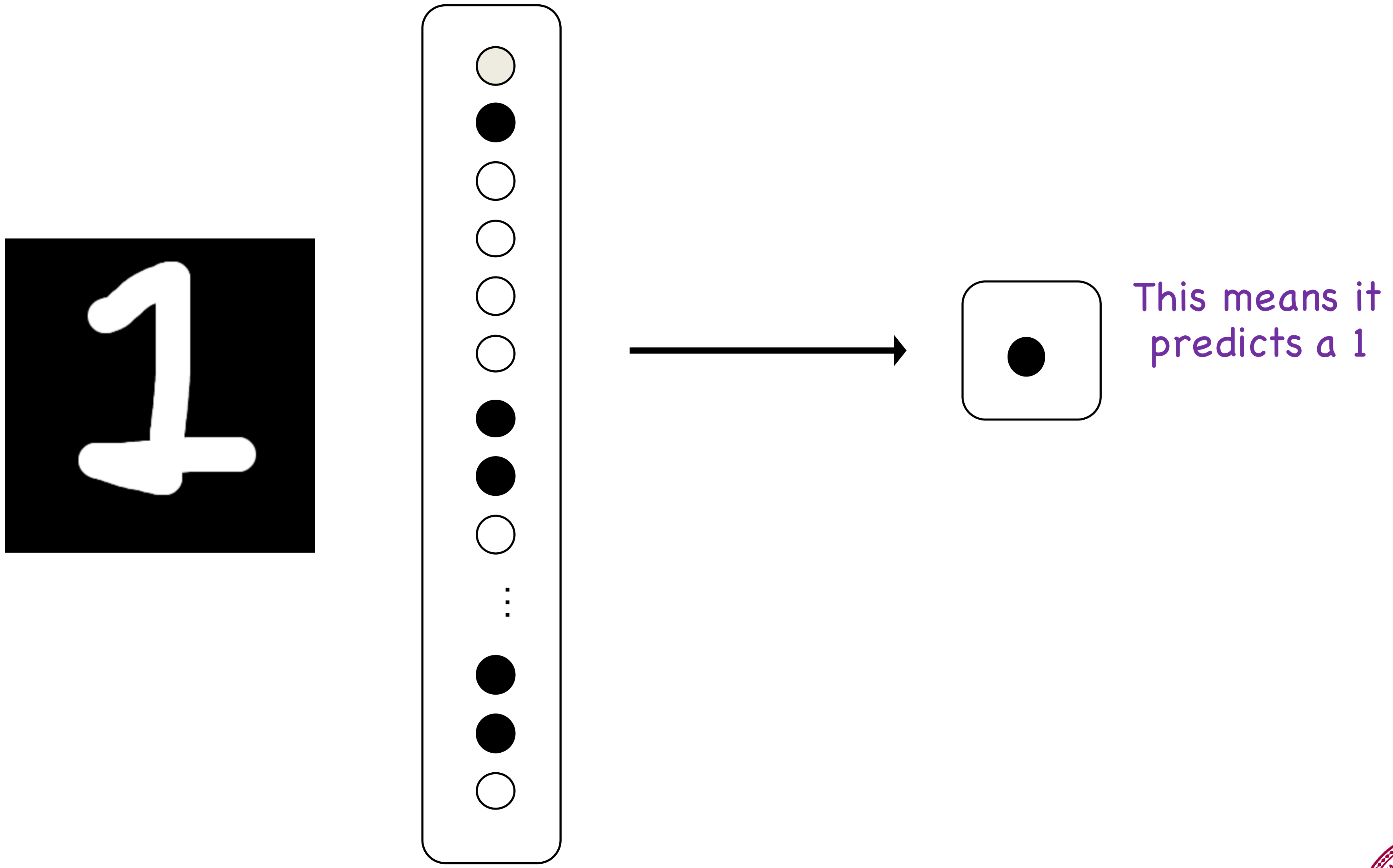
Single Neuron



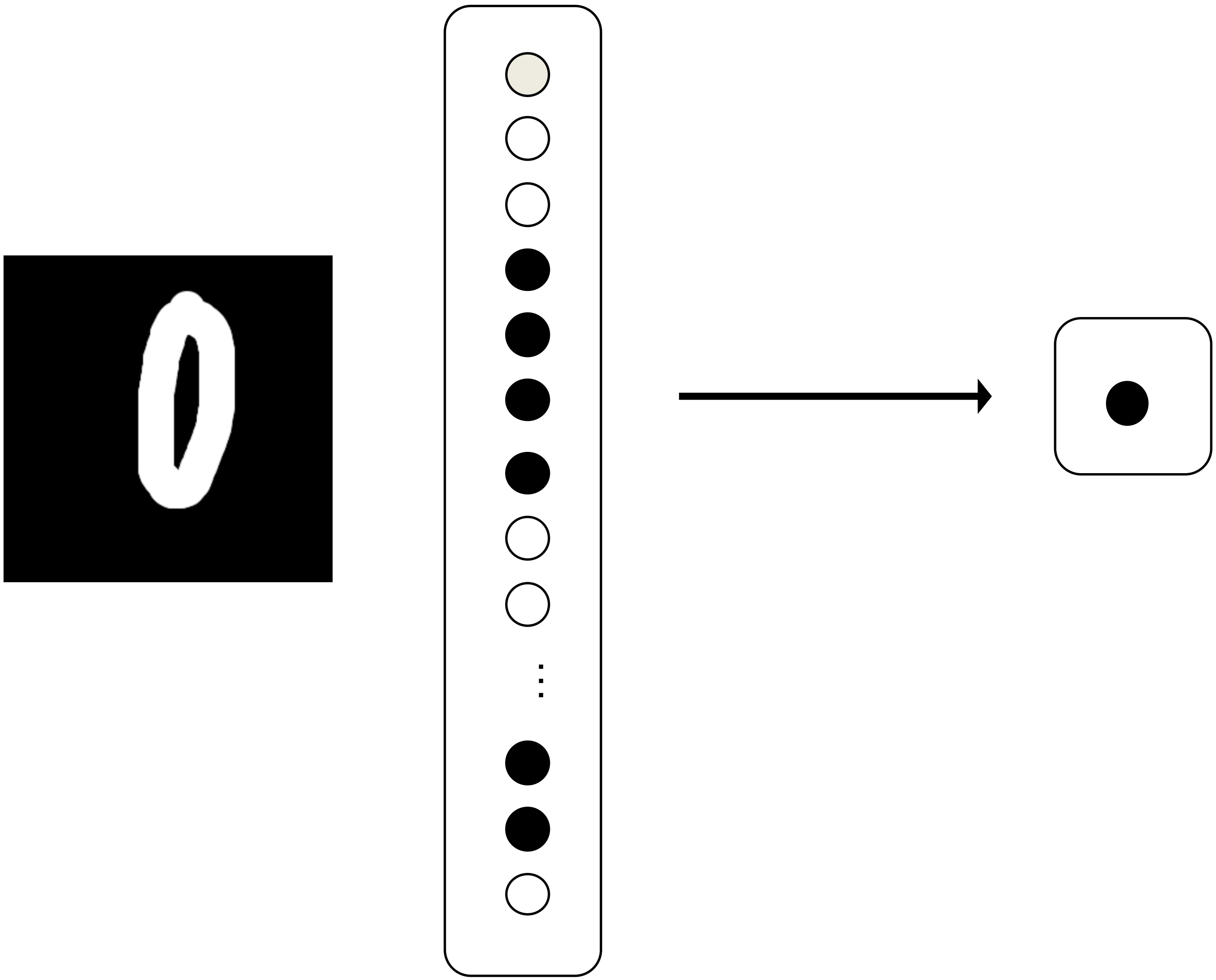
Single Neuron



Single Neuron



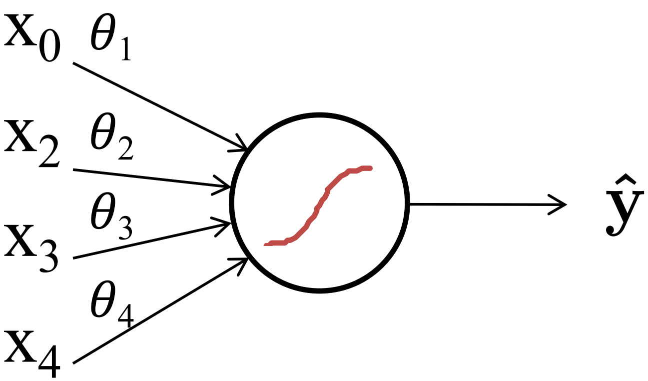
Not So Good



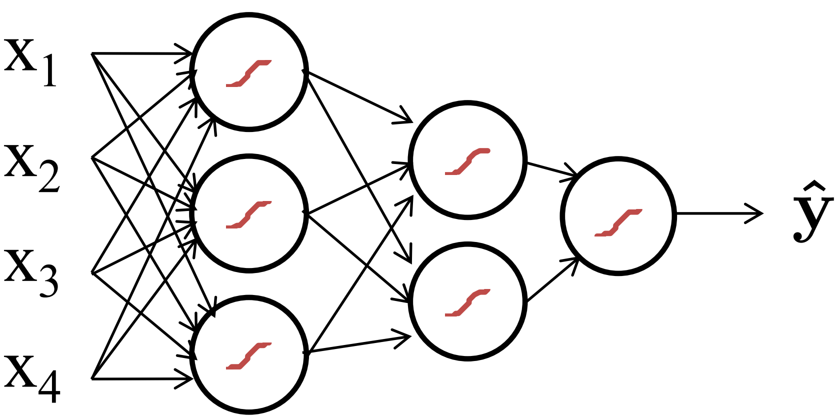
This means it predicts a 1

Logistic Regression and Neural Networks

- Single Neuron:

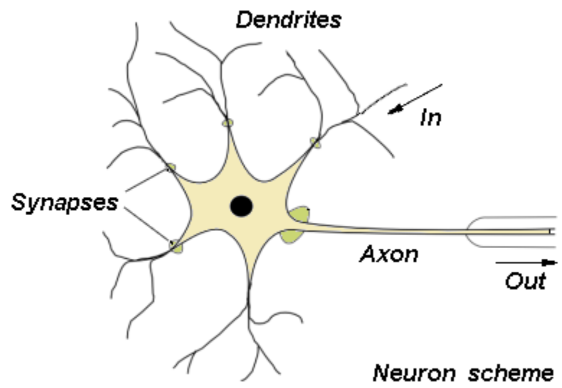


- Neural network

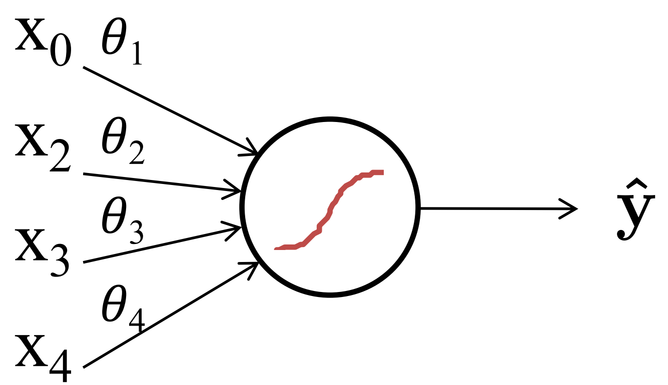


Biological Basis for Neural Networks

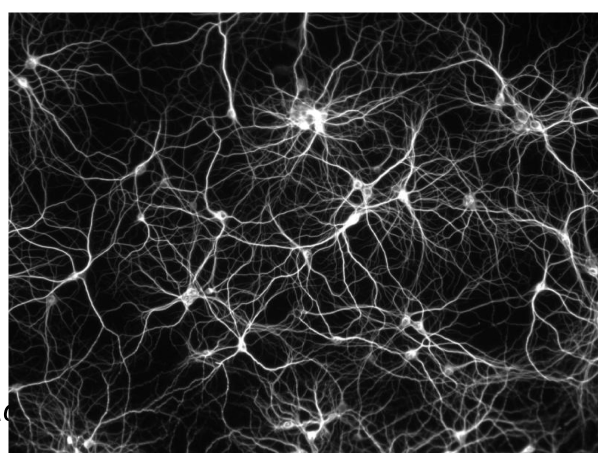
A neuron



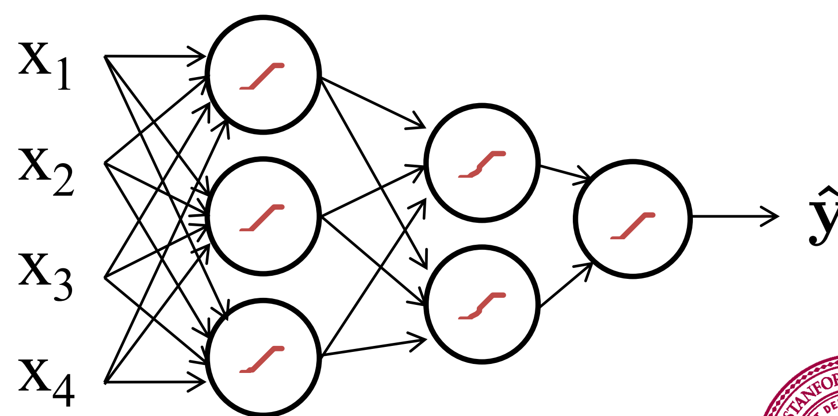
Artificial Neuron



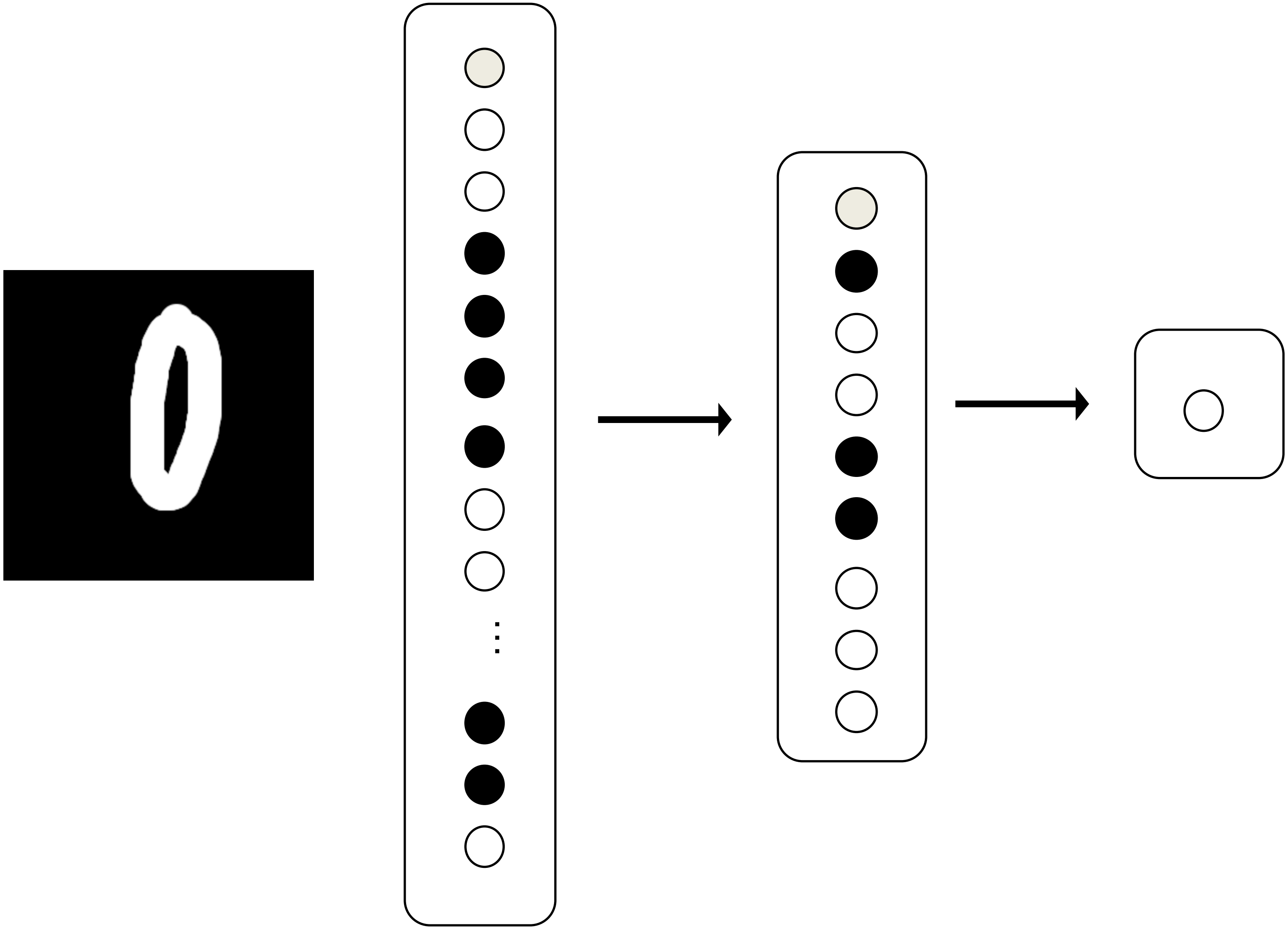
Your brain



Neural Network

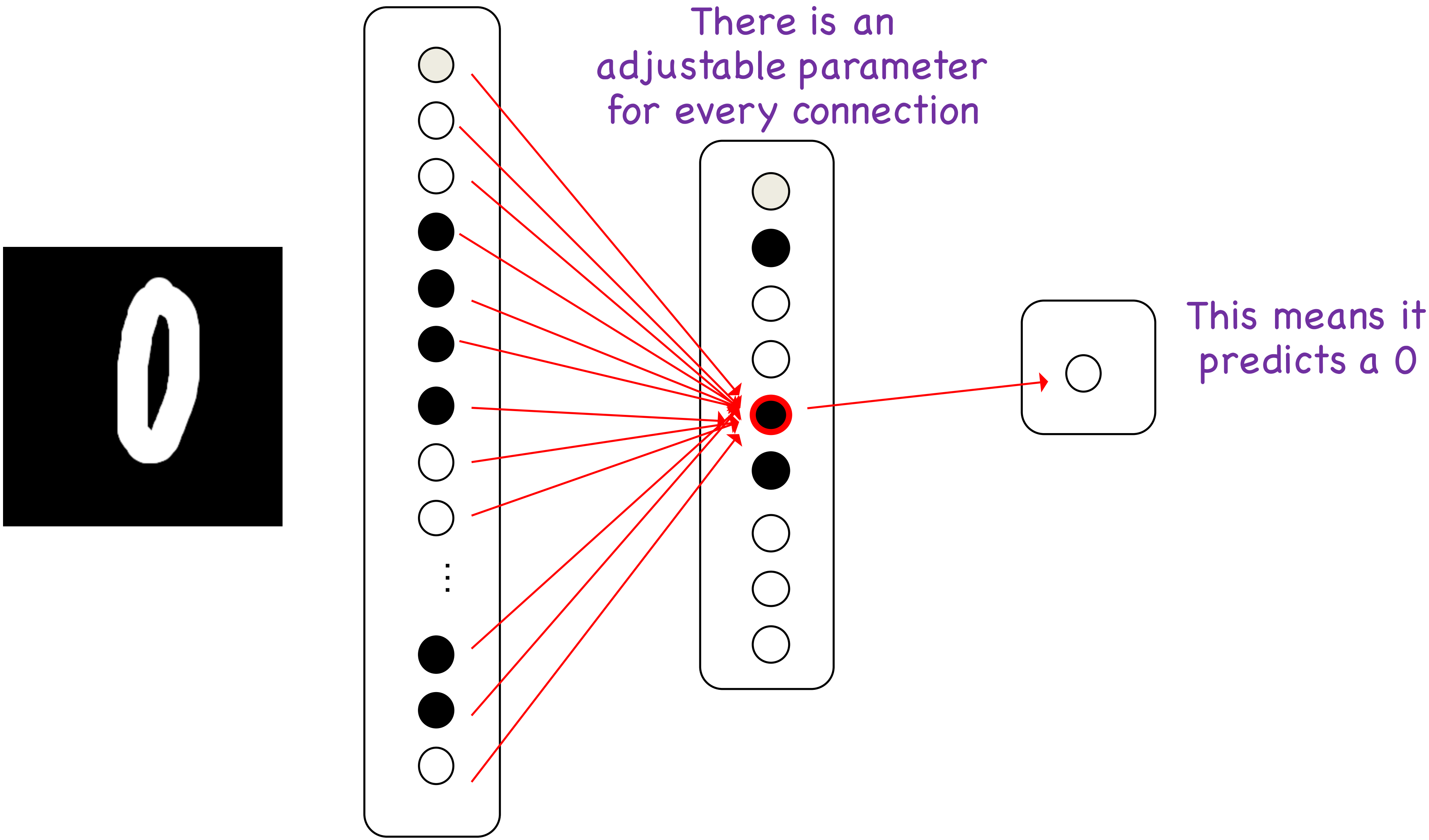


We Can Put Neurons Together



This means it predicts a 0

We Can Put Neurons Together



Look at a single "hidden" neuron

What Does This Look Like in Code?

workspace - Java - Neuron/Neuron.java - Eclipse

NeuralNetwork.j | Tank.java | FileWriter.clas | FileOutputStrea

```java
public class NeuralNetwork extends ConsoleProgram{

    private static final int N_INPUTS = 1024;
    private static final int N_LAYER1 = 20;

    private ArrayList<Neuron> layer1 = null;
    private Neuron prediction = null;

    public void run() {
        loadNeuralNetwork();

        // make predicitons
        GImage birdImage = new GImage("bird6.png");
        GImage planeImage = new GImage("airplane4.png");

        makePrediction(birdImage);
        makePrediction(planeImage);
    }

    private void makePrediction(GImage img) {
        // turn the image into inputs
        ArrayList<Double> inputs = new ArrayList<Double>();
        int[][] pixelArray = img.getPixelArray();
        for(int r = 0; r < pixelArray.length; r++) {
            for(int c = 0; c < pixelArray[0].length; c++) {
                Color color = new Color(pixelArray[r][c]);
                double greyScale = getGrey(color);
                inputs.add(greyScale);
            }
```

Fish.java | AquariumSimulator.java | Neuron.java

```java
public class Neuron extends GraphicsProgram {

    private ArrayList<Double> weights = null;

    public Neuron(String fileName, int n) {
        loadWeightsFromFile(fileName, n);
    }

    public double activate(ArrayList<Double> inputs) {
        double weightedSum = 0.0;
        for(int i = 0; i < inputs.size(); i++) {
            weightedSum += inputs.get(i) * weights.get(i);
        }
        return sigmoid(weightedSum);
    }

    private double sigmoid(double x) {
        return 1.0 / (1.0 + Math.exp(-x));
    }

    private void loadWeightsFromFile(String fileName, int
        weights = new ArrayList<Double>();
        try {
            BufferedReader rd = new BufferedReader(new Fil
            while(true) {
                String line = rd.readLine();
                if(line == null) break;
                weights.add(Double.parseDouble(line));
            }
```

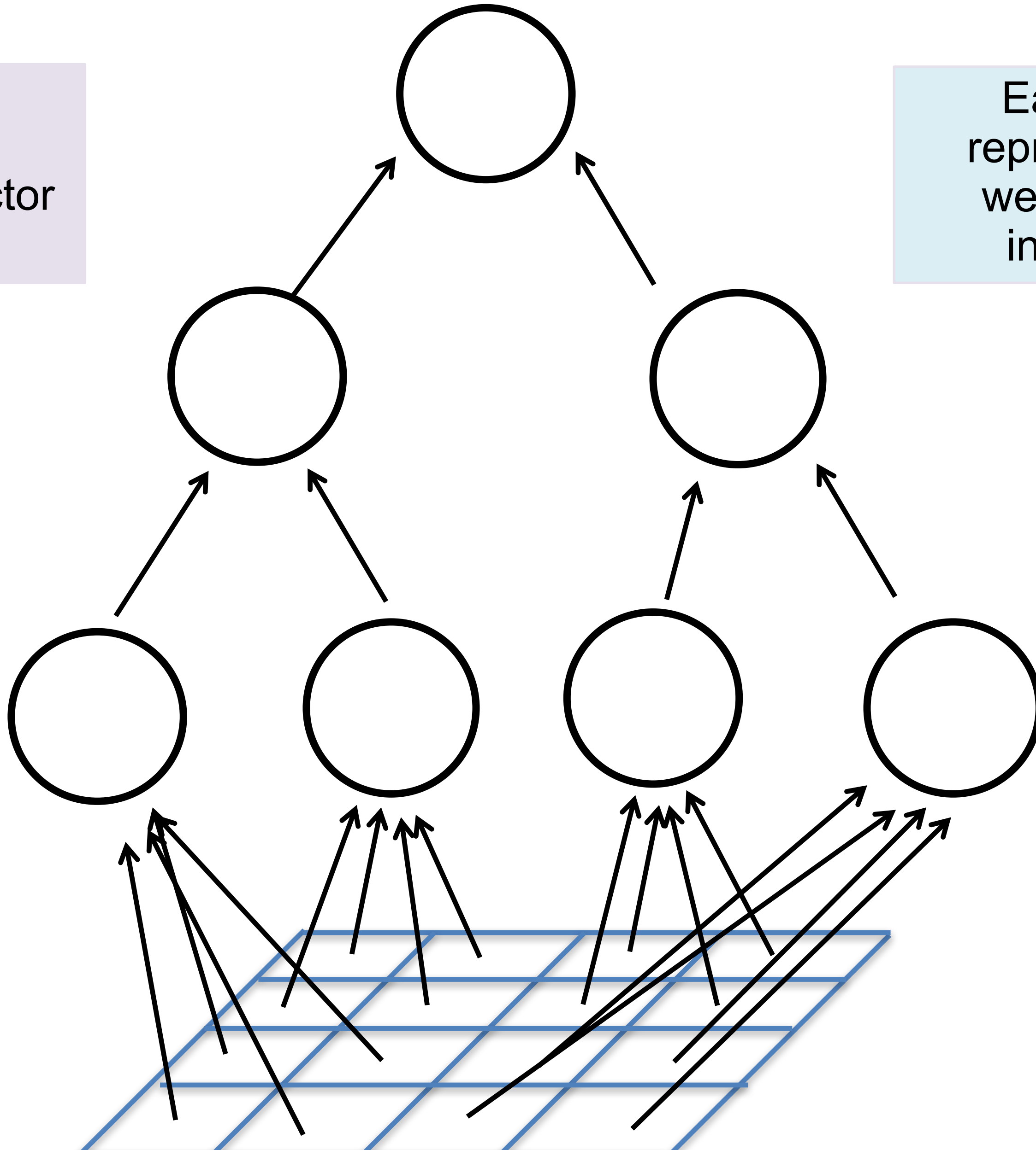
Writable | Smart Insert | 15 : 1

Aside: decomposition

How do we get those weights?

