



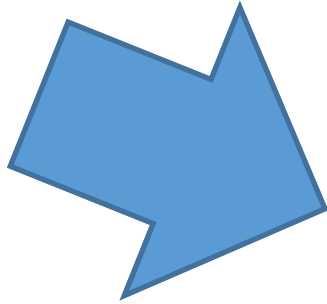
Combinatorics

Chris Piech

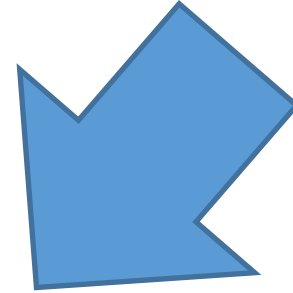
CS109, Stanford University

Review

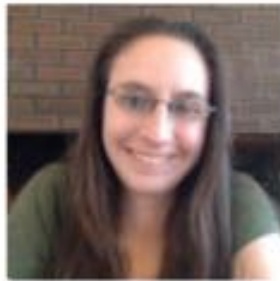
Essential Information



cs109.stanford.edu



CS109 Community



**Dedicated,
intelligent,
hardworking
teaching
assistants**

Counting

We are counting:

of **events**,

of **outcomes**,

of **objects**

Two Key Rules

Counting outcomes with **or**:

Inclusion Exclusion:

If outcomes can come from set A **or** set B, then the total number of outcomes is $|A| + |B| - |A \cap B|$.

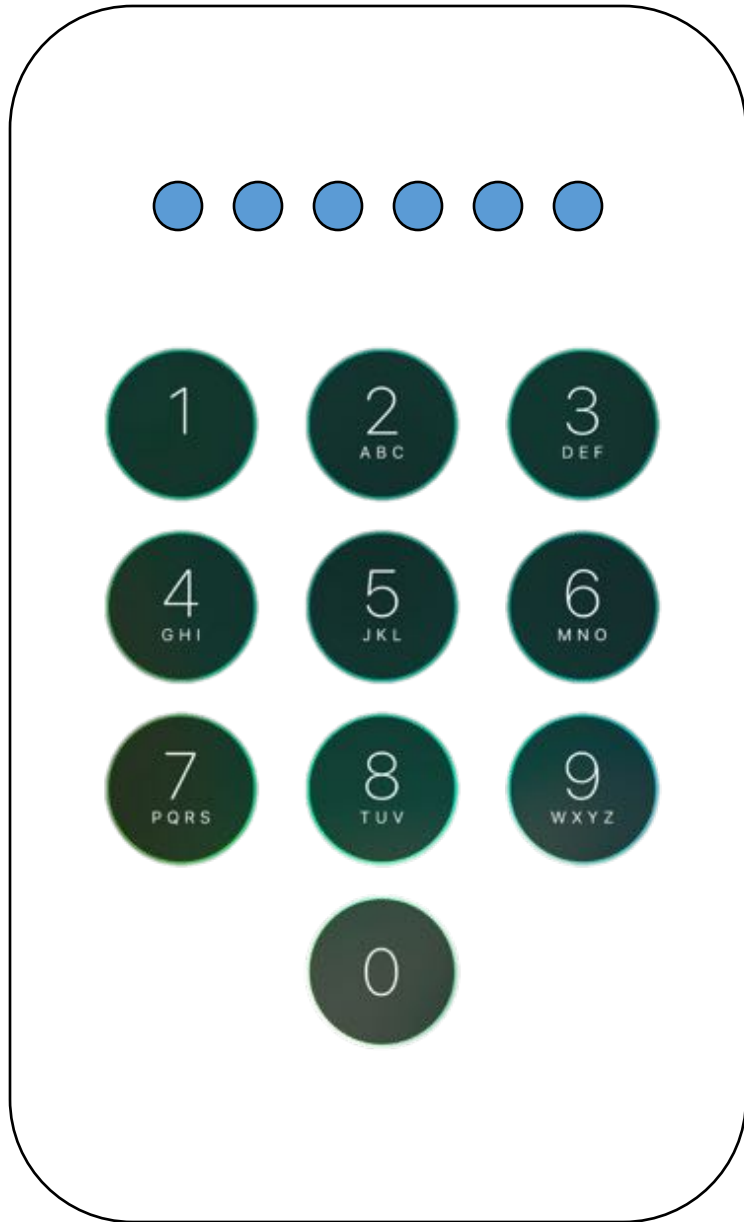
Counting outcomes with **steps**:

Product Rule of Counting:

If outcomes are generated via a process with r **steps**, where step i has n_i outcomes, then the total number of outcomes is:

$$n_1 \times n_2 \times \cdots \times n_r = \prod_{i=1}^r n_i$$

How Many Unique 6 digit passcodes?



Approach: count by steps

Step 1: first digit in passcode
(10 outcomes)

Step 2: second digit in passcode
(10 outcomes)

...

Step 6: sixth digit in passcode
(10 outcomes)

$$\begin{aligned}\text{total} &= n_1 \times n_2 \times \cdots \times n_r \\ &= 10 \times 10 \times 10 \times 10 \times 10 \times 10\end{aligned}$$

End Review

Combinatorics

Counting tasks on n objects

Sort objects
(permutations)

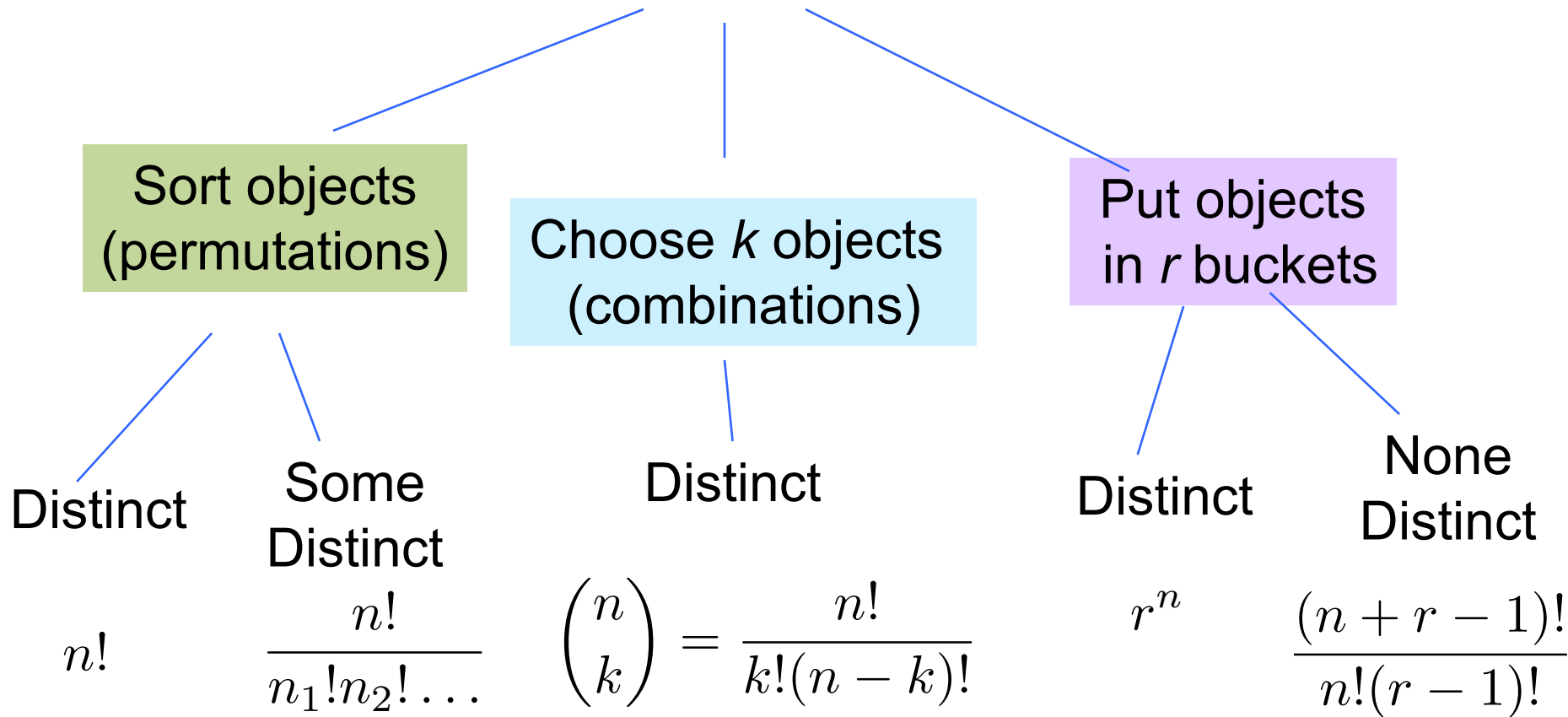
Choose k objects
(combinations)

Put objects
in r buckets



Combinatorics

Counting tasks on n objects



Combinatorics

Counting tasks on n objects

Sort objects
(permutations)

Choose k objects
(combinations)

Put objects
in r buckets

Distinct



Sort n Distinct Objects



Ayesha



Tim



Irina



Joey



Waddie

Sort n Distinct Objects

Sort 5 distinct cans:

Step 1: Chose first can (5 options)



Irina



Sort n Distinct Objects

Sort 5 distinct cans:

Step 1: Chose first can (5 options)

Step 2: Chose second can (4 options)



Irina



Waddie

$$5 \times 4 \times 3 \times 2 \times 1 =$$

... **120 unique sorts**



Sort n Distinct Objects

Def Permutations:

A permutation is an ordered arrangement of distinct object.

n objects can be permuted in:

$$n \times (n - 1) \times (n - 2) \times \cdots \times 2 \times 1 = n!$$

(Select 1st object out of n , then 2nd object out of $n - 1$, etc.)



Sort Distinct Objects



Ayesha



Tim



Irina



Joey



Waddie

= 120



Sort Semi-Distinct Objects



Coke



Tim



Coke



Joey



Waddie

$$= 120/2$$



Sort Semi-Distinct Objects

Making perms of distinct objects is a two step process

Step 1

Step 2

perms of distinct objects

=

perms considering some objects are indistinct

×

perms of just the indistinct objects



Sort Semi-Distinct Objects

perms of
distinct objects

=

perms
considering
some objects
are indistinct

×

perms of just
the indistinct
objects



Sort Semi-Distinct Objects

perms of
distinct objects

perms of just
the indistinct
objects

=

perms
considering
some objects
are indistinct



General Way to Count Permutations

Def: General Permutations:

When there are n objects
 n_1 are the same (indistinguishable) and
 n_2 are the same and

...

n_r are the same,

There are:
$$\frac{n!}{n_1!n_2!\dots n_r!}$$

Unique orderings (“permutations”)



How many orderings?



Coke



Coke0



Coke



Coke0



Coke0

$$= 120 / (3! \times 2!) = 10$$



How many orderings of letters?

$$M00 = \frac{3!}{2!} = 3$$

M I S S I S S I P P I

$$= \frac{11!}{1!4!4!2!} = 34,650$$



Combinatorics

Counting tasks on n objects

Sort objects
(permutations)

Choose k objects
(combinations)