Neural Network

Each node represents a neuron (or a vector of neurons)



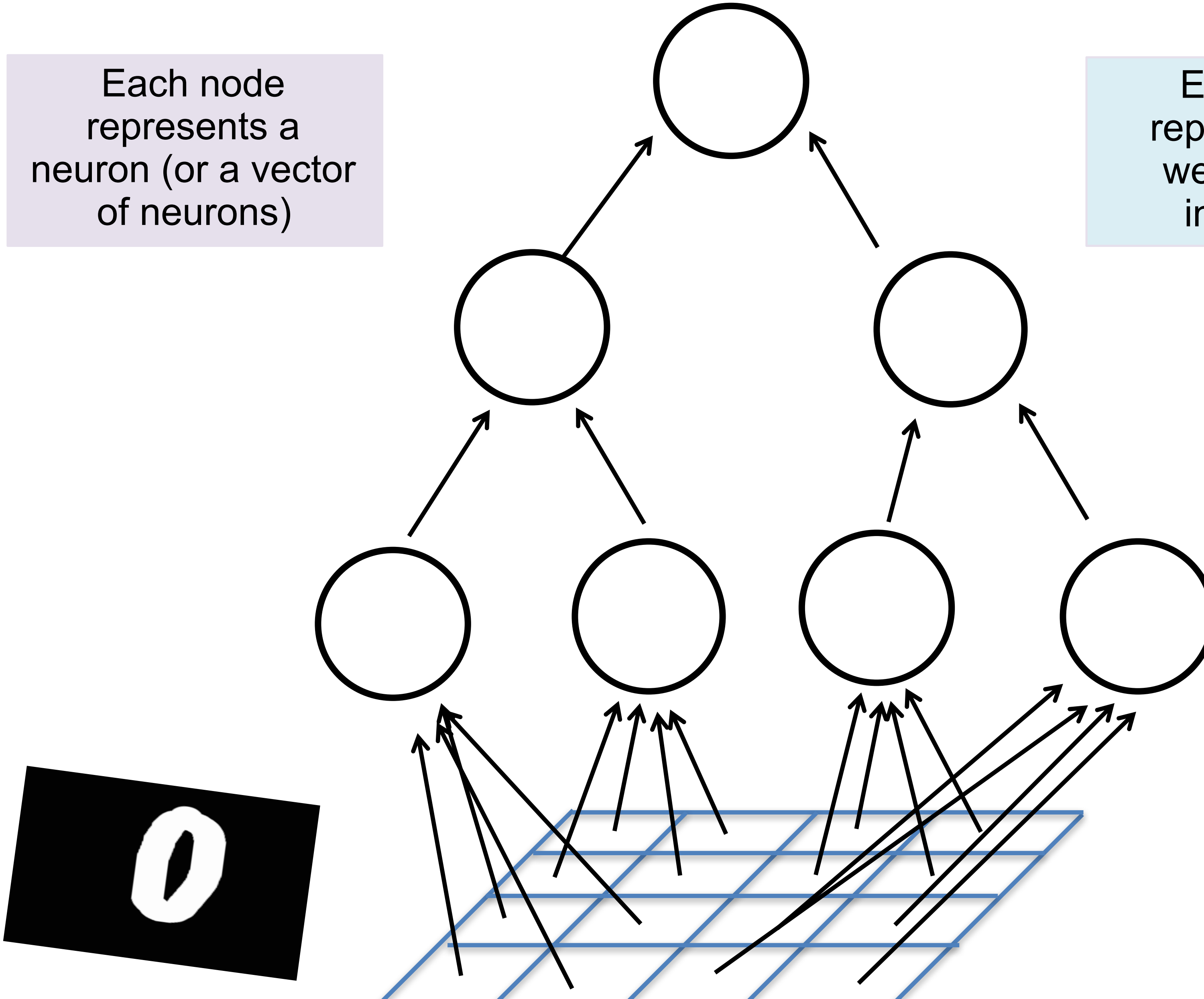
Each edge represents the weight of the interaction

Forward Pass…

Forward Pass

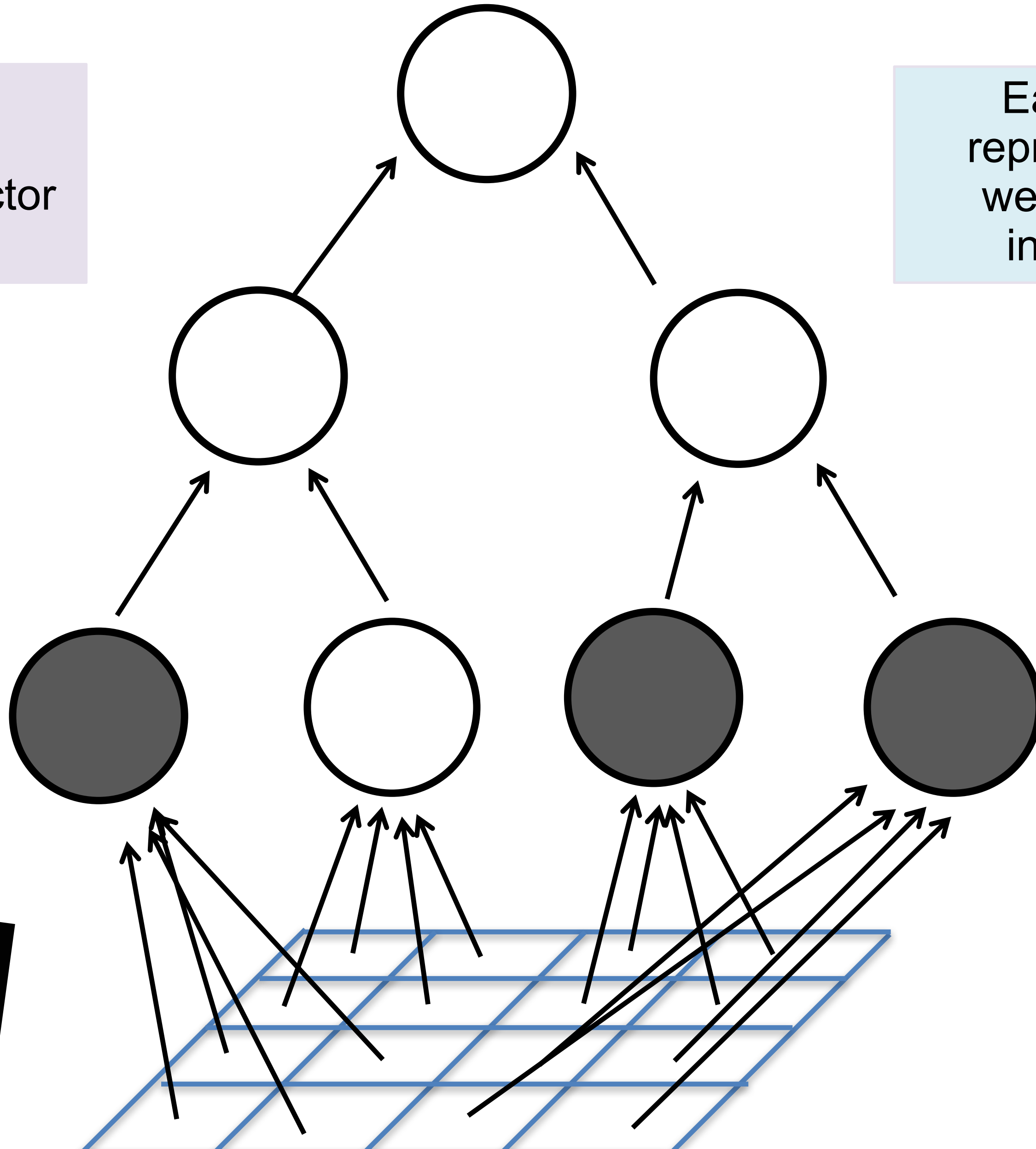
Each node represents a neuron (or a vector of neurons)

Each edge represents the weight of the interaction

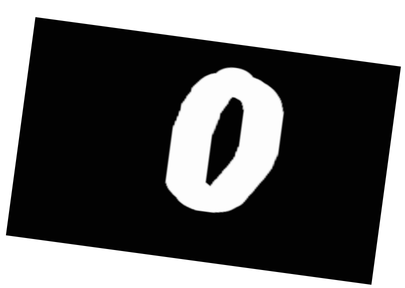


Forward Pass

Each node represents a neuron (or a vector of neurons)



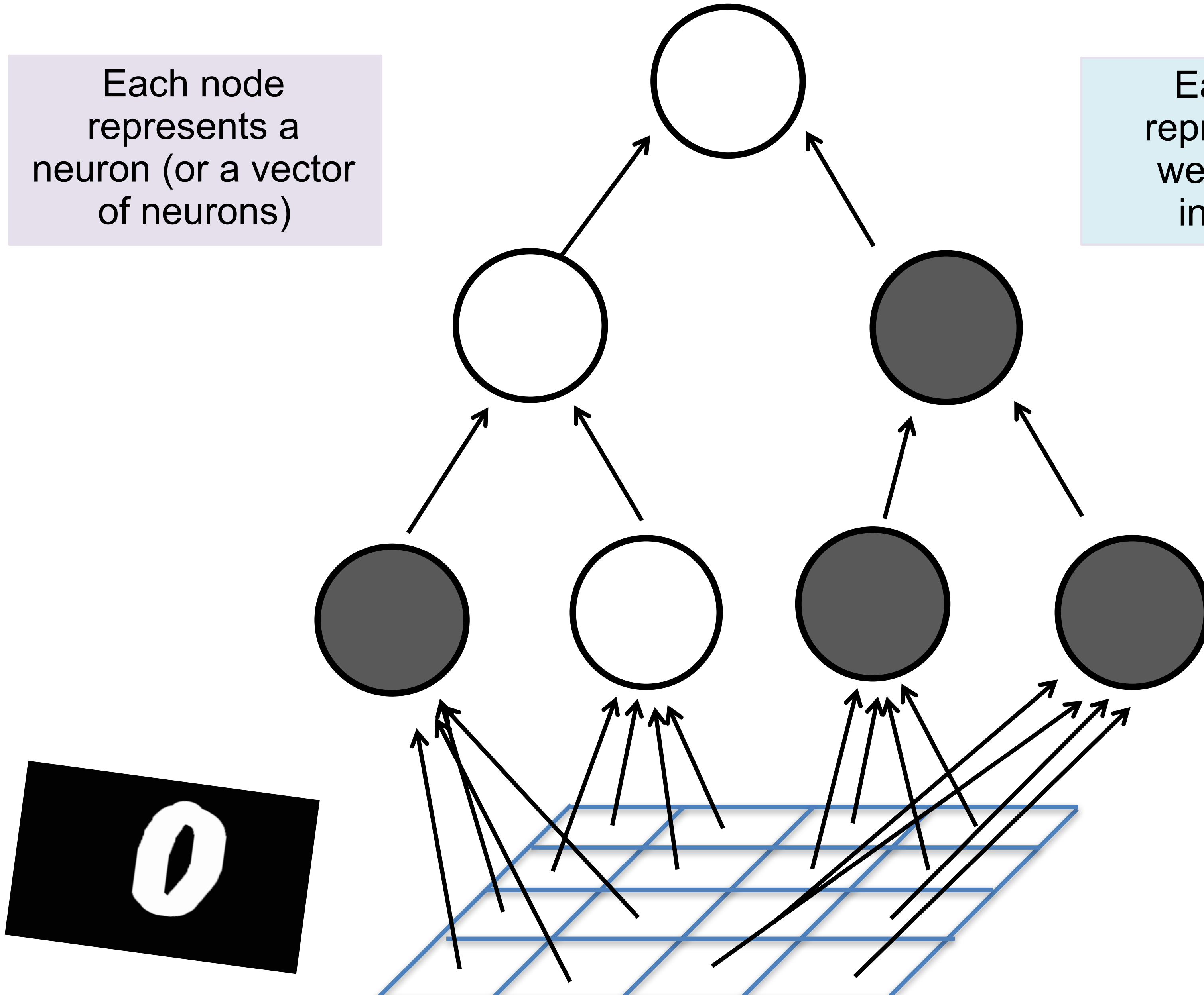
Each edge represents the weight of the interaction



Forward Pass

Each node represents a neuron (or a vector of neurons)

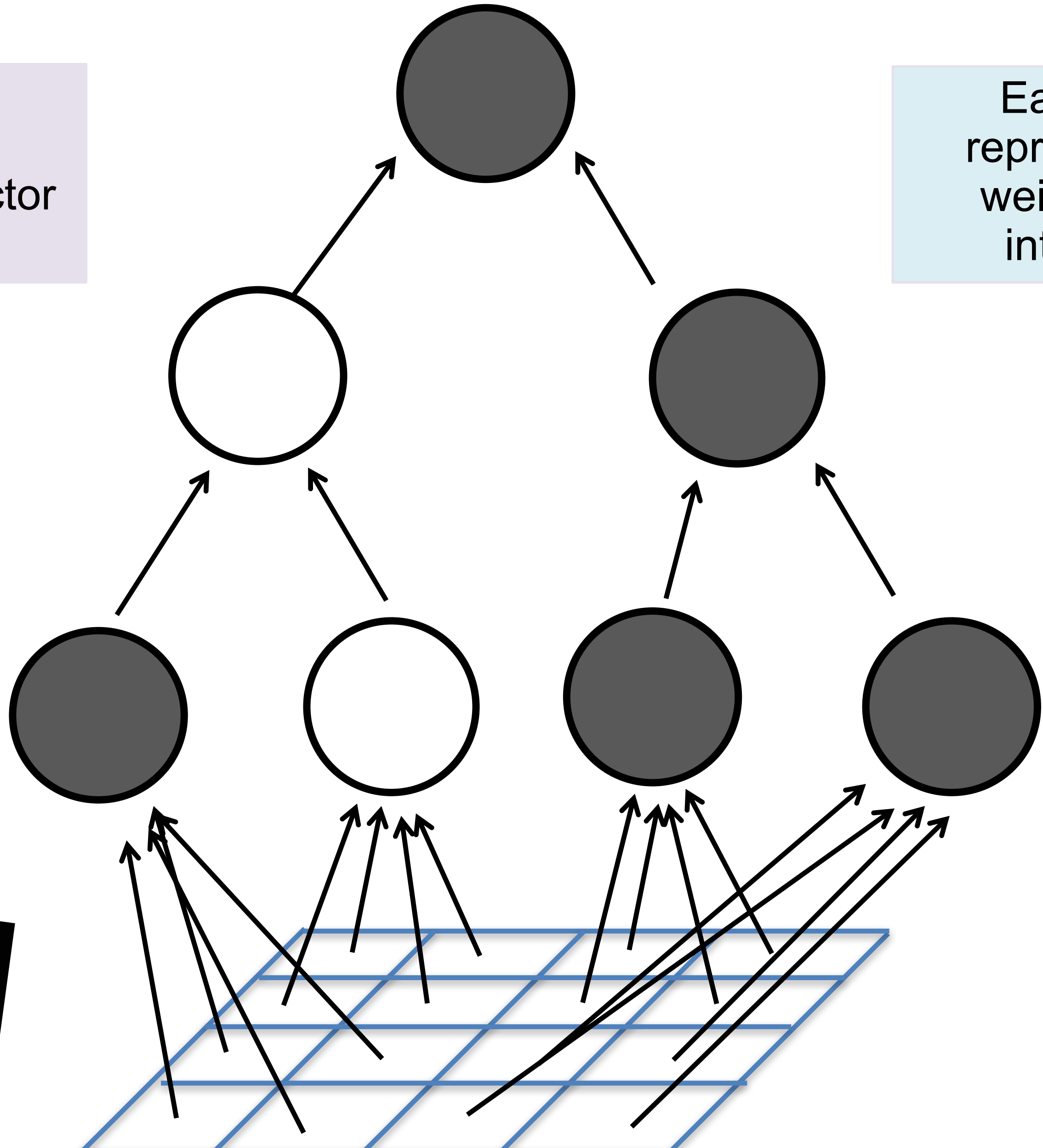
Each edge represents the weight of the interaction

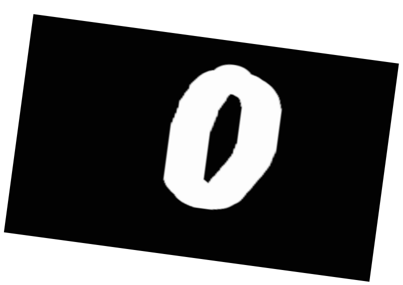


Forward Pass

Each node represents a neuron (or a vector of neurons)

Each edge represents the weight of the interaction

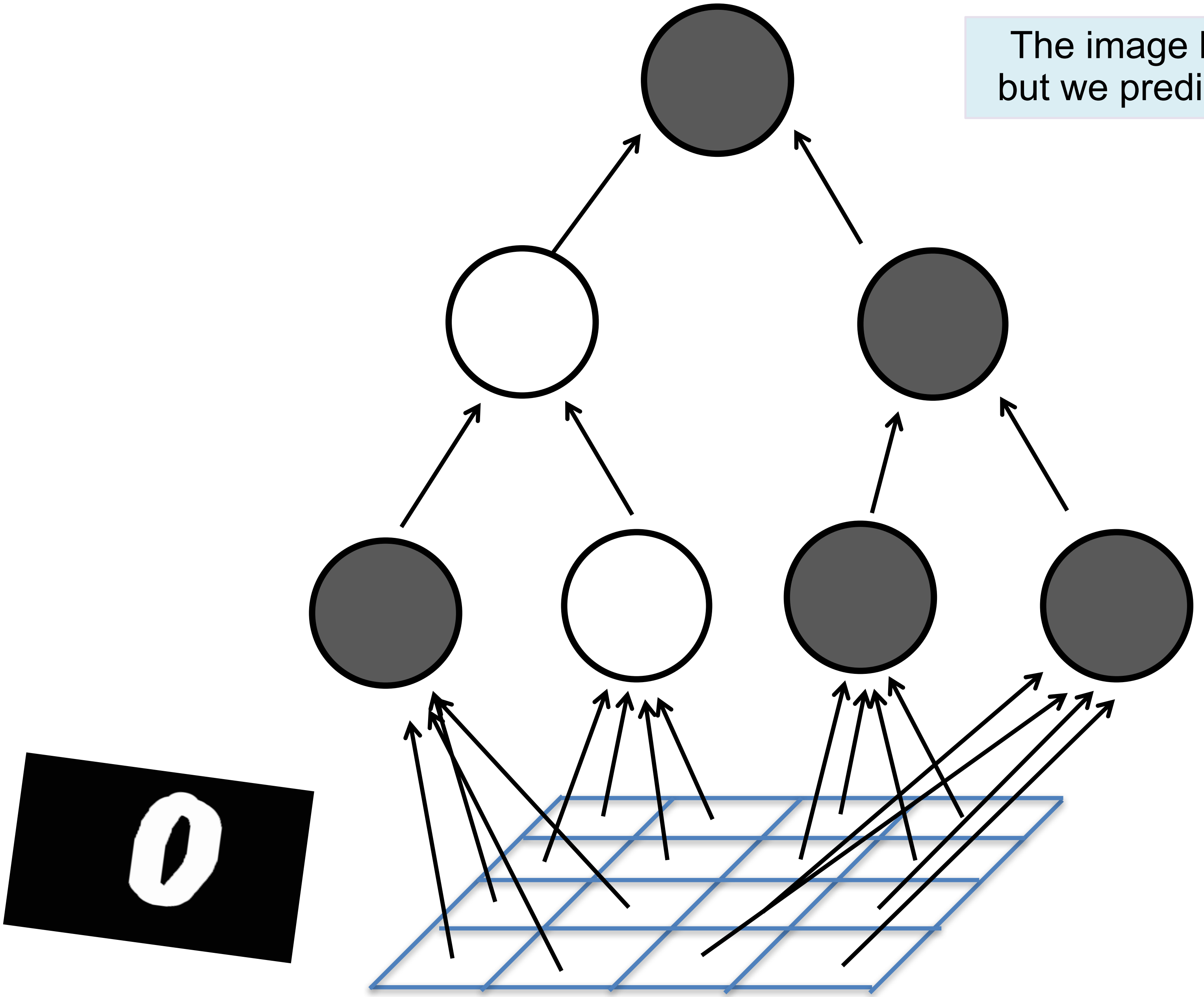




Backward Pass…

Backward Pass

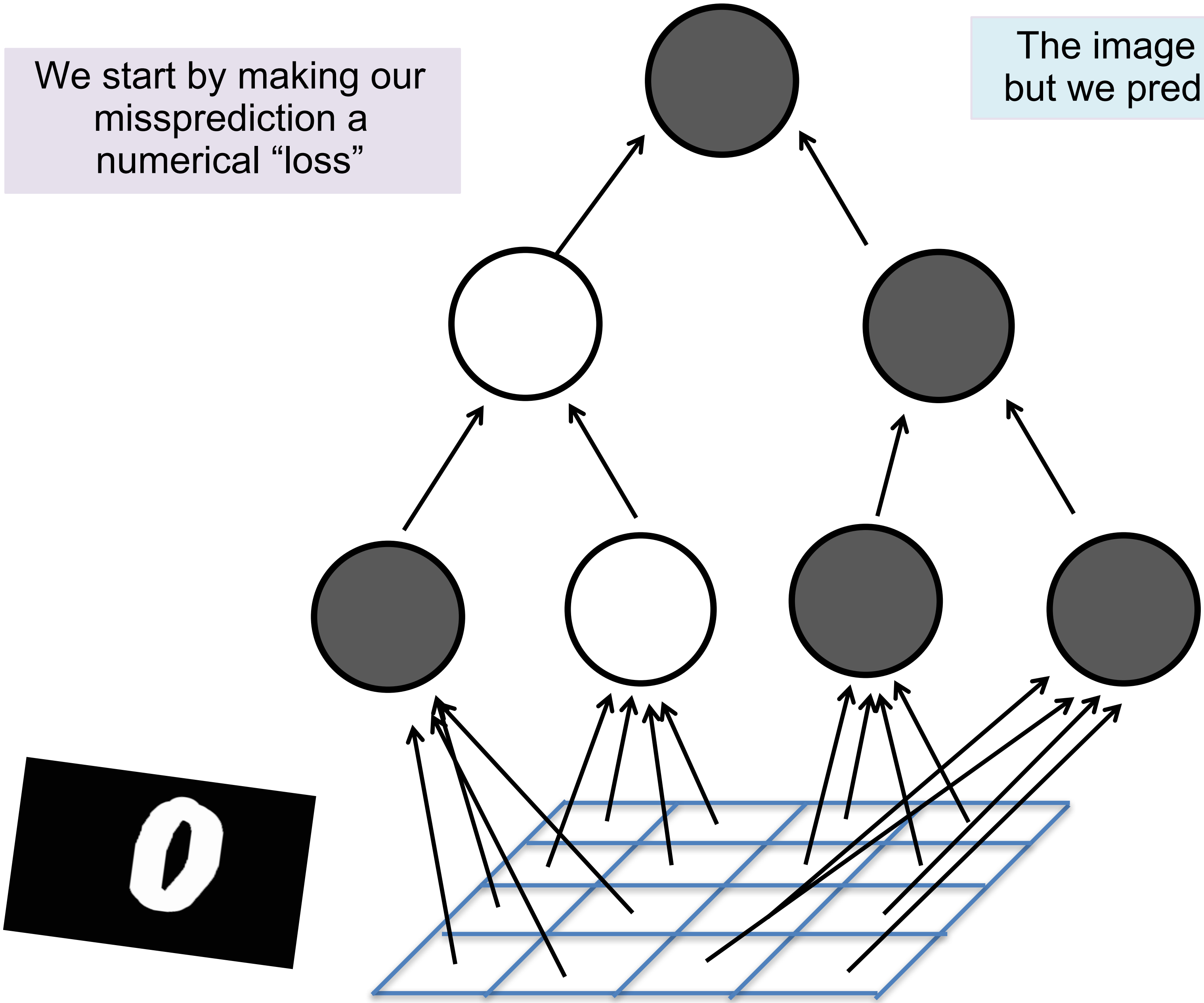
The image had a 0
but we predicted a 1



Backward Pass

We start by making our missprediction a numerical "loss"