Put objects
in r buckets

Distinct

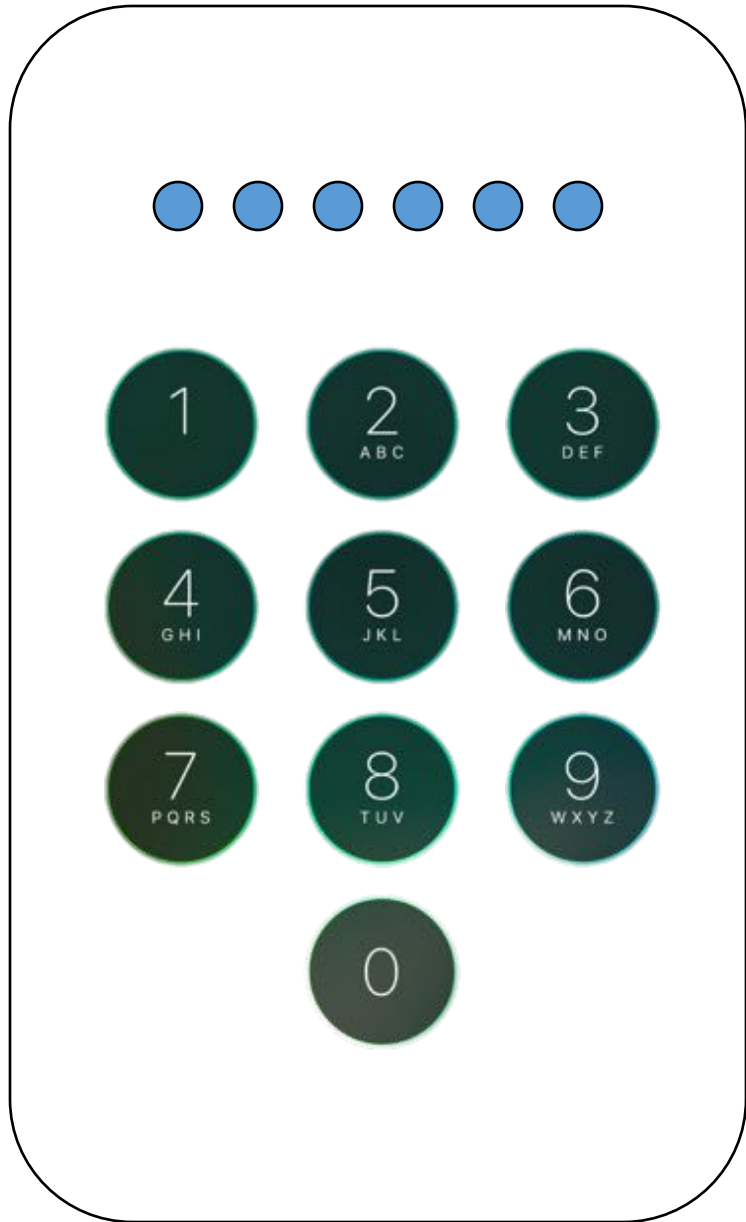
$$n!$$

Some
Distinct

$$\frac{n!}{n_1!n_2!\dots}$$



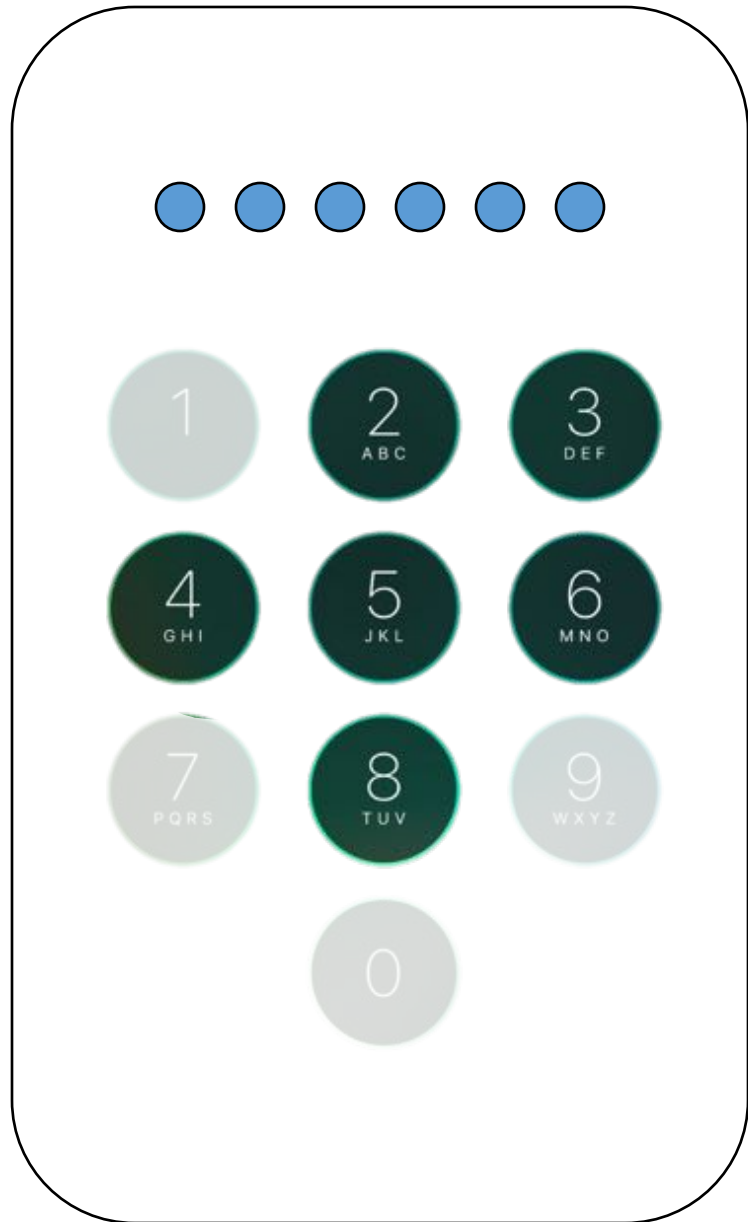
How Many Unique 6 digit passcodes?



How many unique 6 digit passcodes are there?

$$10^6 = 1,000,000$$

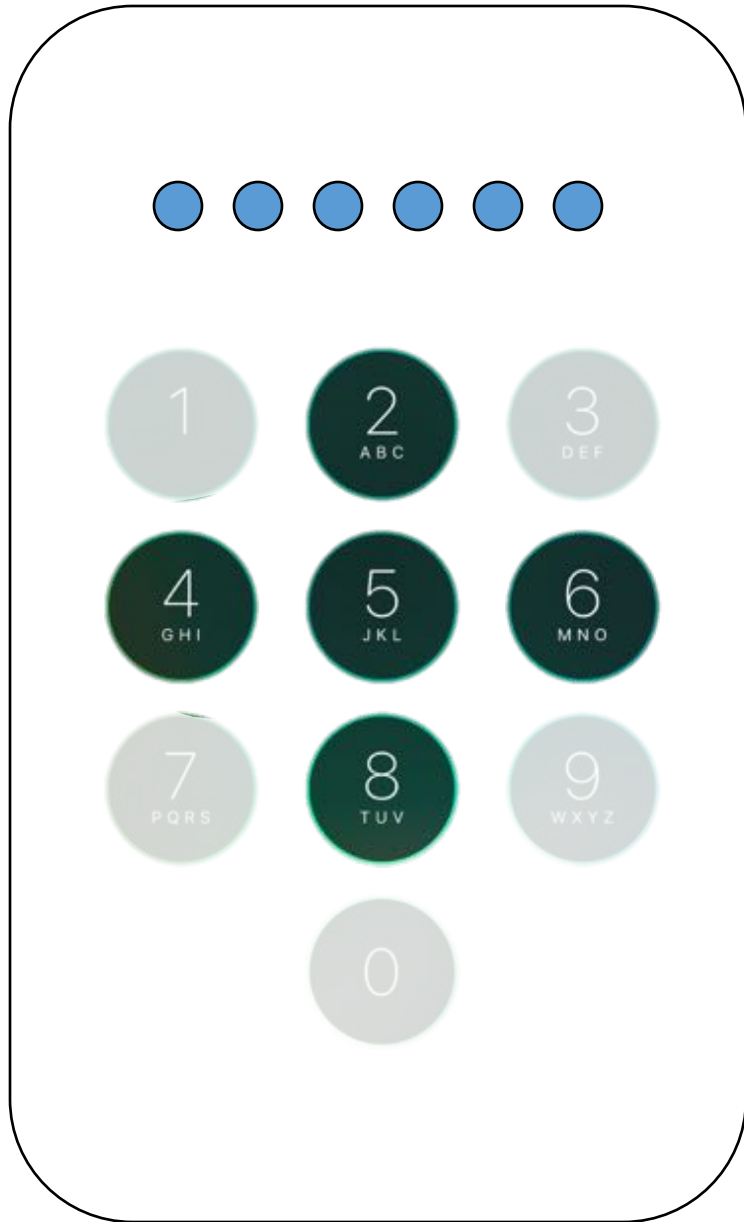
How many possible codes 6 smudges?



If a phone password uses each of **six** distinct numbers, how many unique six digit passcodes are there?

$$6! = 720$$

How many possible codes 5 smudges?



If a phone password uses each of **five** distinct numbers, how many unique six digit passcodes are there?

Five mutually exclusive cases:

2 was repeated

4 was repeated

5 was repeated

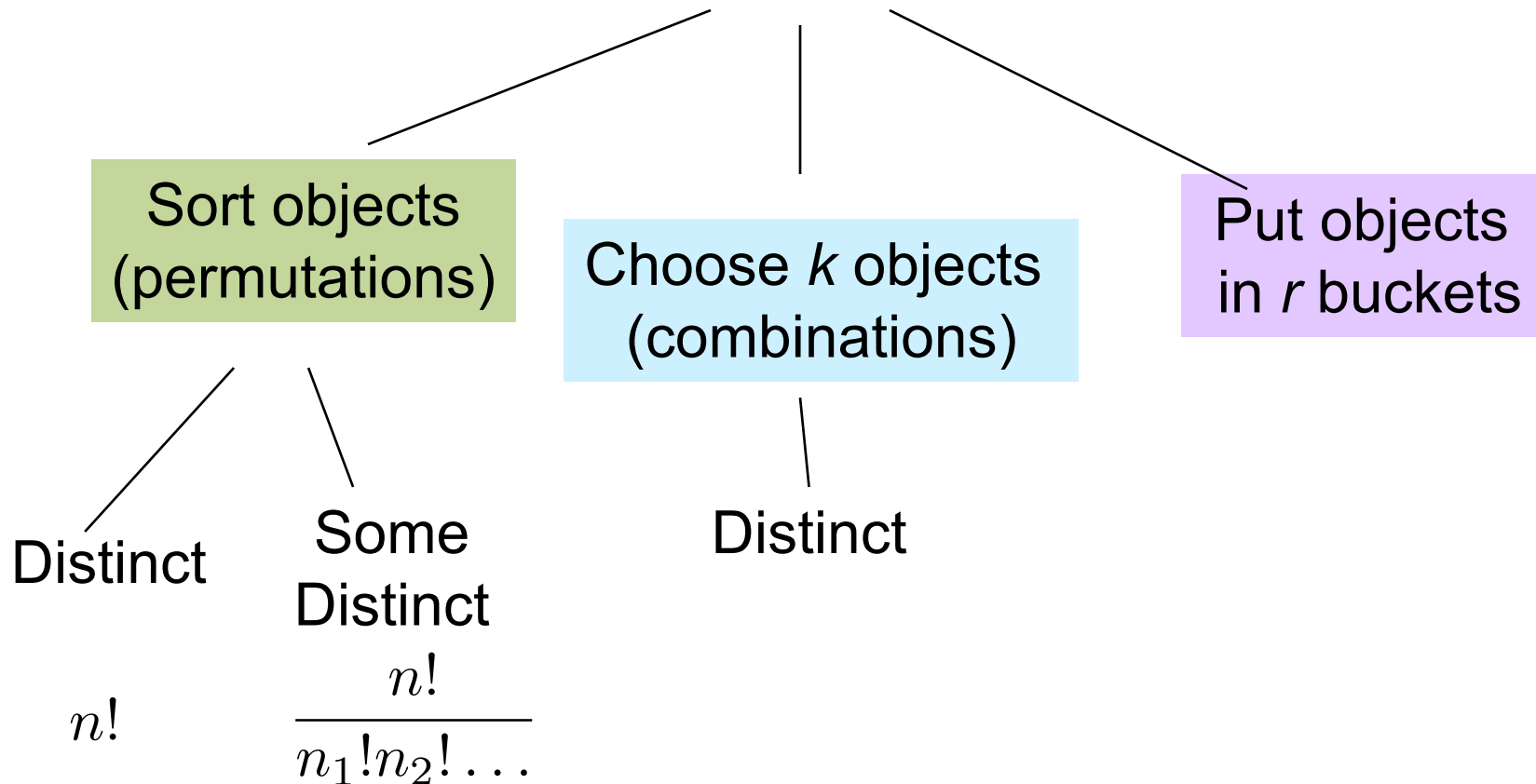
6 was repeated

8 was repeated

$$= 5 \times \frac{6!}{2!} = 1,800$$

Combinatorics

Counting tasks on n objects





Combinatorics

There are $n = 20$ people

How many ways can we **chose** $k = 5$ people to get cake?



Consider this
generative process

Step 1: Randomly order people

There are $n = 20$ people

How many ways can we **chose** $k = 5$ people to get cake?



step 1 ways = $n!$

Step 2: Draw a line at pos k

There are $n = 20$ people

How many ways can we **chose** $k = 5$ people to get cake?



1



2



3



4



5



6



7



8



9



10



11



12



13



14



15



16



17



18



19



20

step 2 ways = 1

Step 3: Allow Cake Group to Mingle

There are $n = 20$ people

How many ways can we **chose** $k = 5$ people to get cake?



6

7

8



9



10



11



12



13



14



15



16



17



18



19



20

$k!$ different permutations
lead to the same mingle

Step 4: Allow nonCake Group to Mingle

There are $n = 20$ people

How many ways can we **chose** $k = 5$ people to get cake?



$(n - k)!$ different permutations lead to the same mingle

Step 4: Allow nonCake Group to Mingle

There are $n = 20$ people

How many ways can we chose $k = 5$ people to get cake?

Randomly
order n
objects

Designate the
first k as
chosen

$$\text{num ways} = n! \times 1 \times \frac{1}{k!(n-k)!}$$

Any ordering of
chosen group is
the same choice

Any ordering of
non-chosen
group is the
same choice

Step 4: Allow nonCake Group to Mingle

There are $n = 20$ people

How many ways can we **chose** $k = 5$ people to get cake?

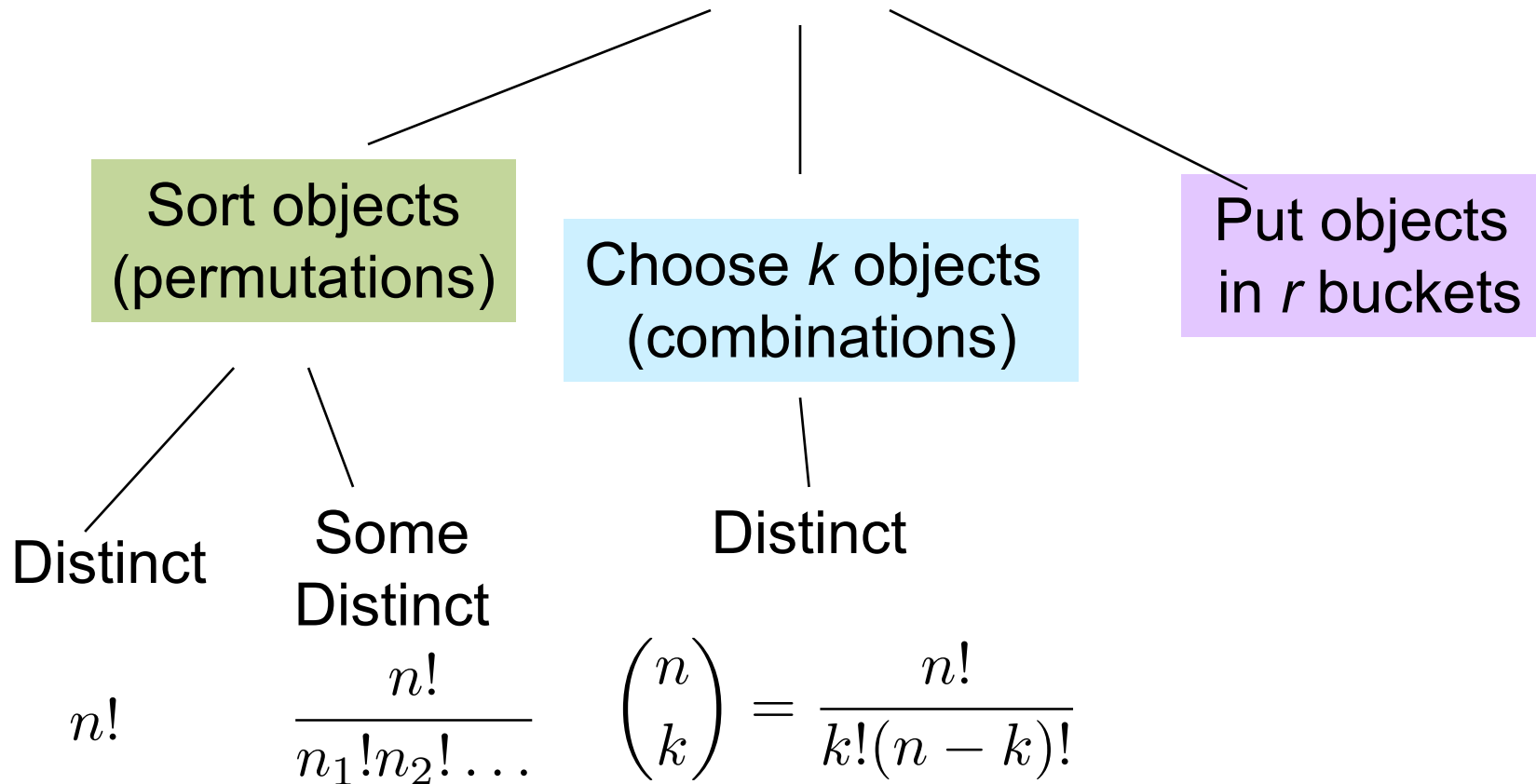
$$\text{num ways} = \binom{n}{k}$$

* Also called binomial coefficients

$$(x + y)^n = \sum_{k=0}^n \binom{n}{k} x^k \cdot y^{n-k}$$

Combinatorics

Counting tasks on n objects



8,000 villagers.
How many distinct ways can you
chose 2 to play a game?

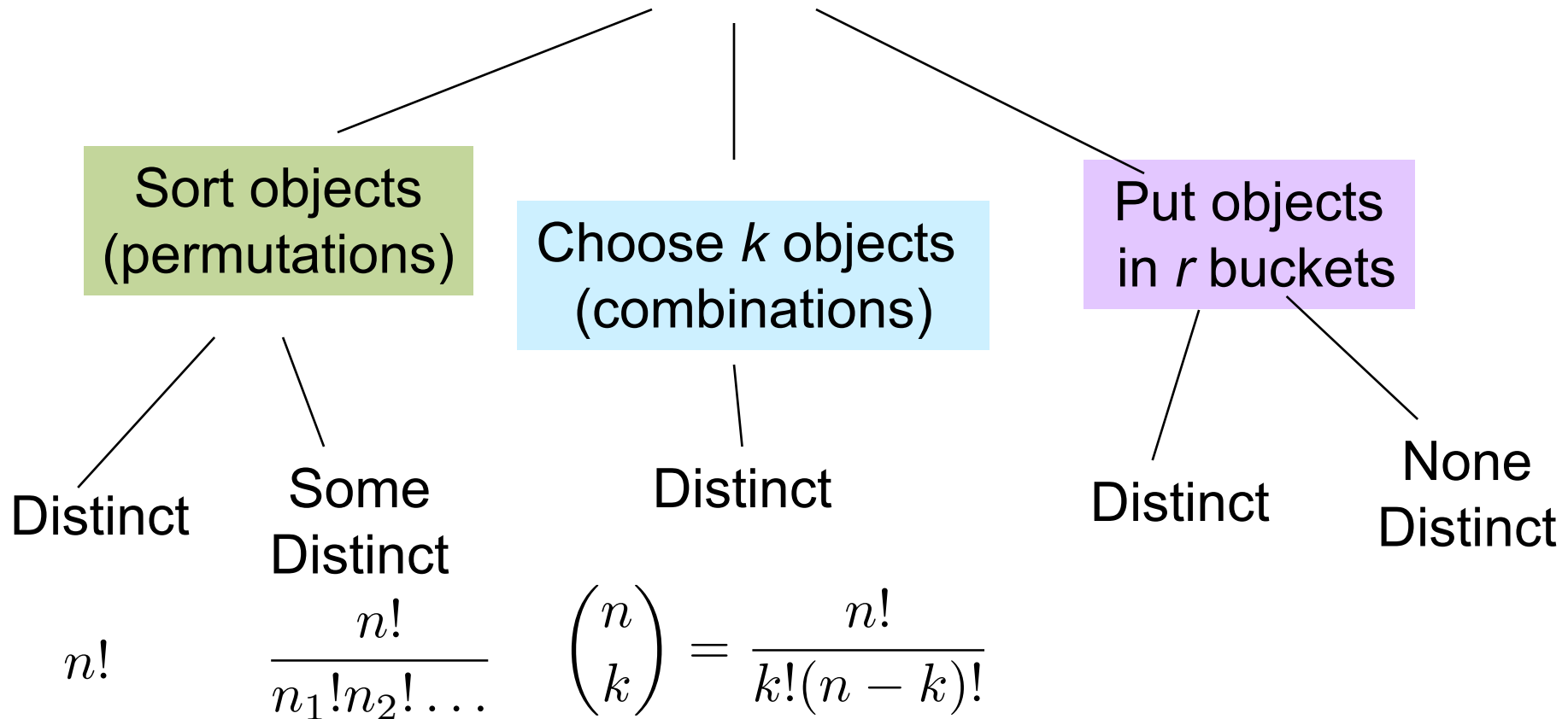
$$= \frac{8000!}{7998!2!} = 31,996,000$$





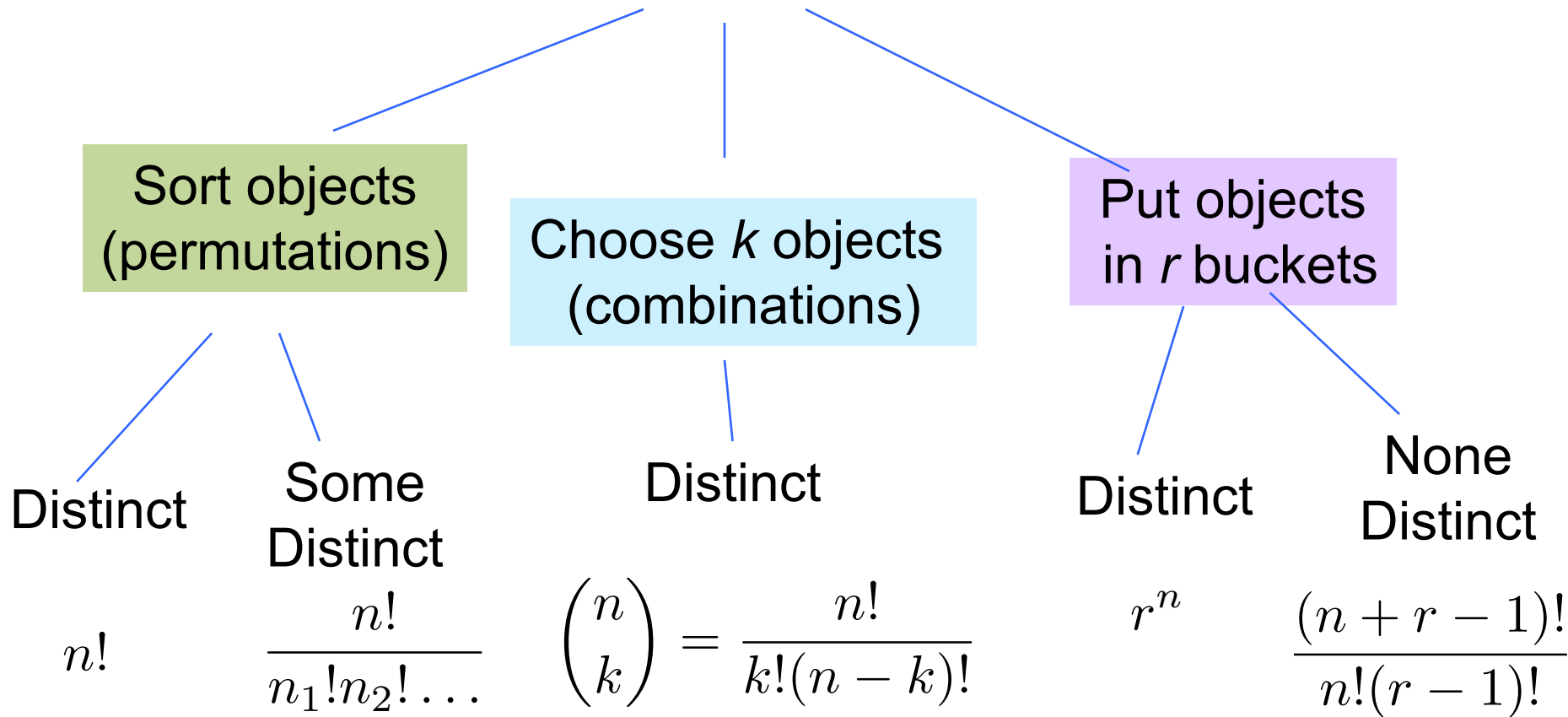
Combinatorics

Counting tasks on n objects



Combinatorics

Counting tasks on n objects



Something is going on in the world of AI

Modern AI
or, How we learned to combine
probability and programming

Brief History

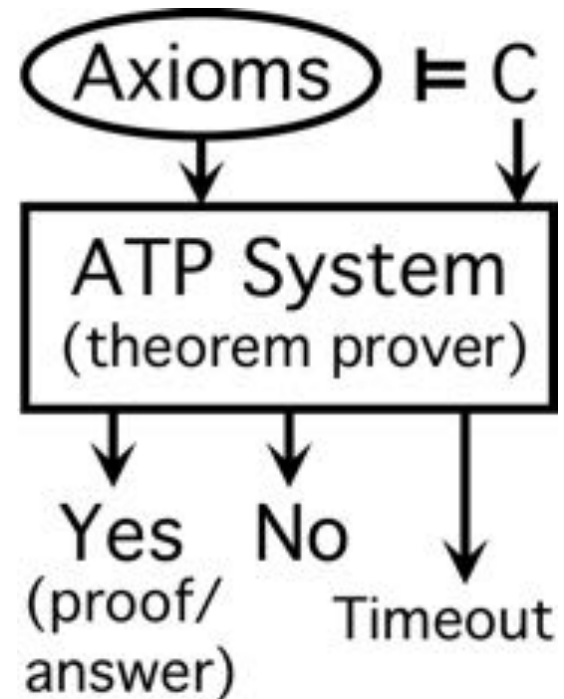


Early Optimism 1950

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

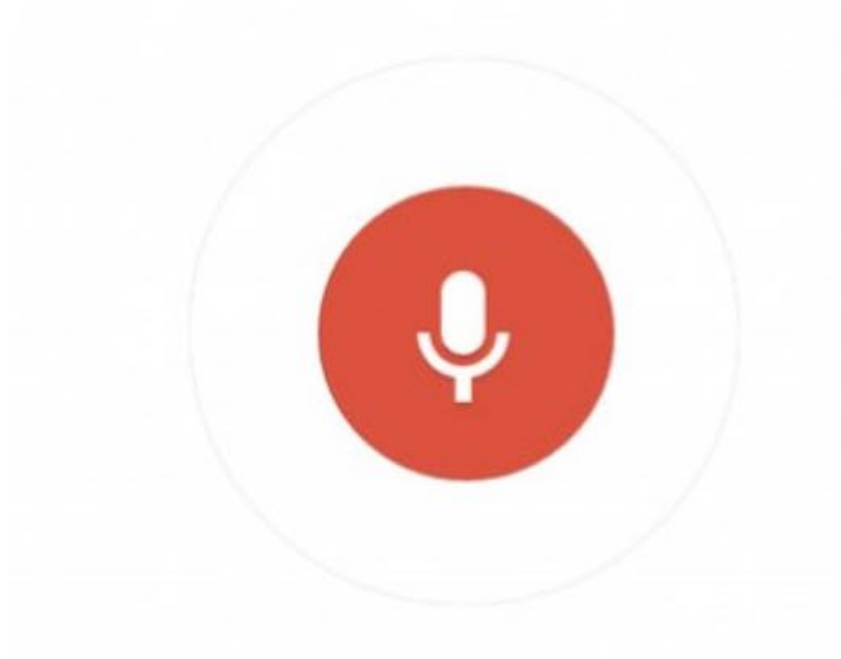


BRACE YOURSELVES



WINTER IS COMING

Told Speech Was 30 Years Out



Almost perfect...



What is going on?