The image had a 0 but we predicted a 1



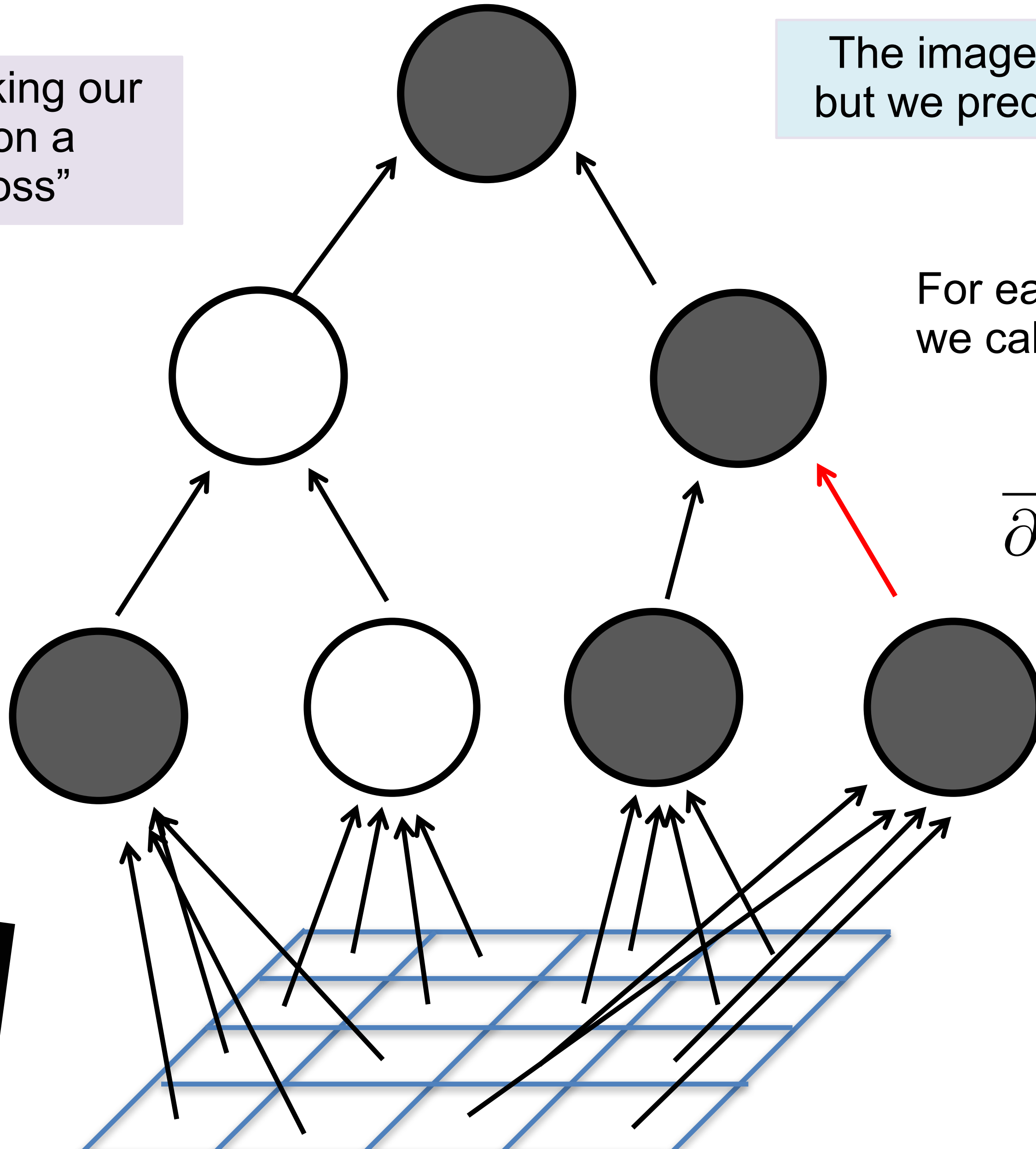
Backward Pass

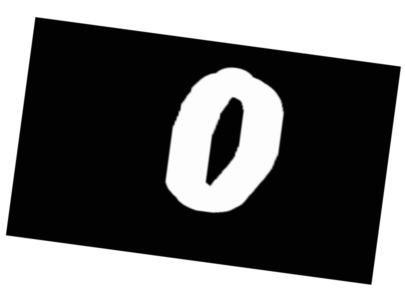
We start by making our misspredication a numerical "loss"

The image had a 0 but we predicted a 1

For each edge weight we calculate

$$\frac{\partial \text{Loss}}{\partial \text{EdgeWeight}}$$





Gradient of output layer params

$$\frac{\partial LL(\theta)}{\partial \theta_i^{(\hat{y})}} = \frac{\partial LL}{\partial \hat{y}} \cdot \frac{\partial \hat{y}}{\partial \theta_i^{(\hat{y})}}$$

$$\hat{y} = \sigma\left(\sum_{j=0}^{m_h} \mathbf{h}_j \theta_j^{(\hat{y})}\right)$$

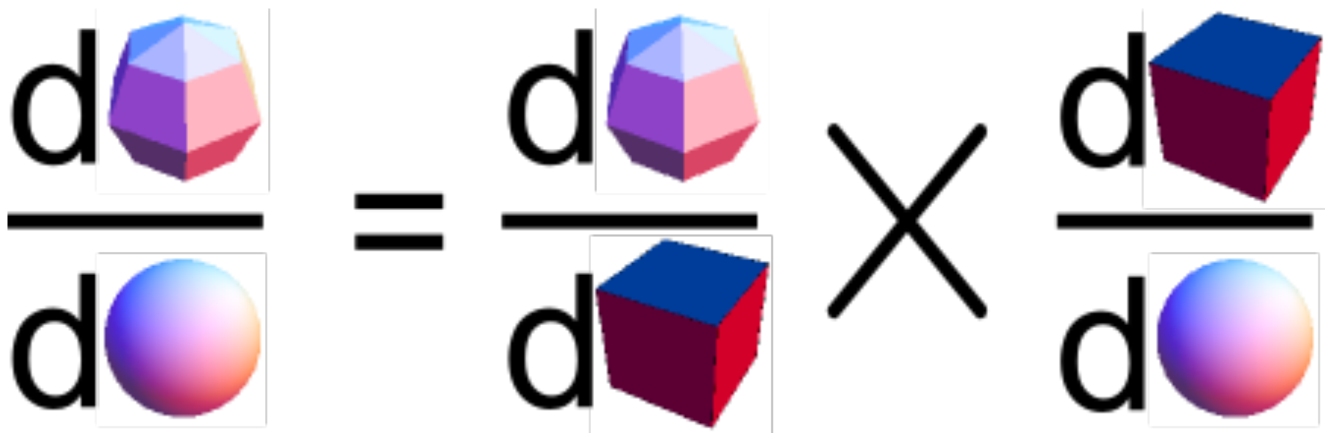
$$\frac{\partial \hat{y}}{\partial \theta_i^{(\hat{y})}} = \sigma\left(\sum_{j=0}^{m_h} \mathbf{h}_j \theta_j^{(\hat{y})}\right)\left[1 - \sigma\left(\sum_{j=0}^{m_h} \mathbf{h}_j \theta_j^{(\hat{y})}\right)\right] \cdot \frac{\partial}{\partial \theta_i^{(\hat{y})}} \sum_{j=0}^{m_h} \mathbf{h}_j \theta_j^{(\hat{y})}$$

$$= \hat{y}[1 - \hat{y}] \cdot \frac{\partial}{\partial \theta_i^{(\hat{y})}} \sum_{j=0}^{m_h} \mathbf{h}_j \theta_j^{(\hat{y})}$$

$$= \hat{y}[1 - \hat{y}] \cdot h_i$$

That looks scarier than it is

Chain Rule Down the Network



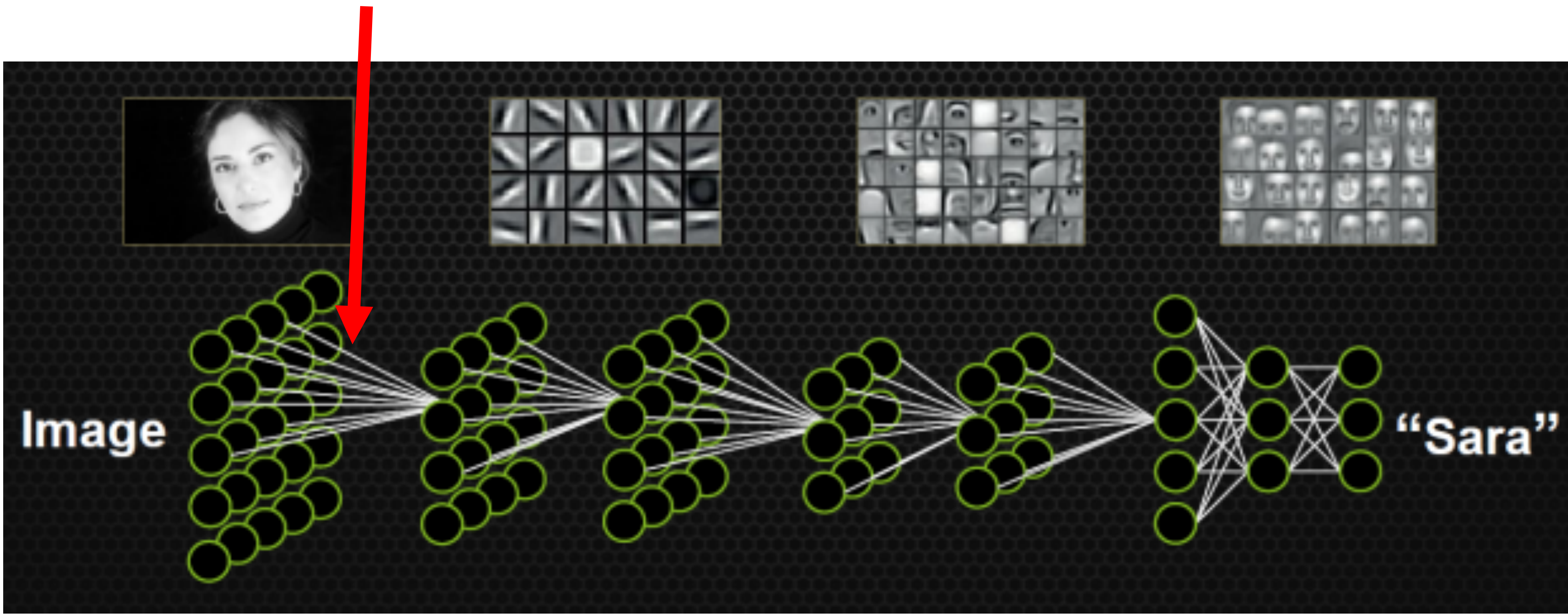
Artificial Neurons: One of the greatest decompositions of our lifetimes

```
model.calculatePartialDerivative(data)
```

```
model.update(data)
```

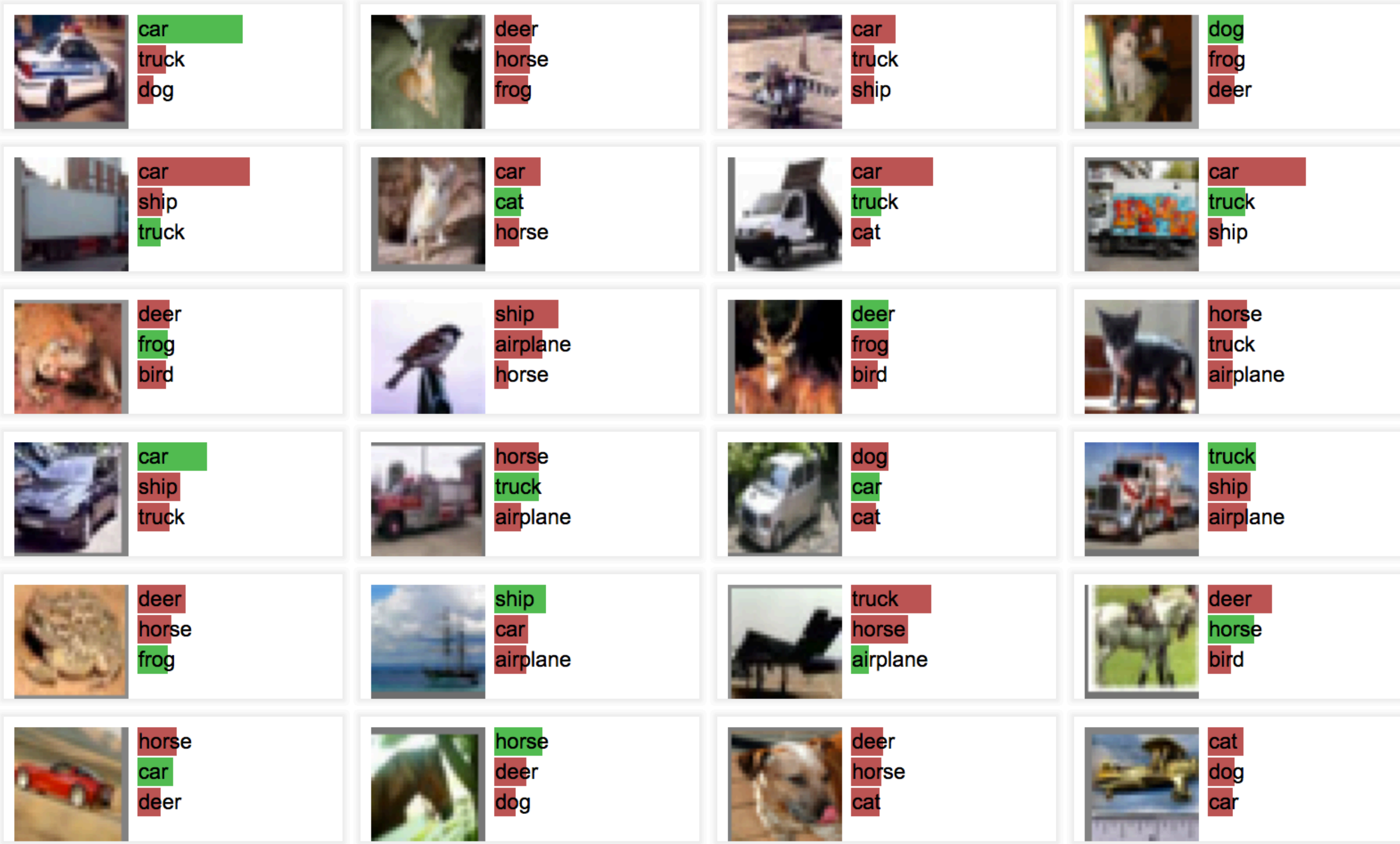
Works for any number of layers

Weight between two neurons



Let's Train!