[suspense]

Focus on one problem

Computer Vision



Piech, CS106A, Stanford University



Classification



That is a picture
of a **one**



Classification



That is a picture
of a **zero**



Classification



That is a picture
of an **zero**



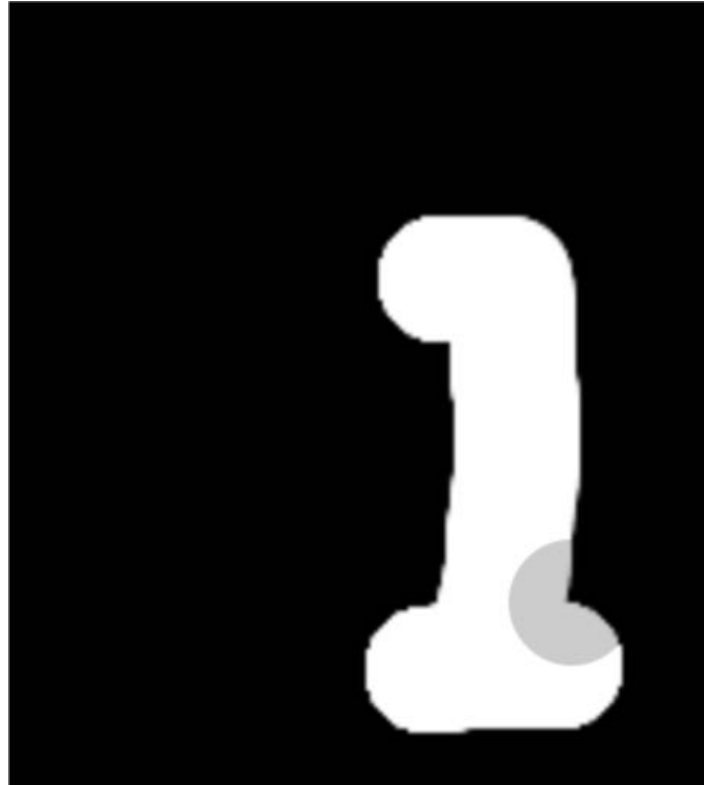
* It doesn't have to be
correct all of the time

Can you do it?

What number is this?



What number is this?



How about now?

What a computer sees

0	0	1	0	1	0	1	0	0	0	1	1	1	0	1
1	0	0	1	0	1	1	1	0	1	0	0	0	0	0
1	1	1	0	1	0	0	1	1	0	0	1	0	1	0
1	1	1	1	1	0	0	0	0	0	1	1	0	1	1
0	0	0	1	1	0	0	1	0	0					
1	0	0	1	1	0	0	0	1	0					
1	1	0	1	1	0	0	1	1	0					
1	0	1	0	0	1	0	0	1	0					
0	0	0	0	1	0	1	0	1	1					
0	1	1	0	0	0	0	0	1	1					
0	0	1	0	1	1	1	0	0	0					
0	1	1	1	0	1	0	0	1	0					
1	1	0	0	0	0	0	0	0	0					
0	0	0	0	0	0	0	0	1	1					
0	0	1	1	1	0	1	0	1	1					



What a human sees



Very hard to Program



```
public class HarryHat extends ConsoleProgram {  
  
    public void run() {  
        println("Todo: Write program");  
    }  
  
}
```

Two Great Ideas

1. Probability from Examples

2. Artificial Neurons

Two Great Ideas

1. Probability from Examples

2. Artificial Neurons

1. Probability From Examples



When Does the Magic Happen?

Lots of
Data + Sound
Probability



Machine Learning

Basically just a rebranding of statistics
and probability.



Vision is Hard

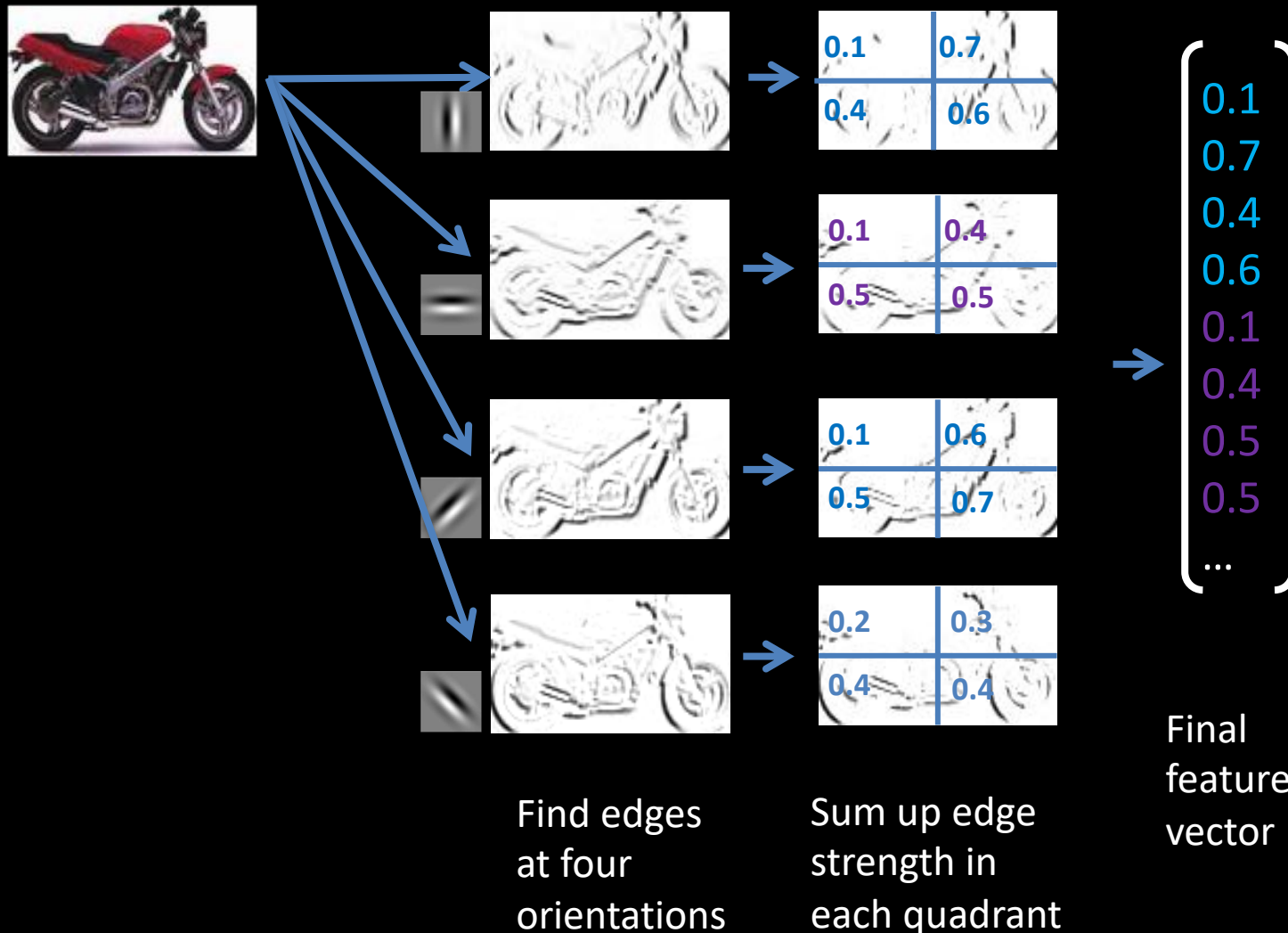
You see this:



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

Human Designed Features



Some Great Thinkers



Daphne Koller

Straight ML Not Perfect...