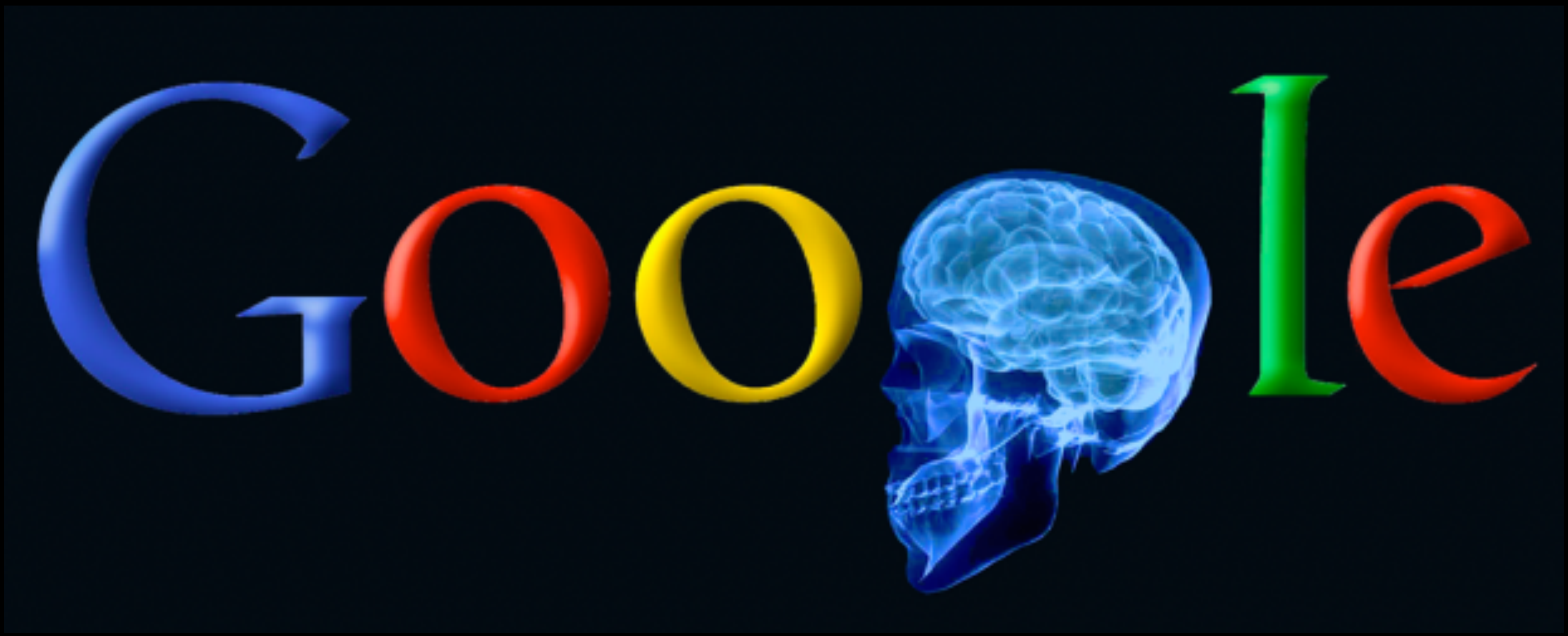
test accuracy based on last 200 test images: 0.2894736842105263



http://cs.stanford.edu/people/karpathy/convnetjs/demo/classify2d.html

Like lego pieces

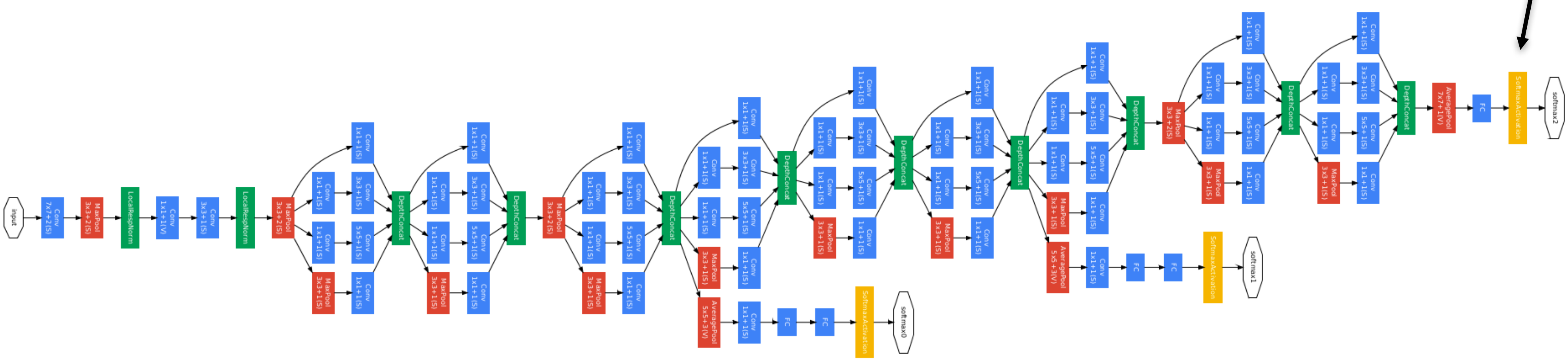
GoogLeNet Brain



1 Trillion Artificial Neurons

GoogLeNet Brain Graph

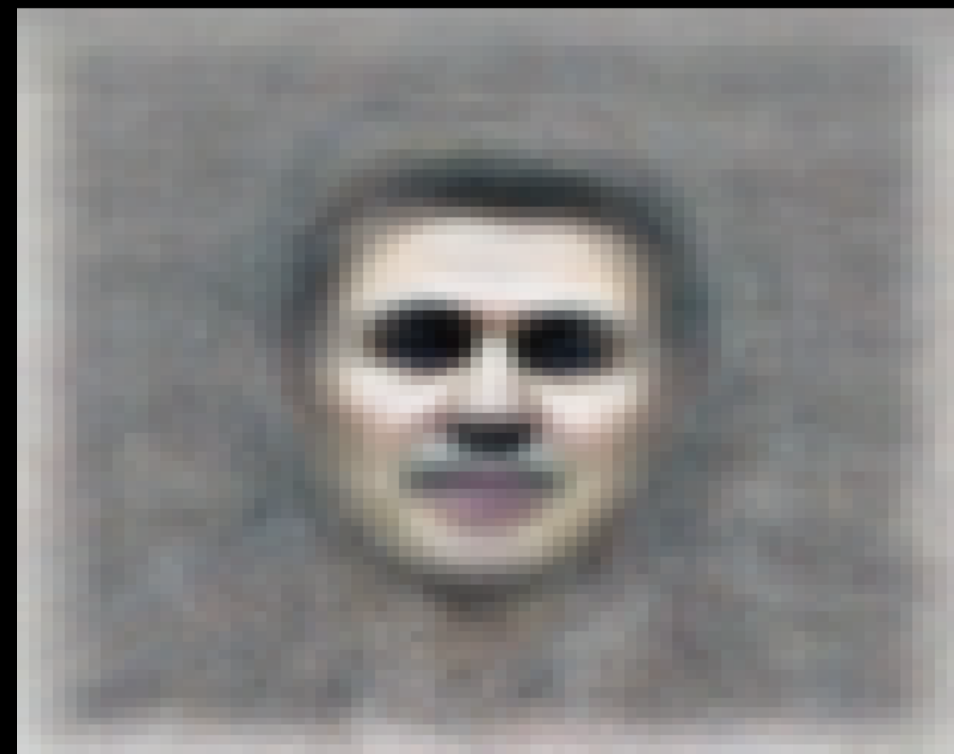
Multiple,
Multi class output



22 layers deep

The Face Neuron

Top stimuli from the test set



Optimal stimulus
by numerical optimization

Le, et al., *Building high-level features using large-scale unsupervised learning*. ICML 2012

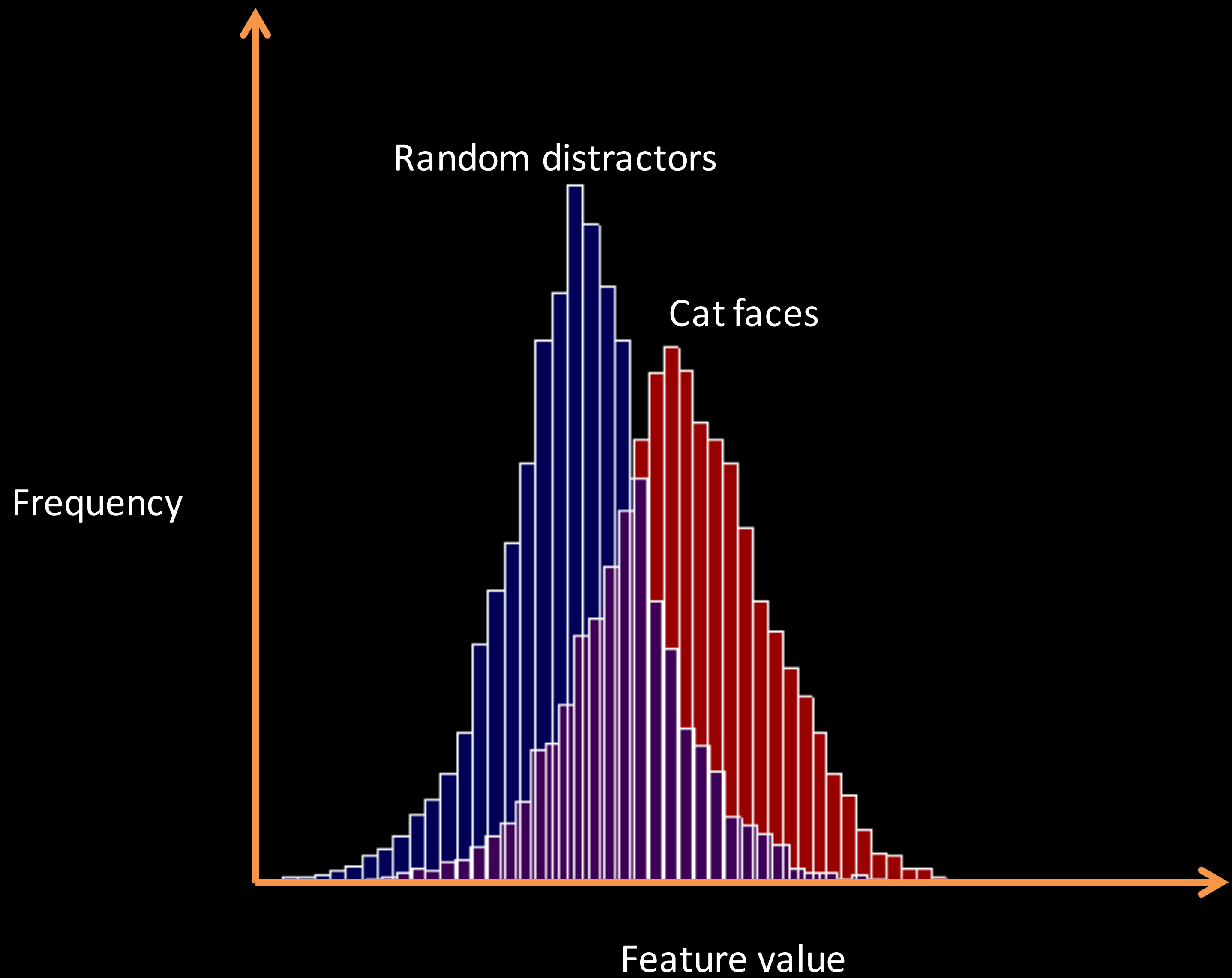
The Cat Neuron

Top stimuli from the test set

Optimal stimulus
by numerical optimization

Le, et al., *Building high-level features using large-scale unsupervised learning*. ICML 2012