Two Great Ideas

1. Probability from Examples

2. Artificial Neurons

Two Great Ideas

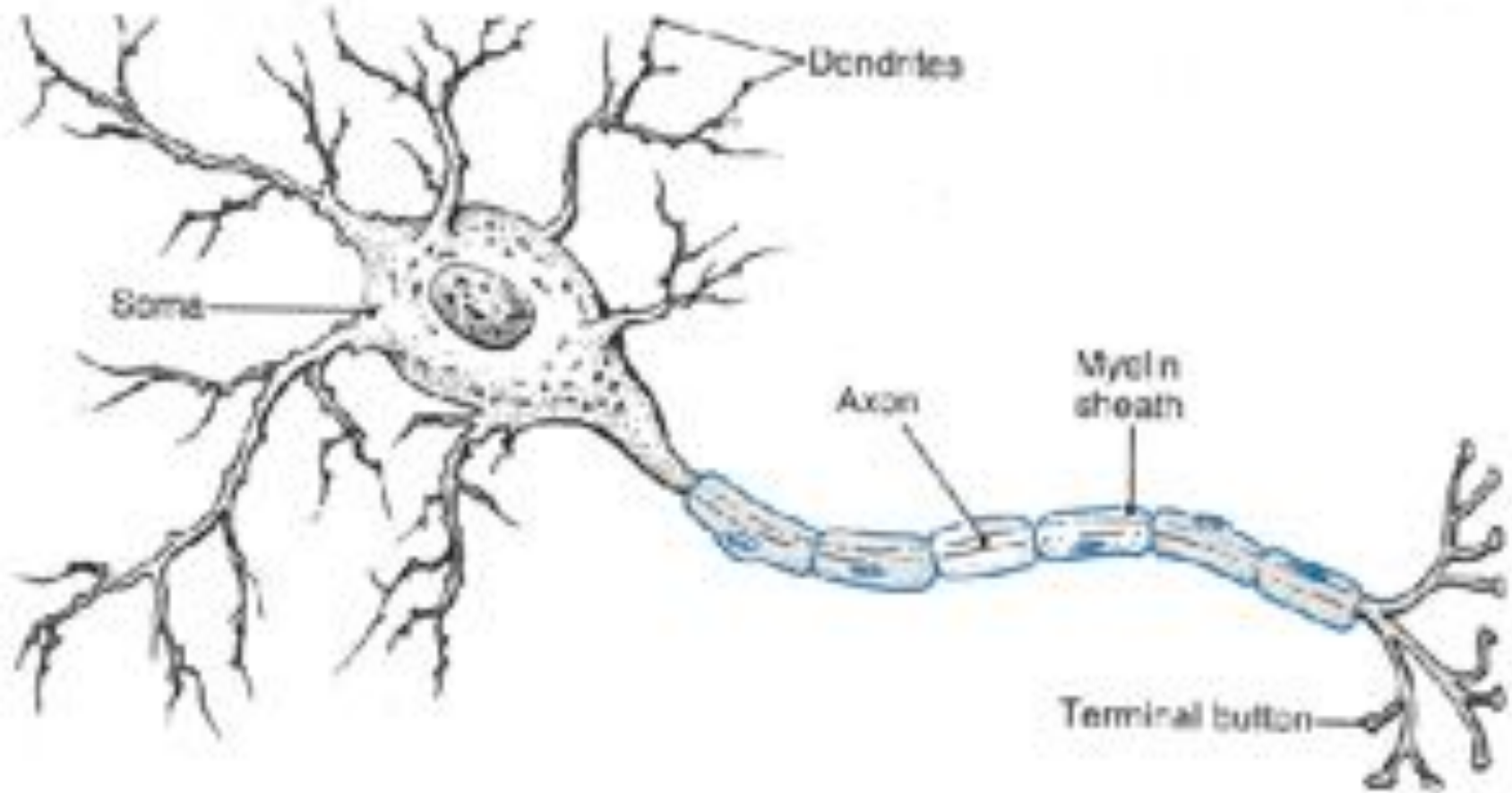
1. Probability from Examples

2. Artificial Neurons

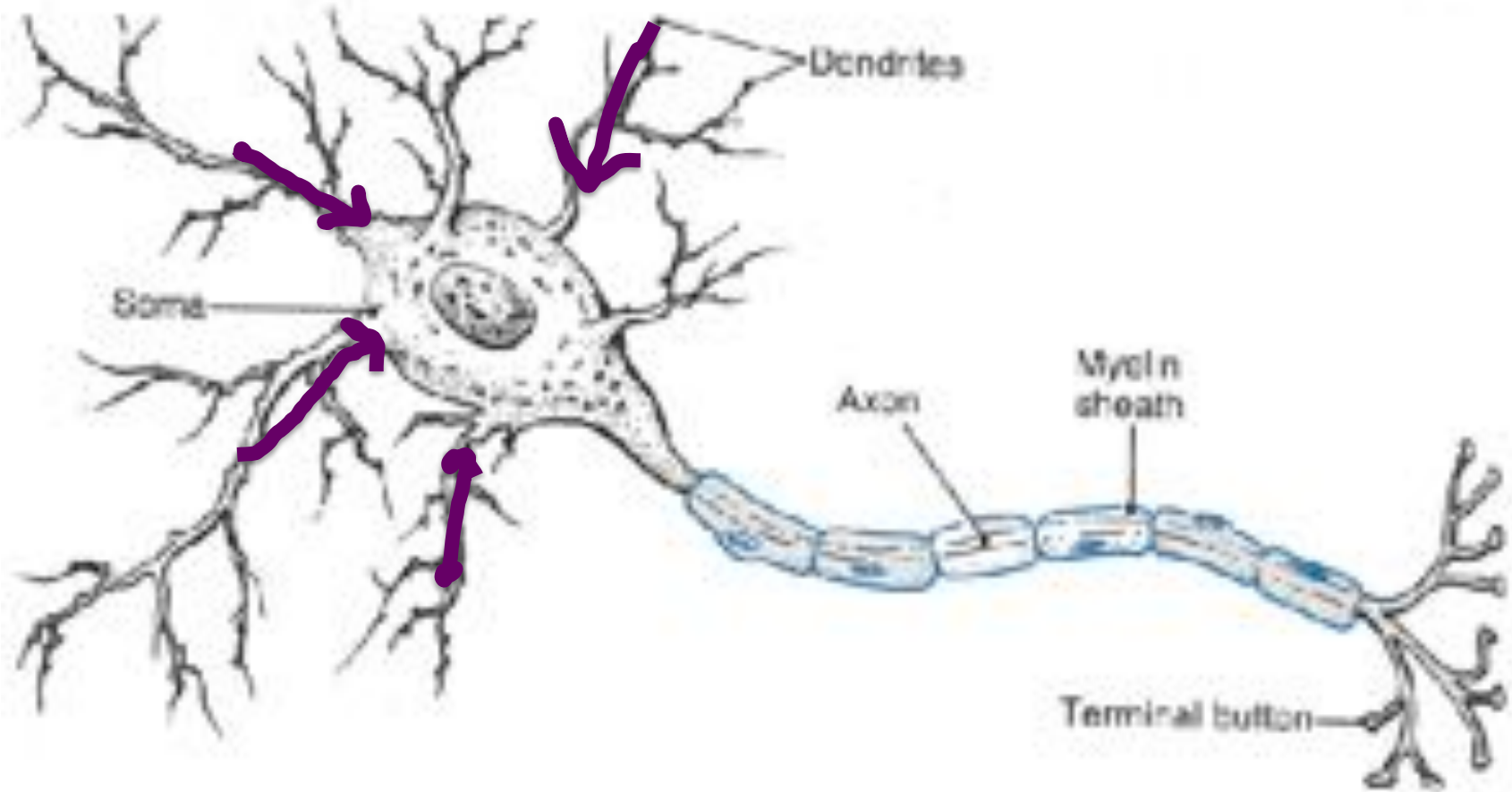
2. Artificial Neurons



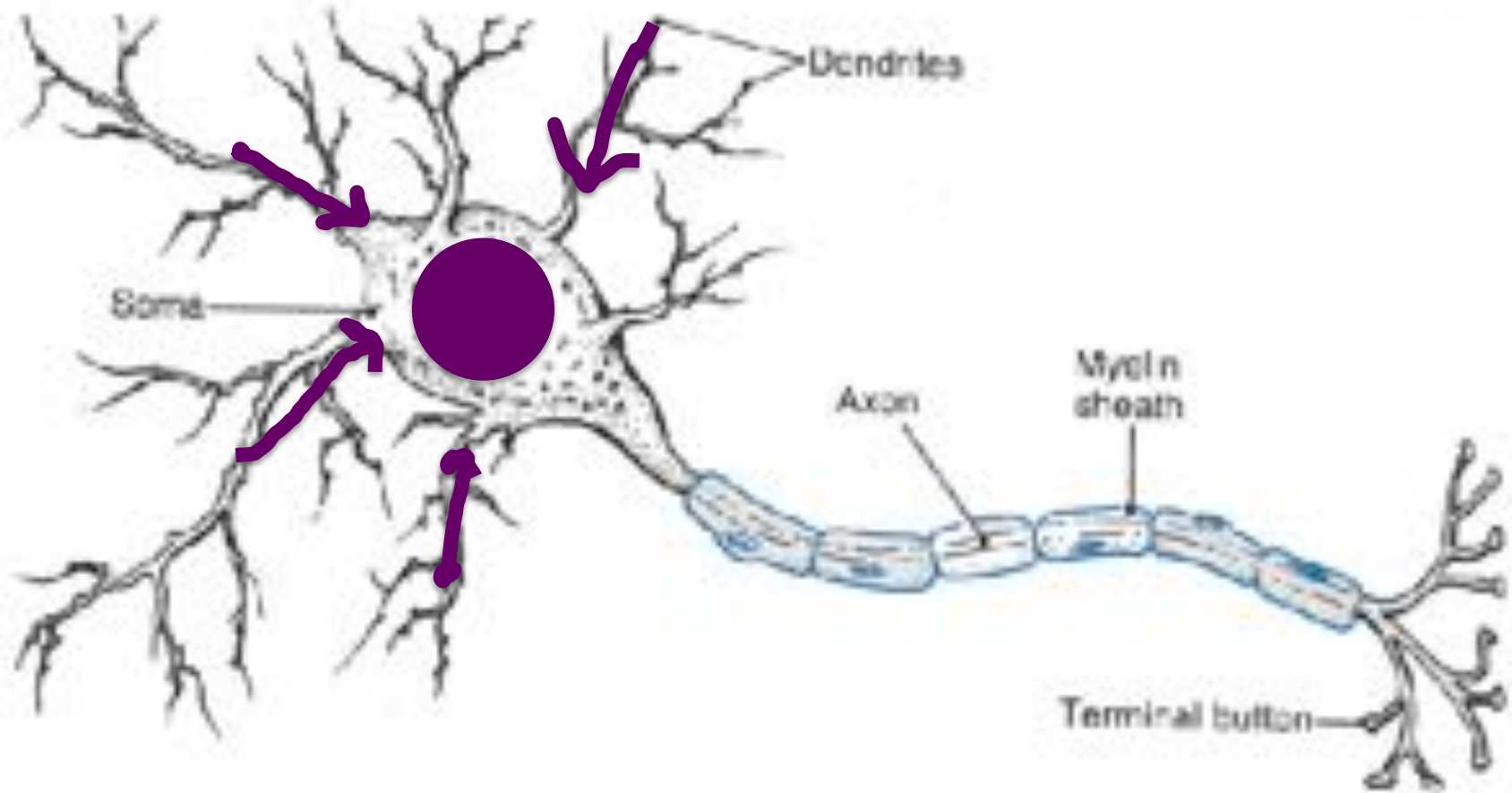
Neuron



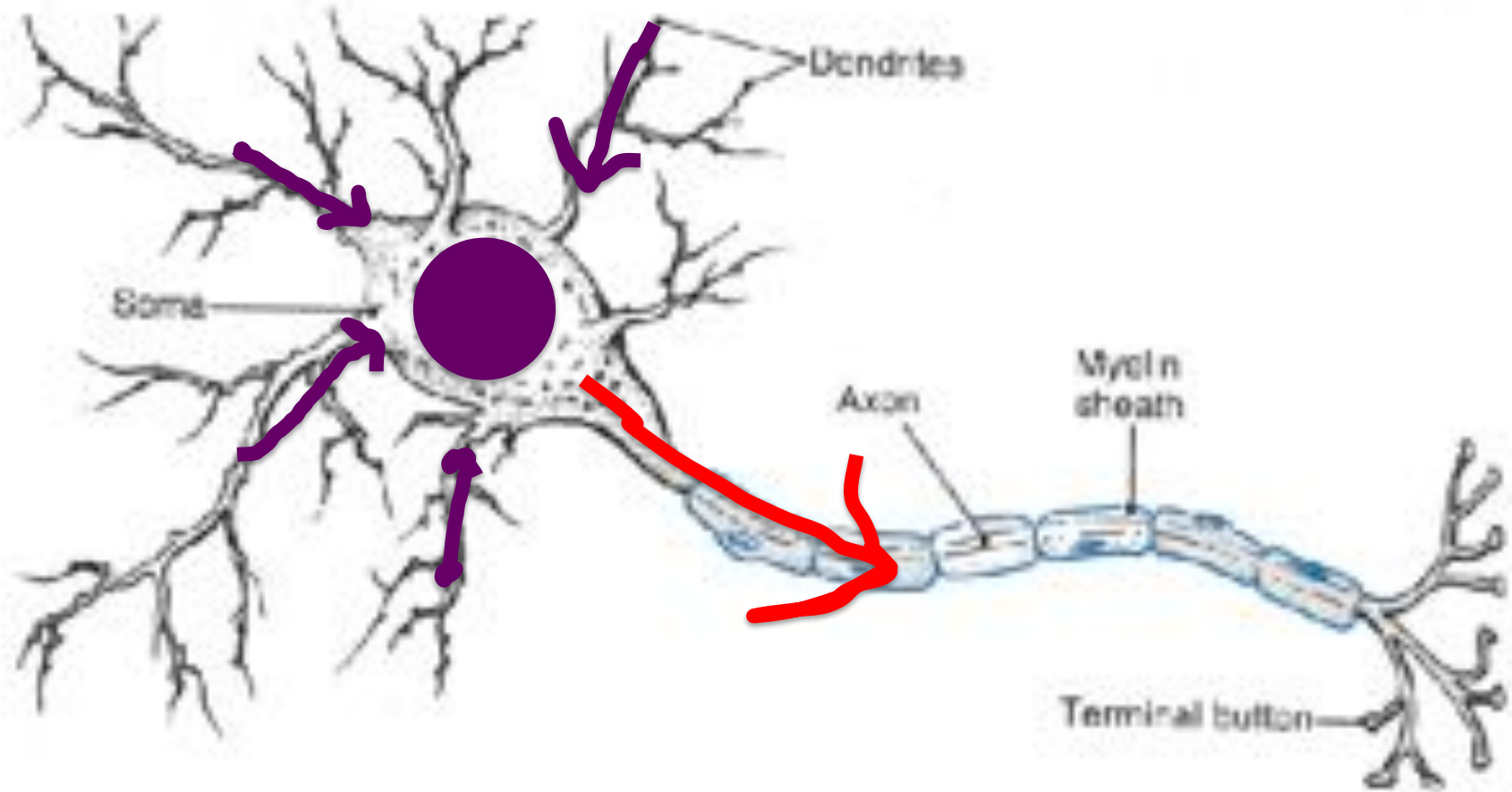
Neuron



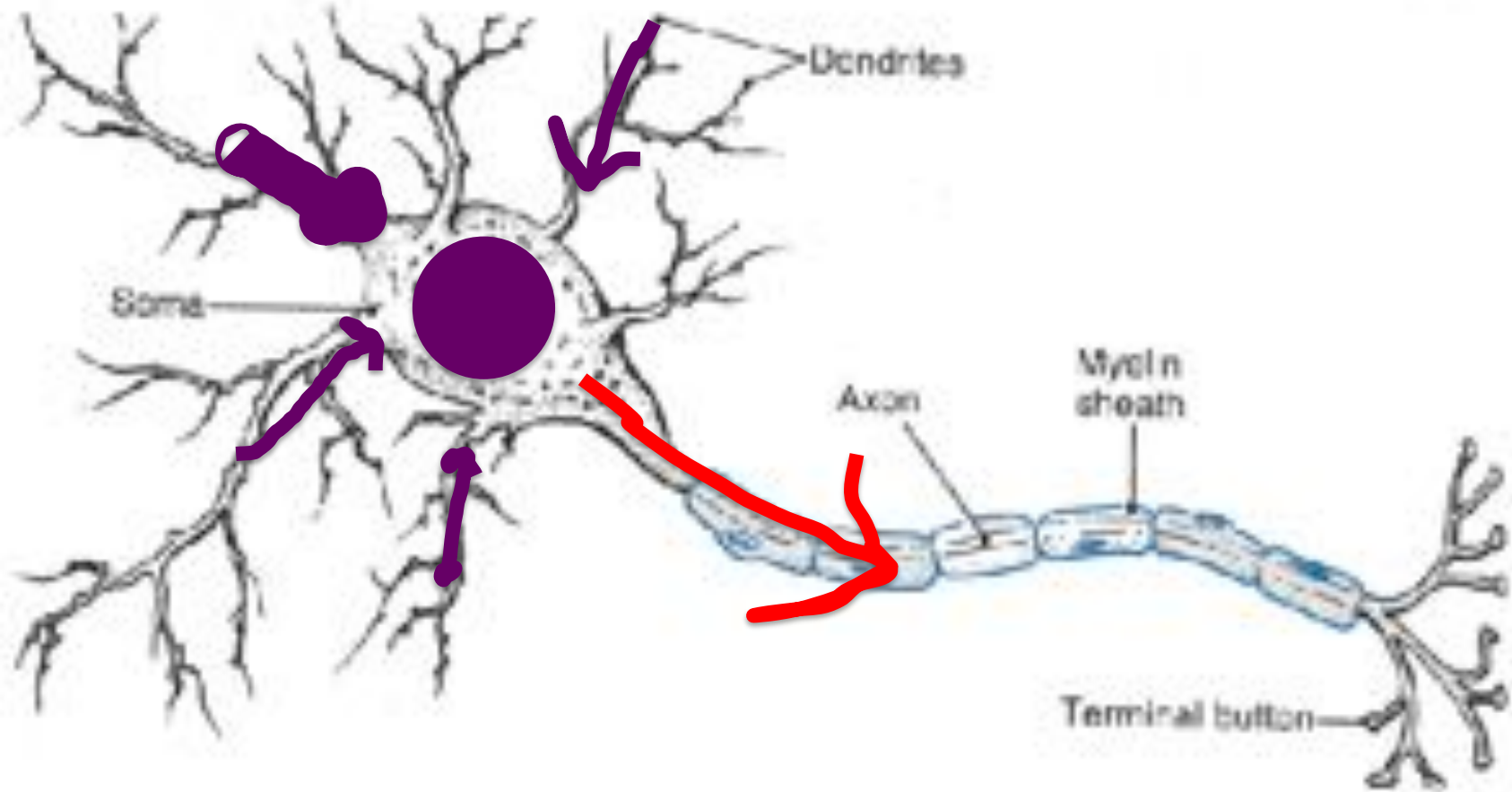
Neuron



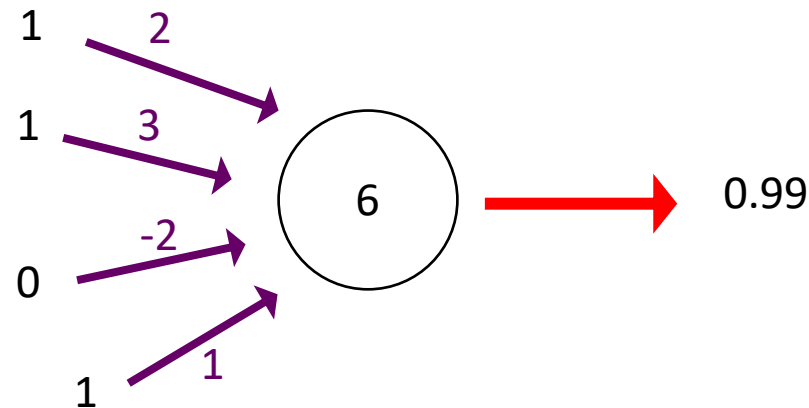
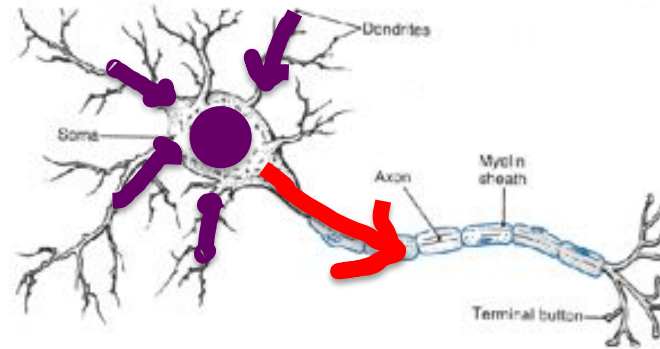
Neuron