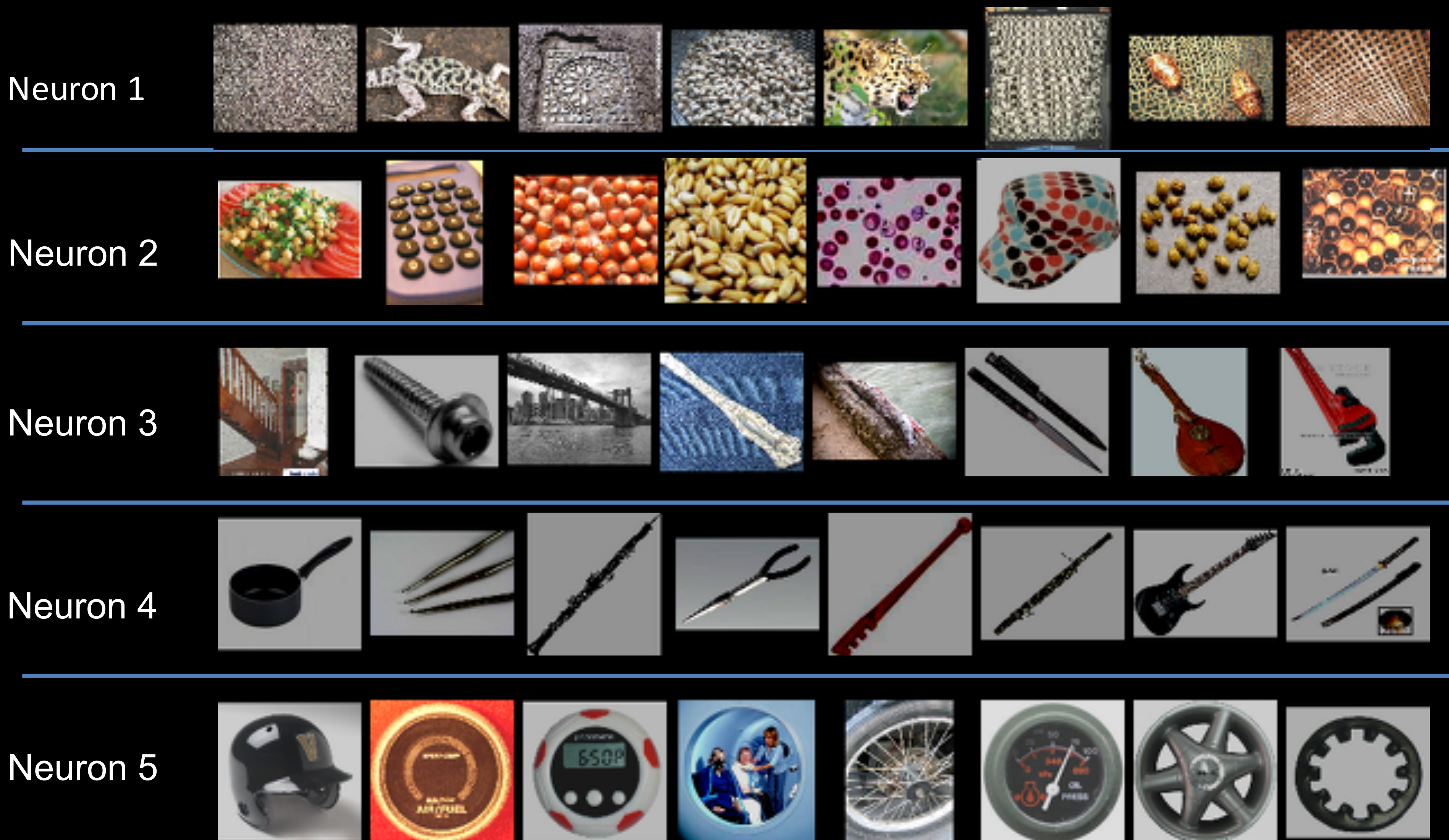
The Cat Neuron



Le, et al., *Building high-level features using large-scale unsupervised learning*. ICML 2012


Hire the smartest people in the world
Invent cat detector

Best Neuron Stimuli



Le, et al., *Building high-level features using large-scale unsupervised learning*. ICML 2012

Best Neuron Stimuli

Neuron 6

Neuron 7

Neuron 8

Neuron 9

Le, et al., *Building high-level features using large-scale unsupervised learning*. ICML 2012

Best Neuron Stimuli

Neuron 10

Neuron 11

Neuron 12

Neuron 13

Le, et al., *Building high-level features using large-scale unsupervised learning*. ICML 2012

ImageNet Classification

22,000 categories

14,000,000 images

Hand-engineered features (SIFT, HOG, LBP),
Spatial pyramid, SparseCoding/Compression

Le, et al., *Building high-level features using large-scale unsupervised learning*. ICML 2012

22,000 is a lot!

…
smoothhound, smoothhound shark, Mustelus mustelus
American smooth dogfish, Mustelus canis
Florida smoothhound, Mustelus norrisi
whitetip shark, reef whitetip shark, Triaenodon obseus
Atlantic spiny dogfish, Squalus acanthias
Pacific spiny dogfish, Squalus suckleyi
hammerhead, hammerhead shark
smooth hammerhead, Sphyrna zygaena
smalleye hammerhead, Sphyrna tudes
shovelhead, bonnethead, bonnet shark, Sphyrna tiburo
angel shark, angelfish, Squatina squatina, monkfish
electric ray, crampfish, numbfish, torpedo
smalltooth sawfish, Pristis pectinatus
guitarfish
roughtail stingray, Dasyatis centroura
butterfly ray
eagle ray
spotted eagle ray, spotted ray, Aetobatus narinari
cownose ray, cow-nosed ray, Rhinoptera bonasus
manta, manta ray, devilfish
Atlantic manta, Manta birostris
devil ray, Mobula hypostoma
grey skate, gray skate, Raja batis
little skate, Raja erinacea
…

Stingray

Mantaray

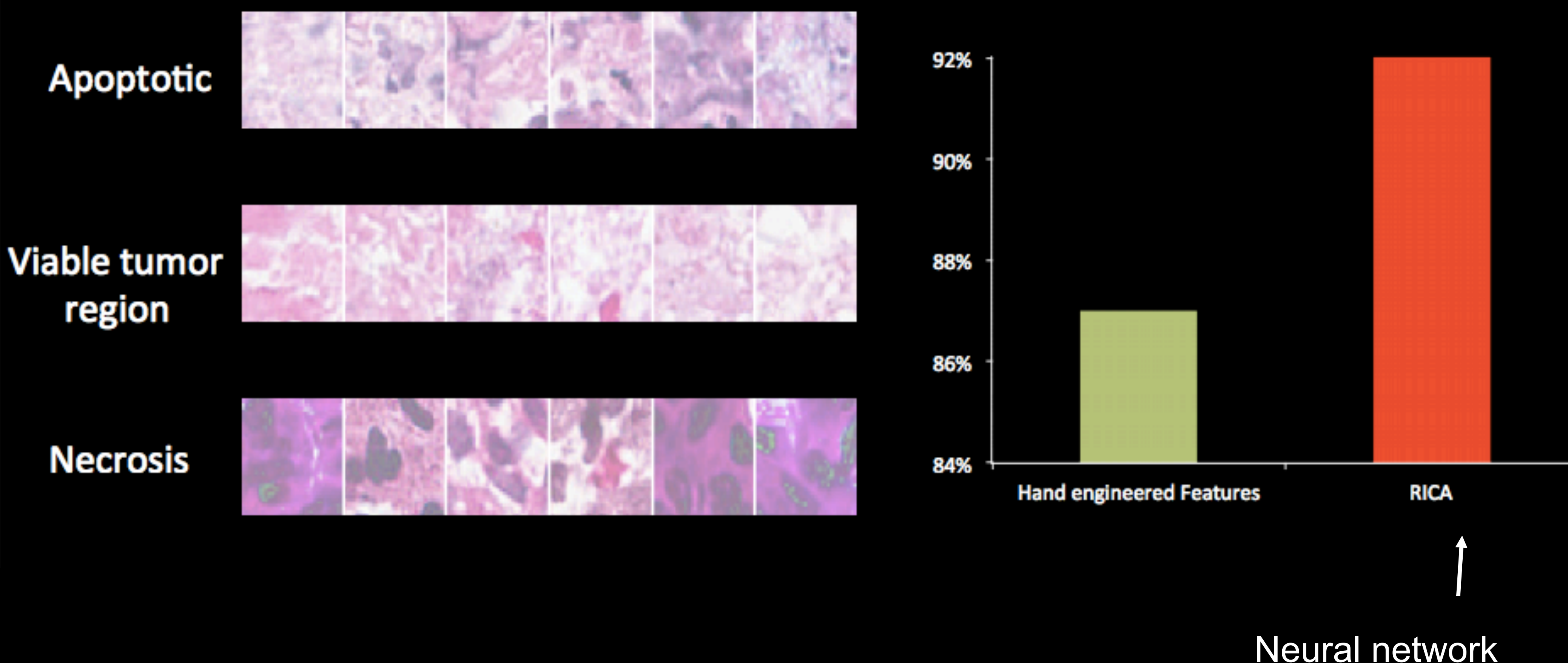
0.005%	1.5%	?
Random guess	Pre Neural Networks	GoogLeNet

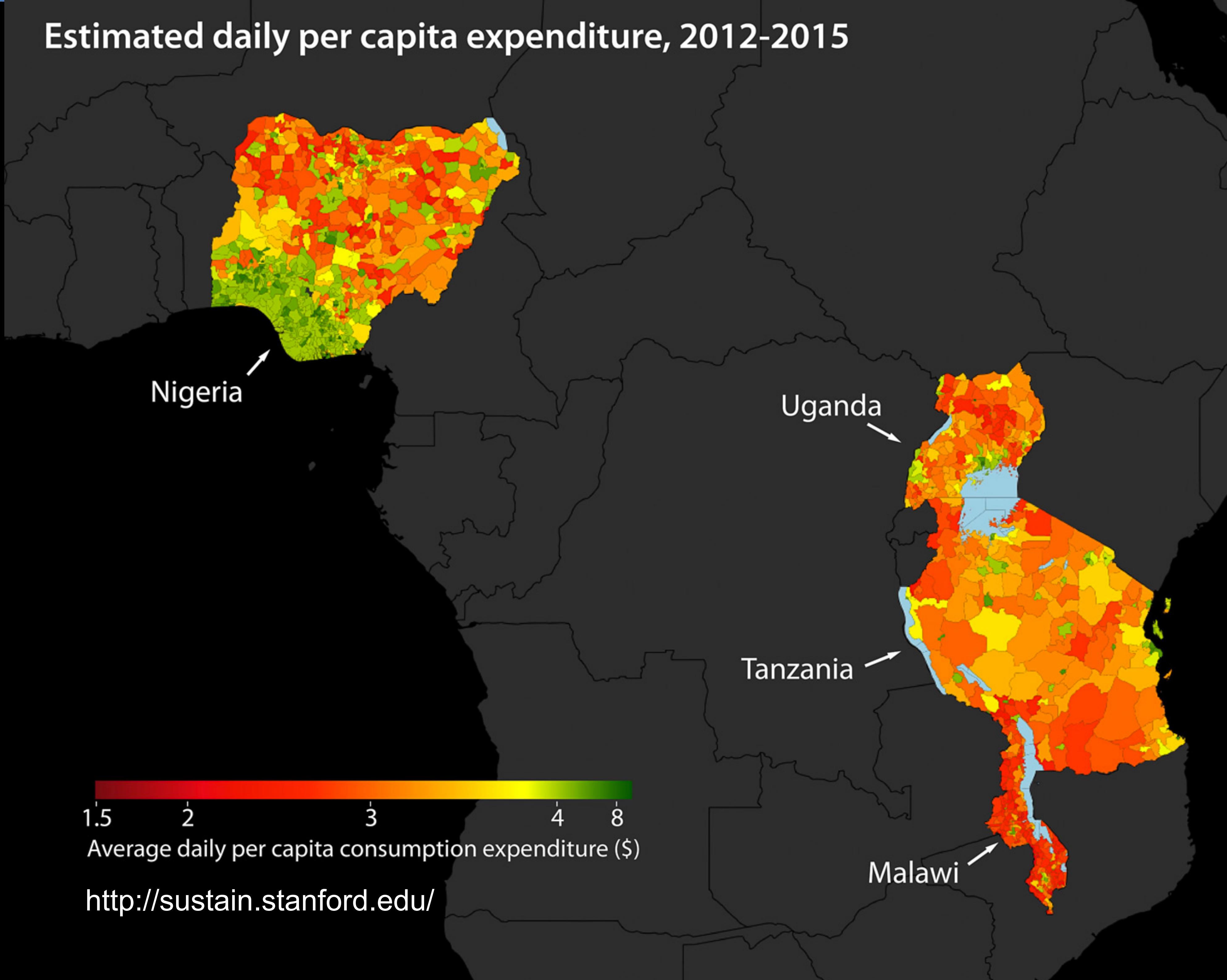
Le, et al., *Building high-level features using large-scale unsupervised learning*. ICML 2012