Some Inputs are More Important



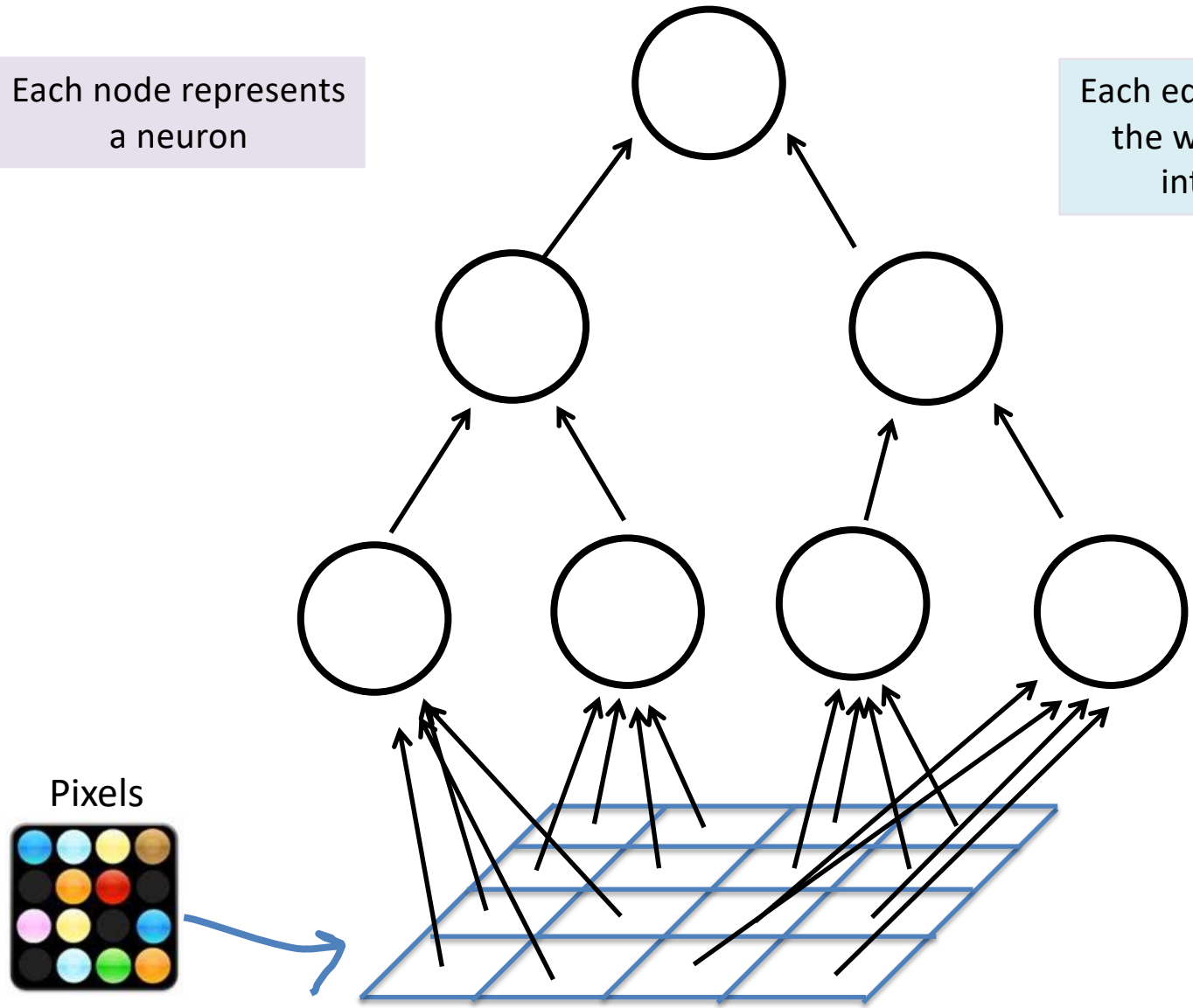
Artificial Neuron



Neural Network

Each node represents a neuron

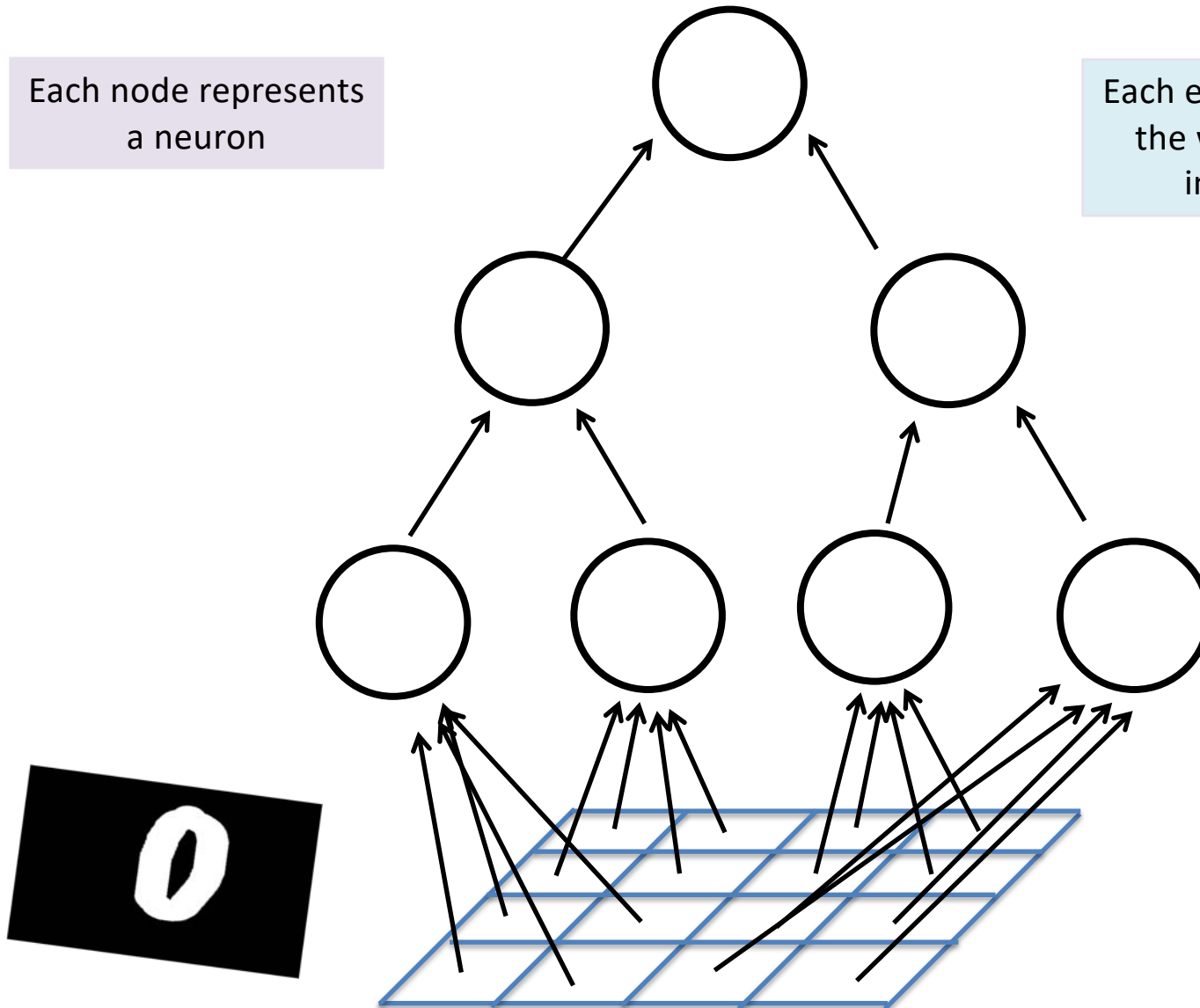
Each edge represents the weight of the interaction



Neural Network

Each node represents a neuron

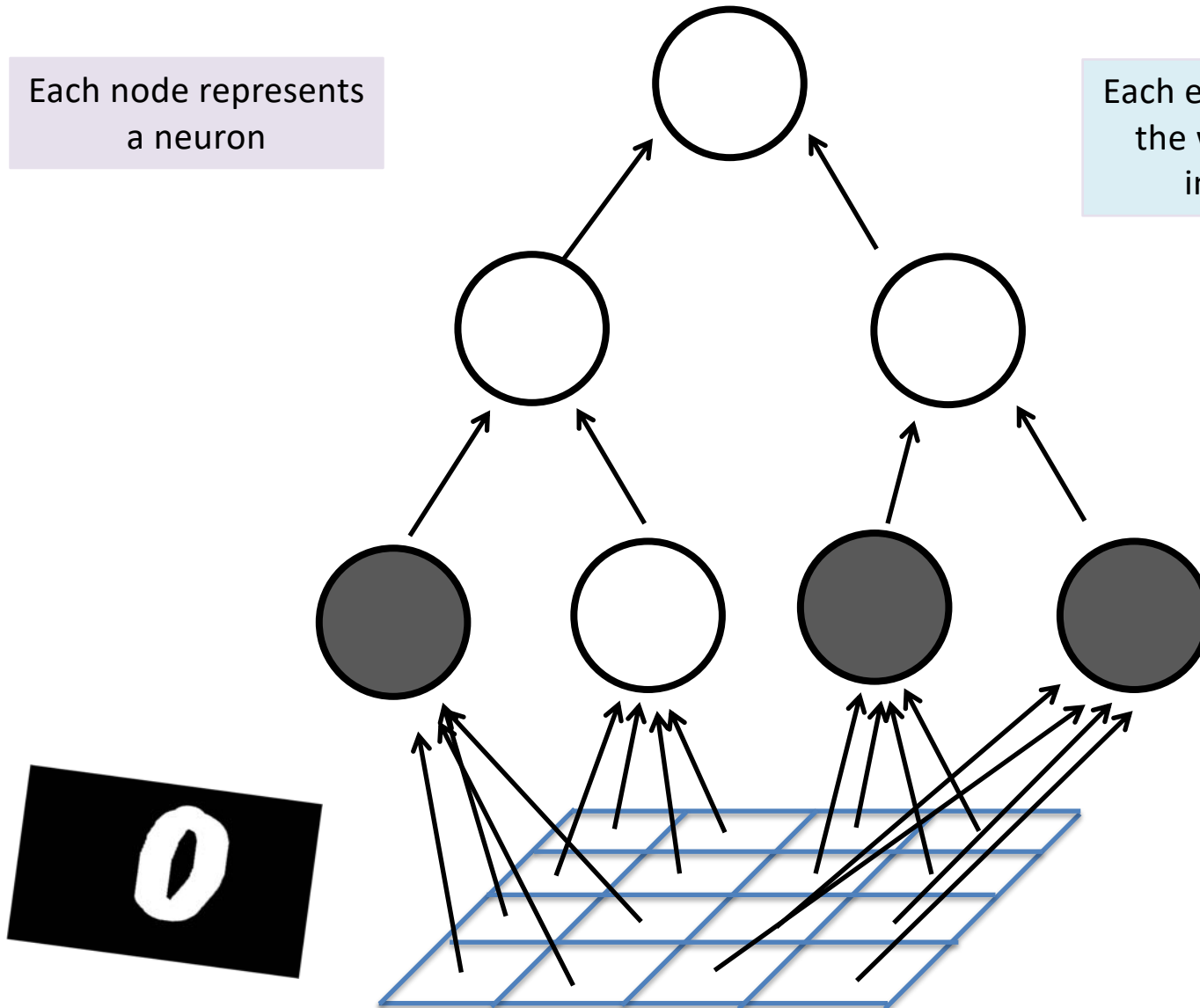
Each edge represents the weight of the interaction



Neural Network

Each node represents a neuron

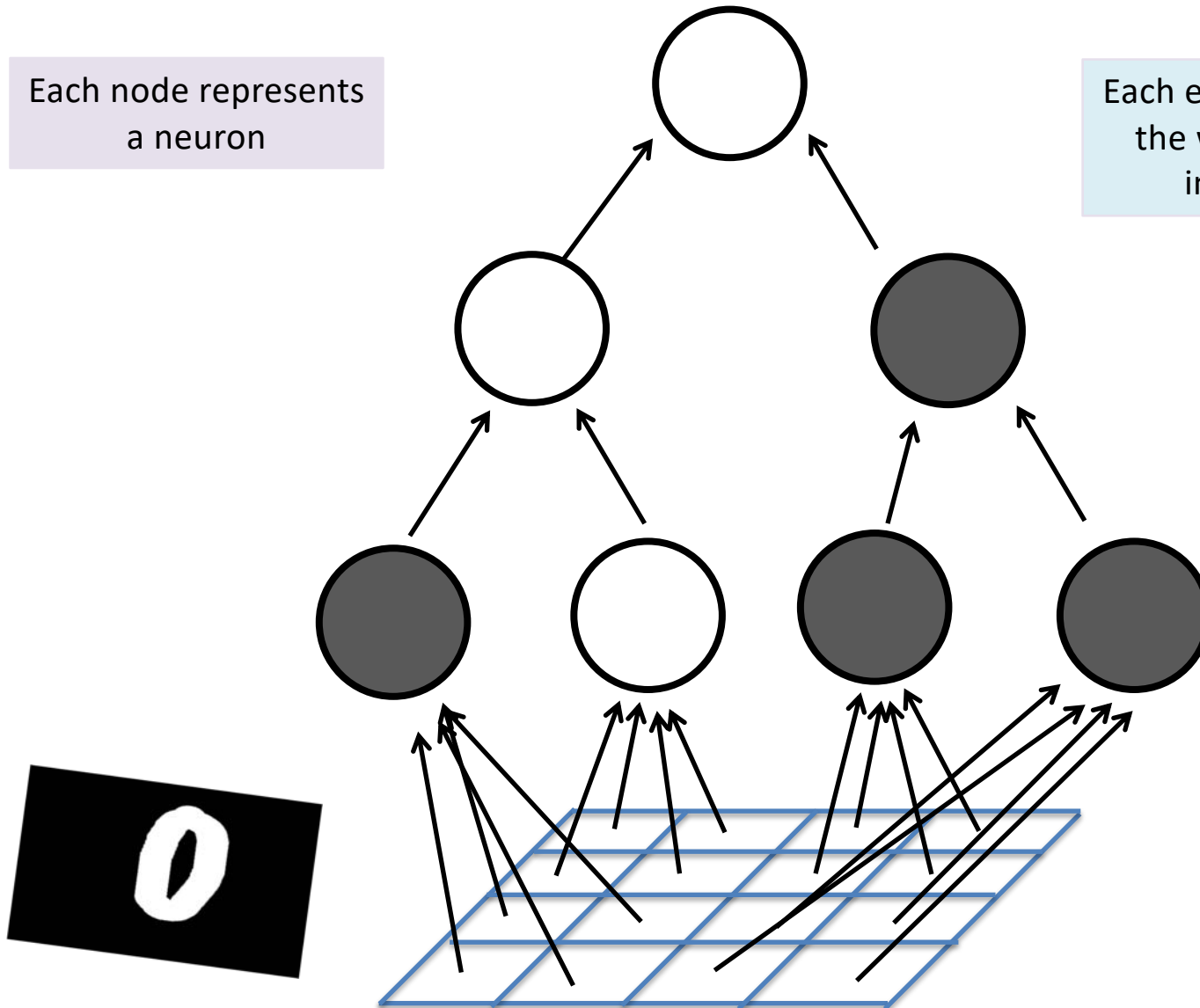
Each edge represents the weight of the interaction