0.005%	1.5%	43.9%
Random guess	Pre Neural Networks	GoogLeNet

Szegedy et al, Going Deeper With Convolutions, CVPR 2015

Vision has Social Implications



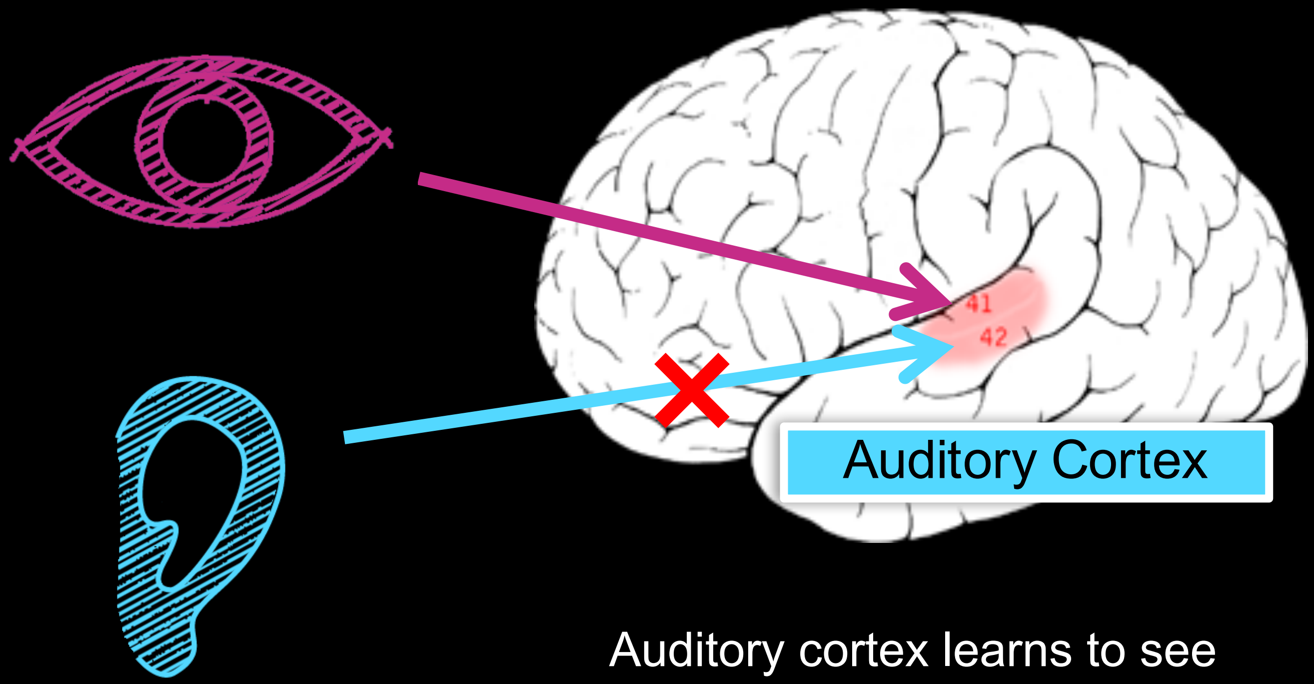

Estimated daily per capita expenditure, 2012-2015
Nigeria
Uganda
Tanzania
Malawi
1.5
2
3
4
8
Average daily per capita consumption expenditure ($)
http://sustain.stanford.edu/

One Algorithm Hypothesis

Much of perception in the brain can be explained with a single learning algorithm.

[Andrew Ng]

One Algorithm Hypothesis

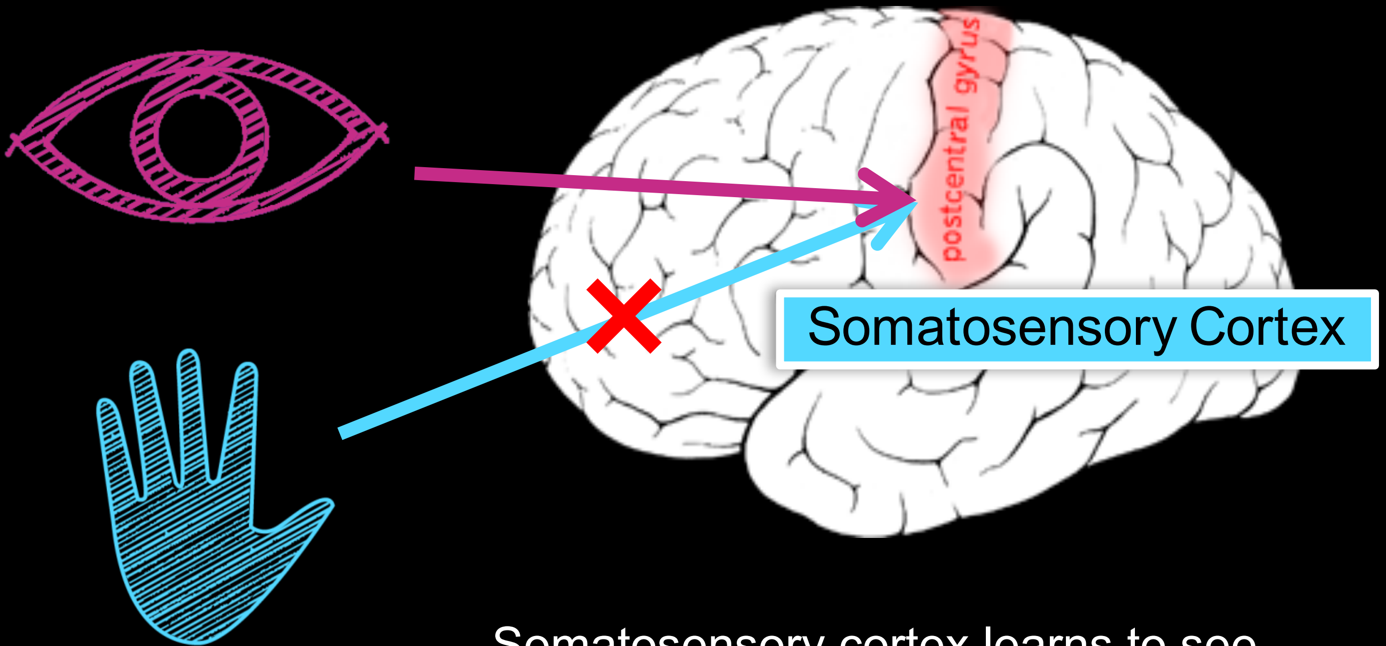


Auditory cortex learns to see

[Roe et al., 1992]

[Andrew Ng]

One Algorithm Hypothesis



Somatosensory cortex learns to see

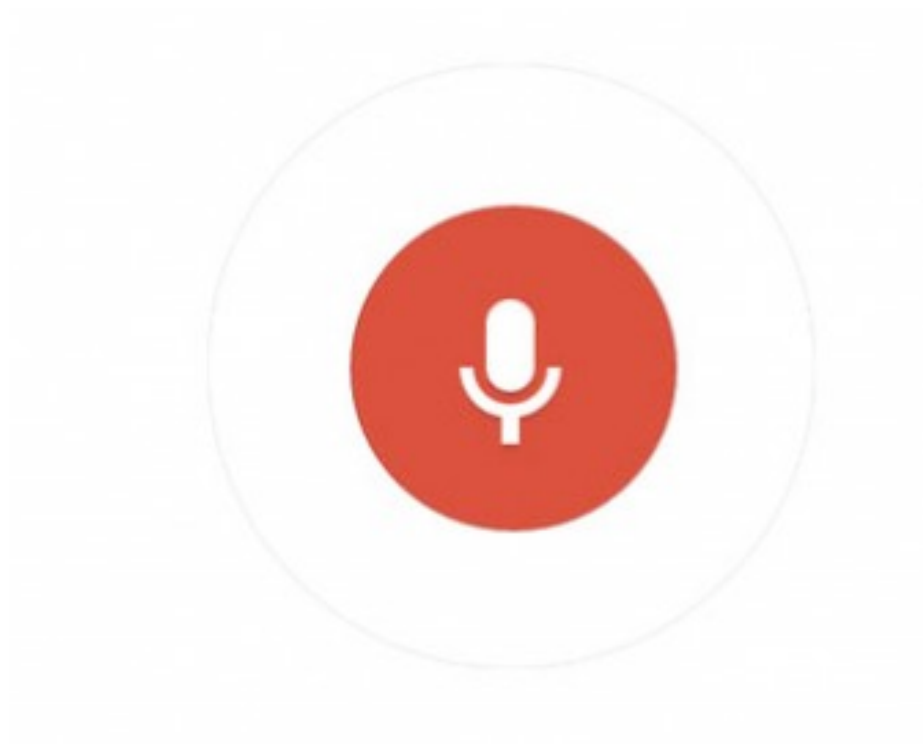
[Metin & Frost, 1989]

[Andrew Ng]

Tl;dr our brain is constantly decomposing

Told Vision Was 30 Years Out

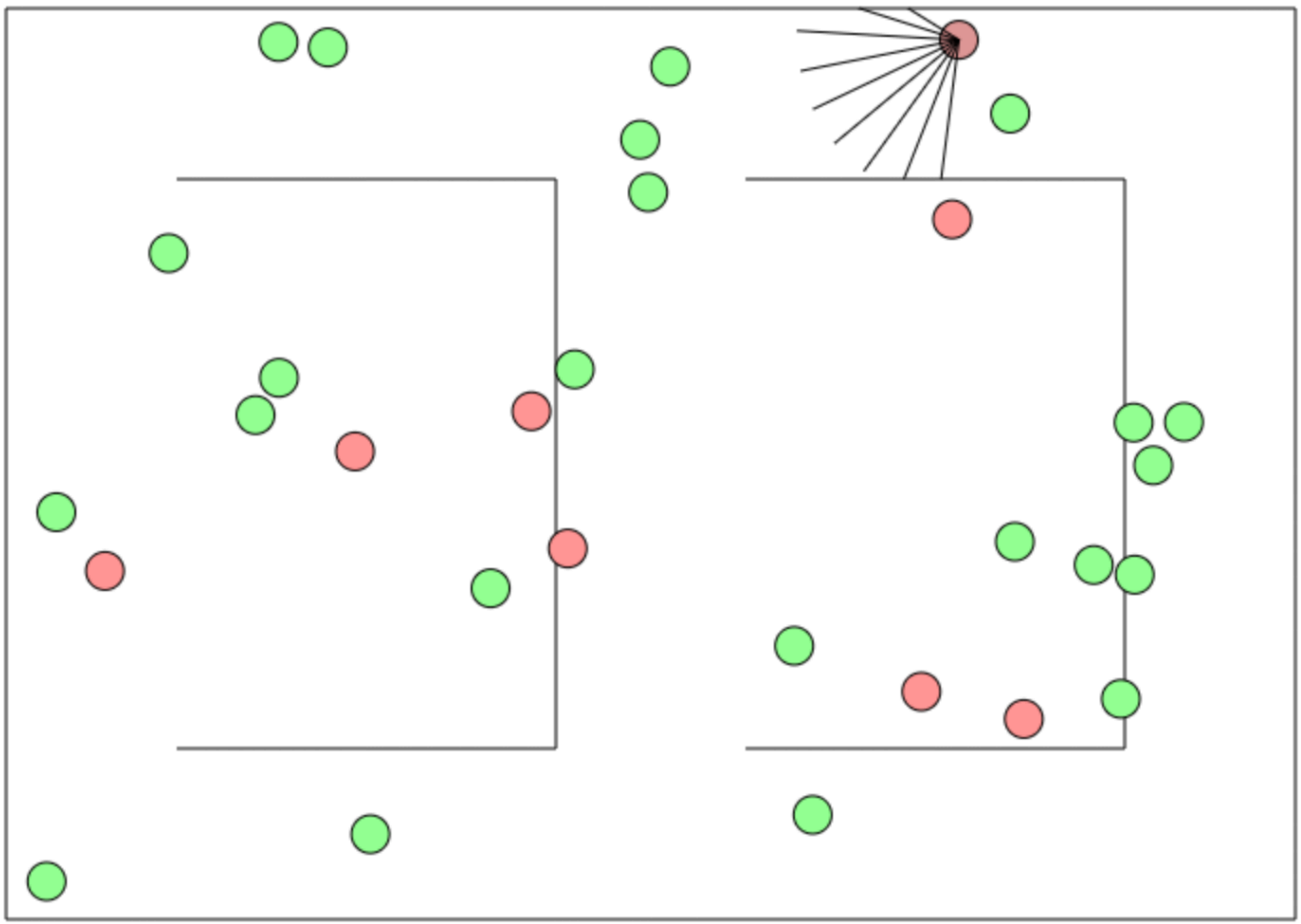
Told Speech Was 30 Years Out



Almost perfect…

Huge Progress

Deep Reinforcement Learning



http://cs.stanford.edu/people/karpathy/convnetjs/demo/rldemo.html

The end