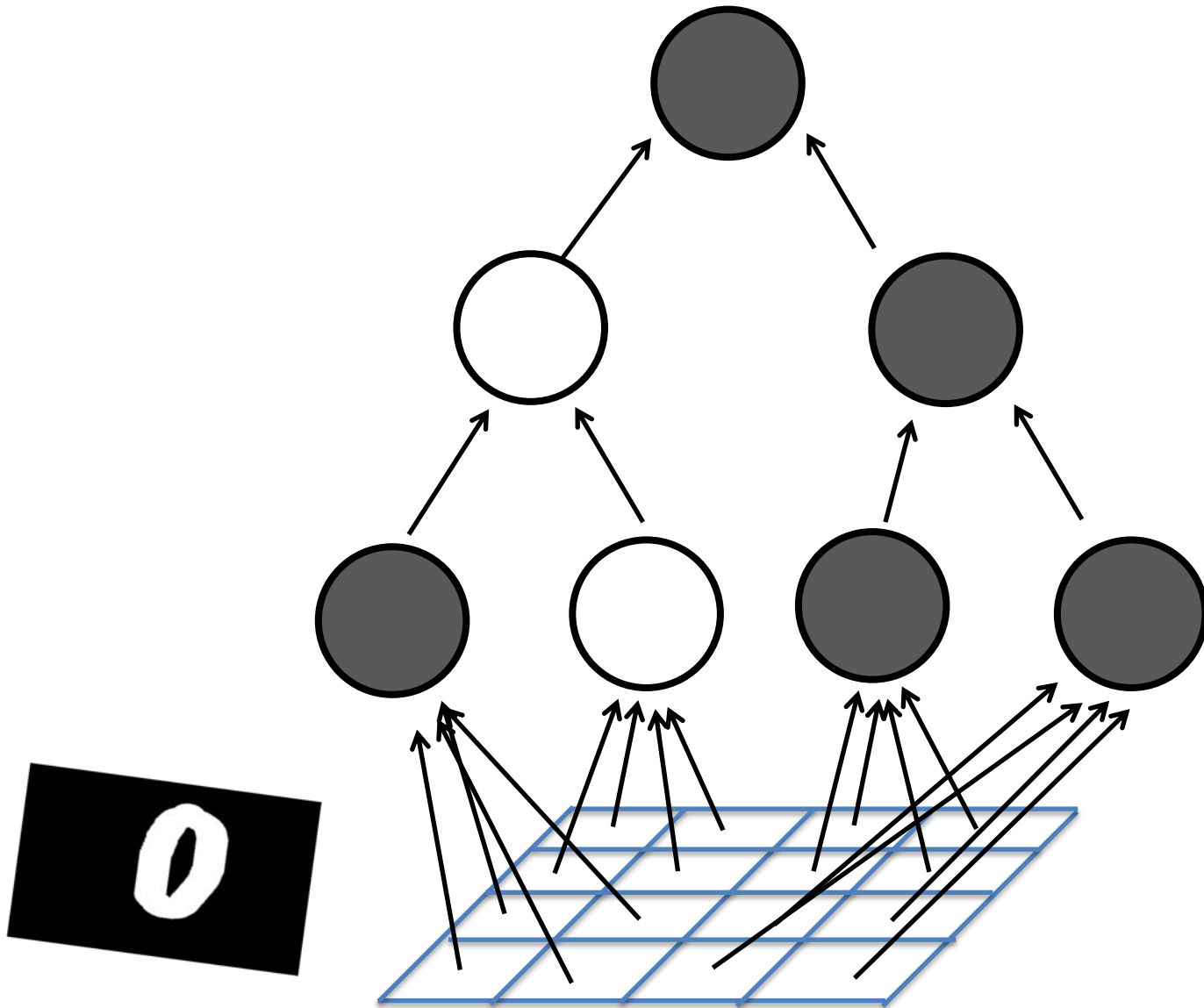
Neural Network

Each node represents
a neuron

Each edge represents
the weight of the
interaction

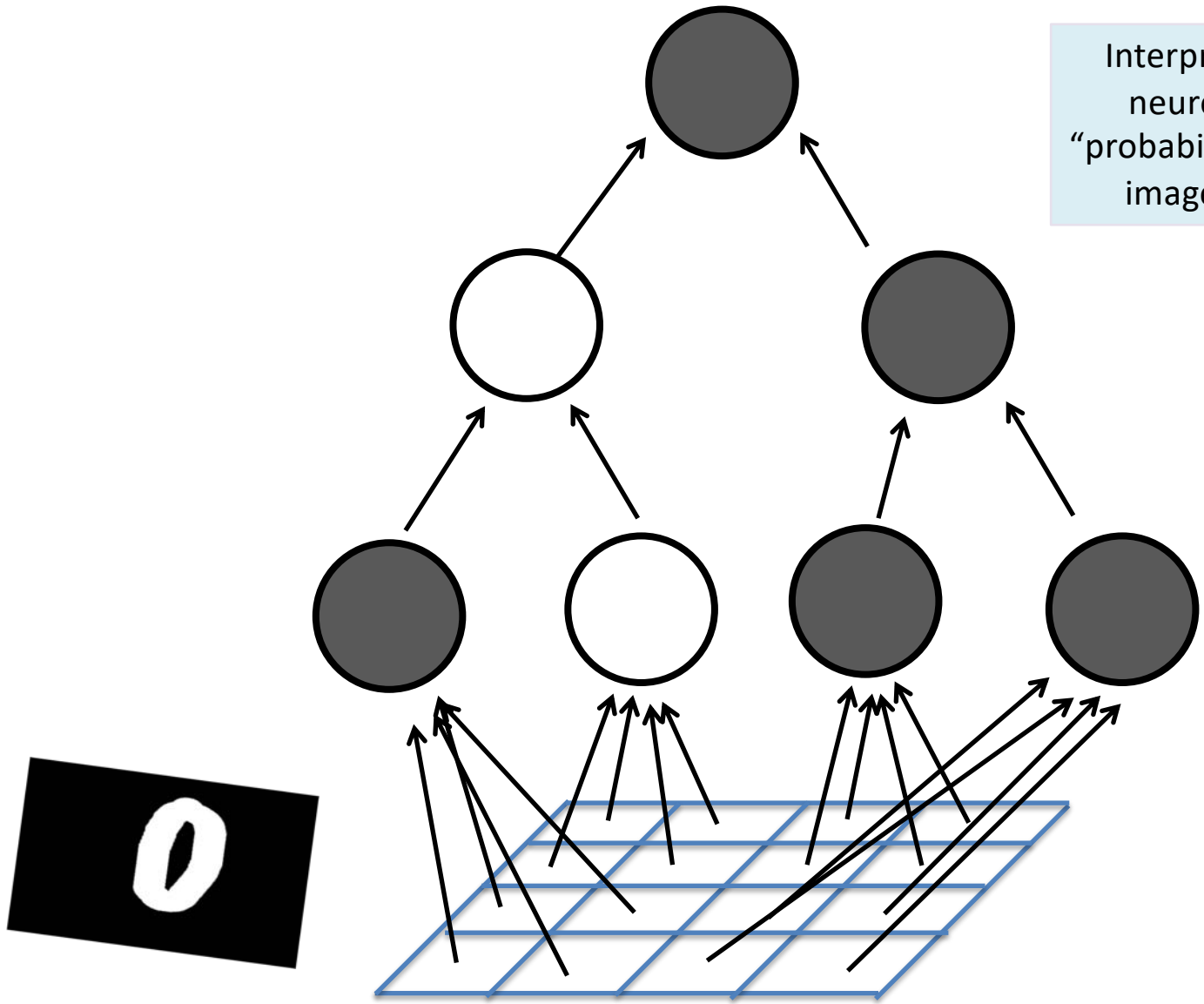


Neural Network

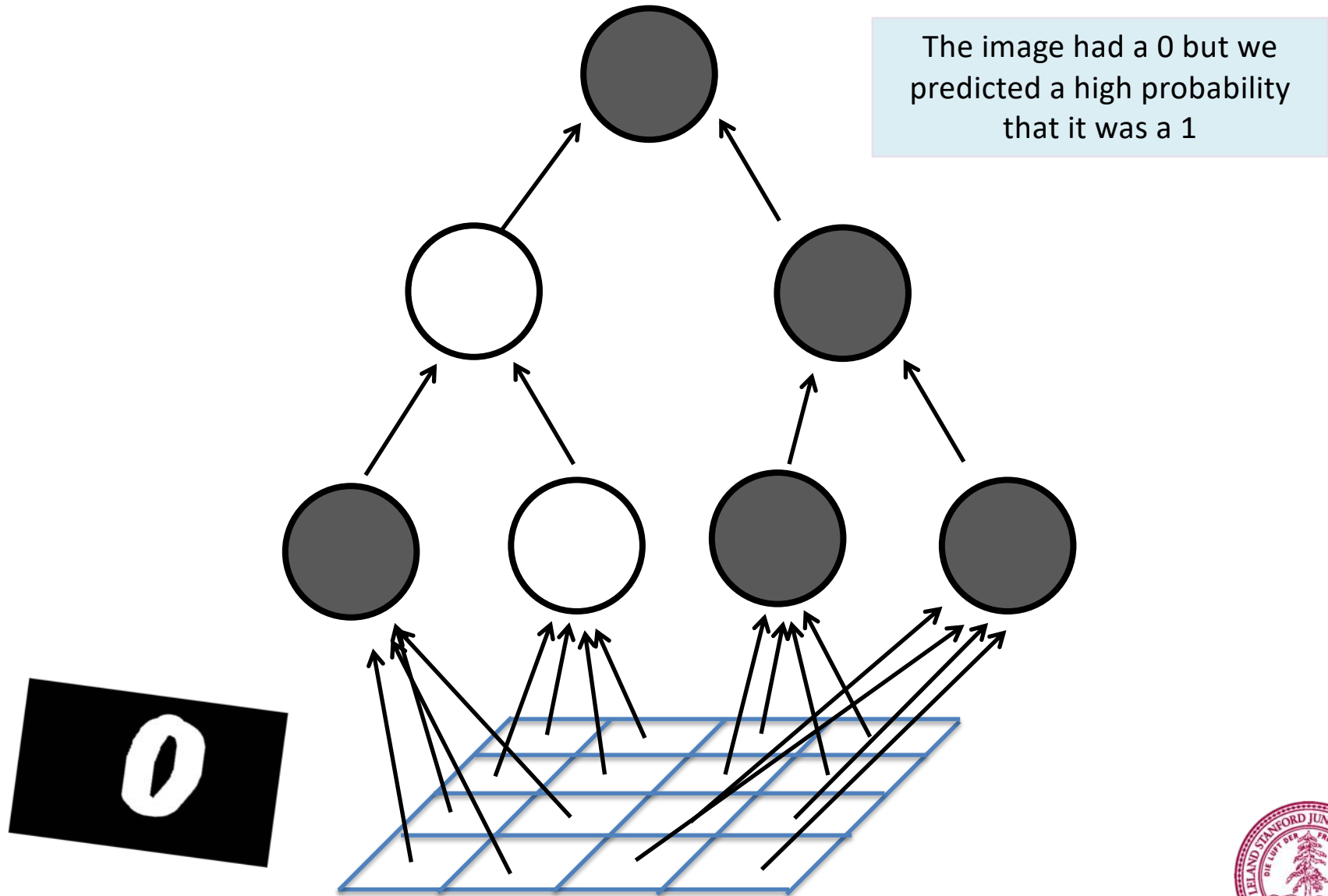


Neural Network

Interpret the last neuron as the “probability” that the image is of a 1



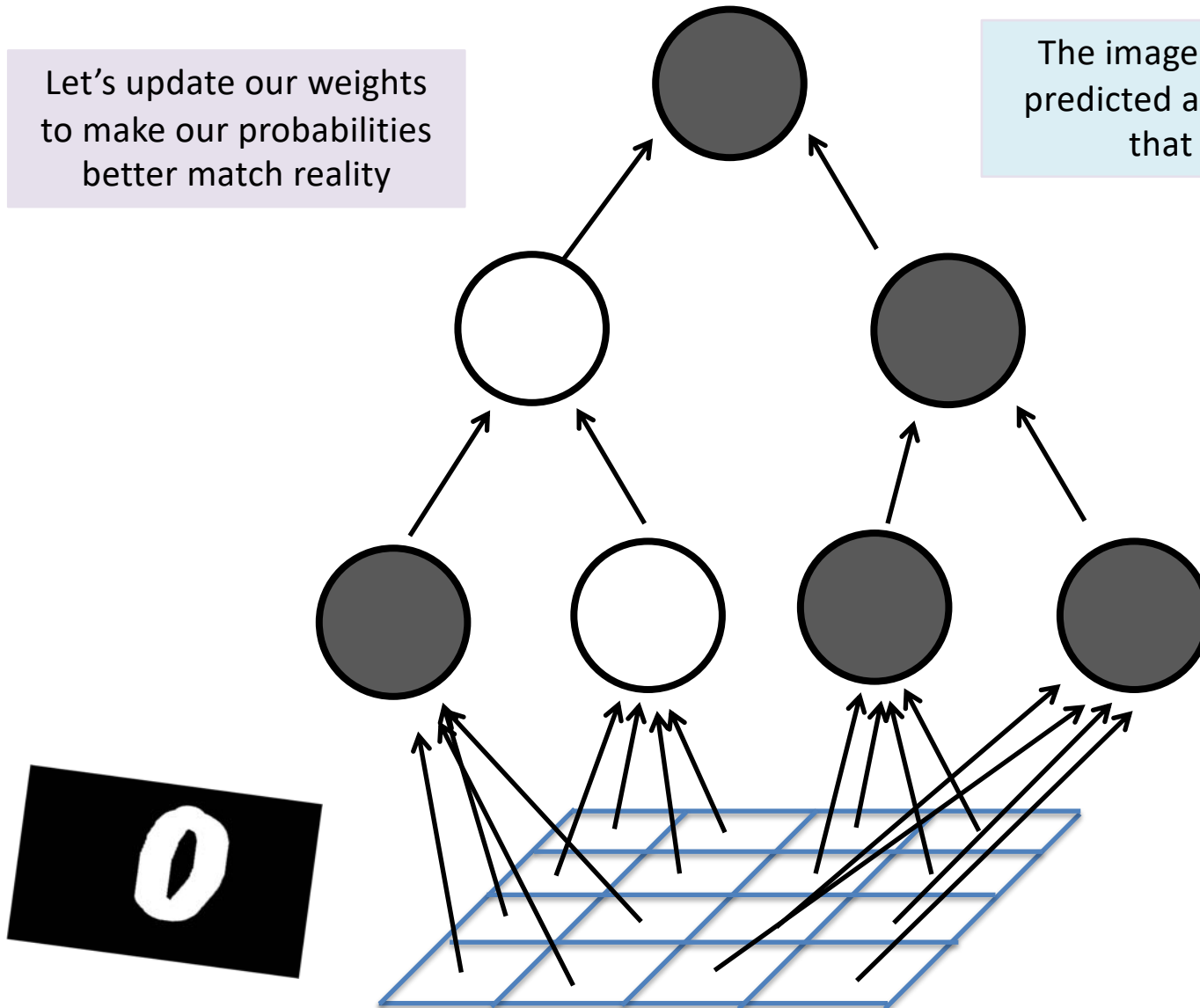
Neural Network



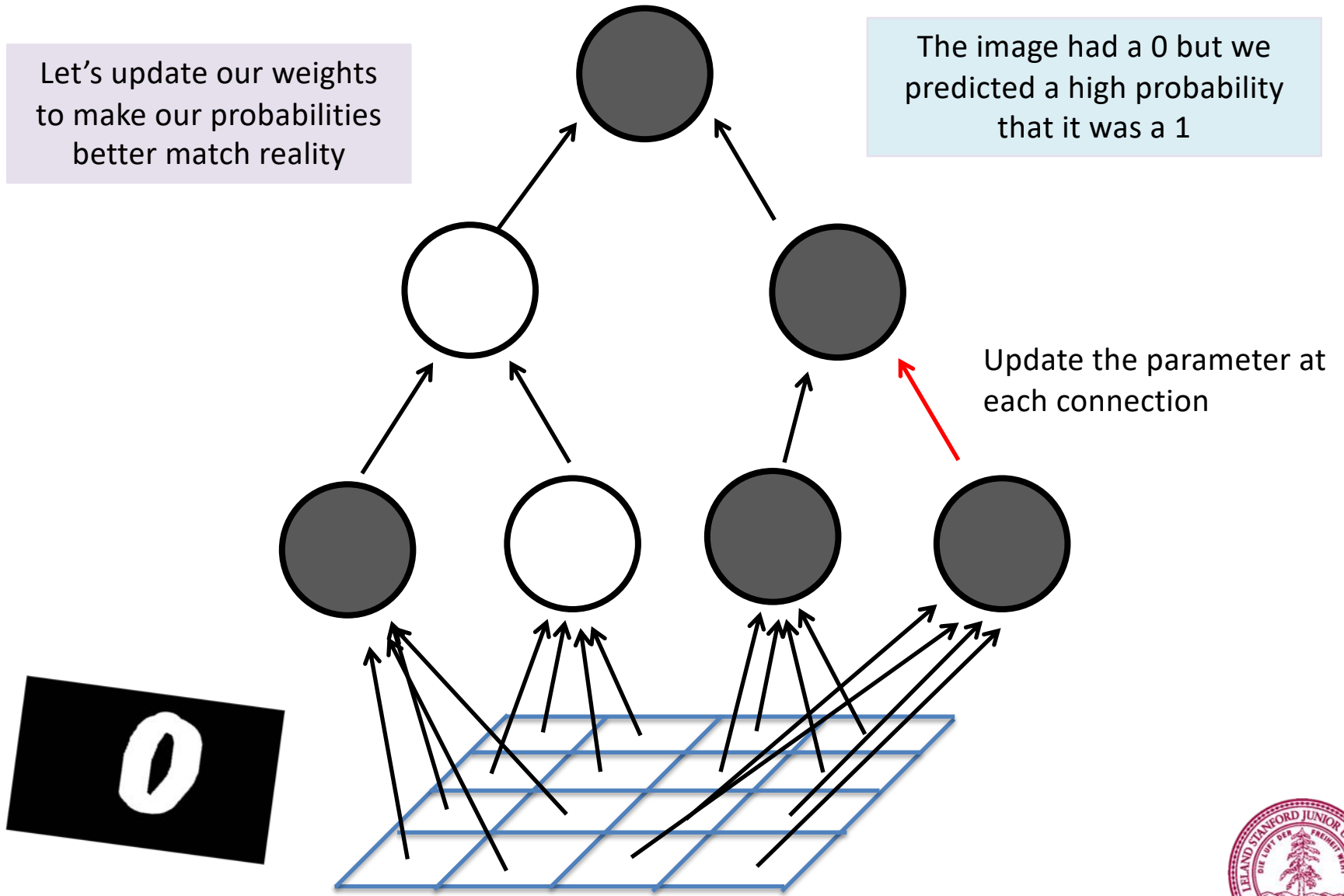
Neural Network

Let's update our weights to make our probabilities better match reality

The image had a 0 but we predicted a high probability that it was a 1



Neural Network



Gradient of output layer params

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

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

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

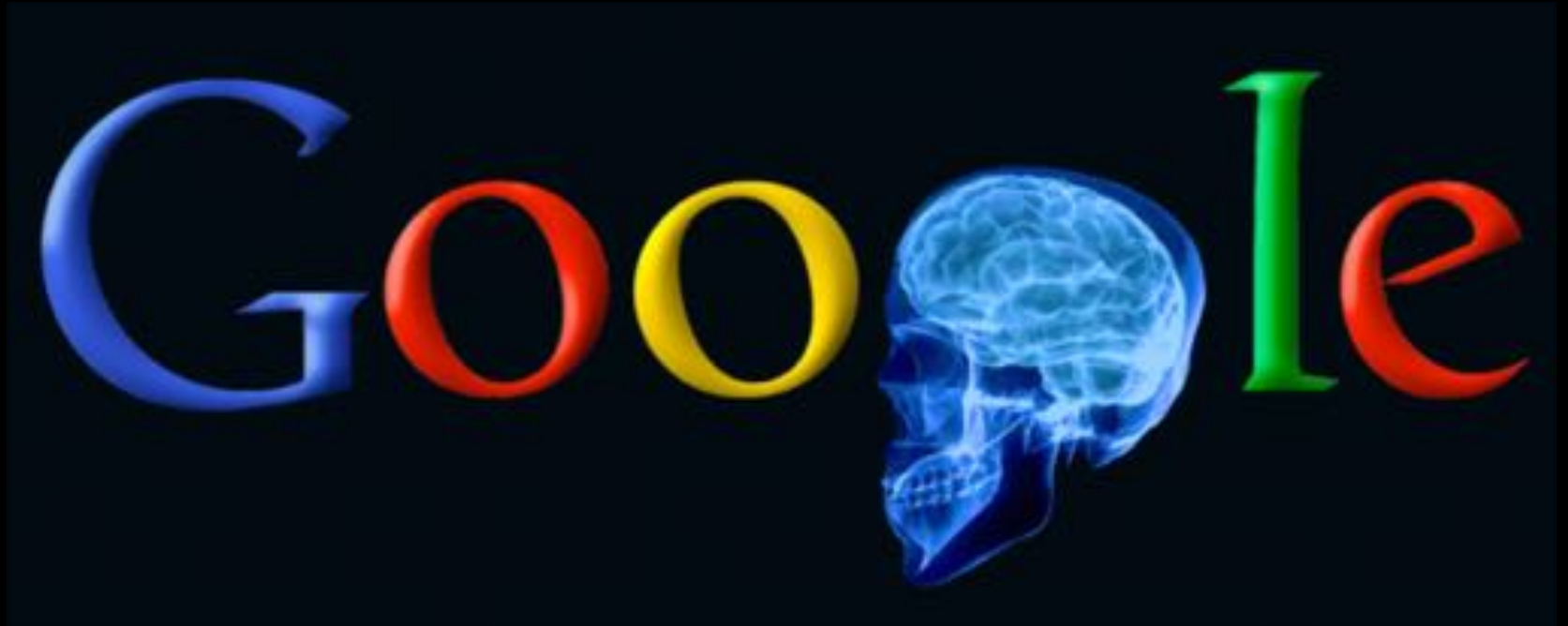
$$= \hat{y} [1 - \hat{y}] \cdot \frac{\partial}{\partial \theta_i^{(\hat{y})}} \sum_{j=0}^{m_h} \mathbf{h}_j \theta_j^{(\hat{y})}$$

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

You will be able to do this.



Google Brain



Google Brain



1 Trillion Artificial Neurons

Other Neurons

Neuron 1



Neuron 2



Neuron 3



Neuron 4



Neuron 5



A Neuron That Fires When It Sees Cats



Top stimuli from the test set



Optimal stimulus
by numerical optimization

ImageNet Classification

...

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

rougtail 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



ImageNet Classification

0.005%

Random guess

1.5%

Pre Neural Networks

?

GoogLeNet

ImageNet Classification

0.005%

Random guess

1.5%

Pre Neural Networks

43.9%

GoogLeNet

ImageNet Classification

0.005%

Random guess

1.5%

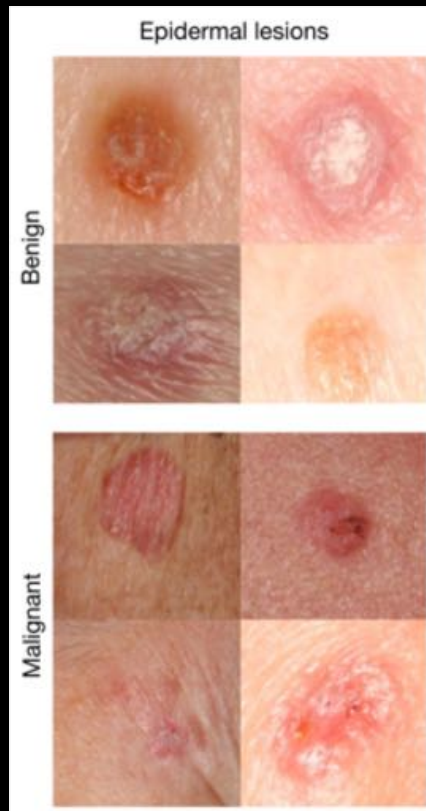
Pre Neural Networks

82.7%

NASNet



Where is this useful?



A machine learning algorithm performs **better than** the best dermatologists.

Developed this year, at Stanford.

Esteva, Andre, et al. "Dermatologist-level classification of skin cancer with deep neural networks." *Nature* 542.7639 (2017): 115-118.

Open Problem: One Shot Learning

B Lake, R Salakhutdinov, J Tenenbaum. Science 2015.

Human-level concept learning through probabilistic program induction.



४	५	६	७	८
९	०	१	२	३

Current deep learning methods are not enough to move the needle as far as we want, **especially on socially relevant problems** that often do not have the benefit of massive public datasets. The best new ideas are coming from probability theory



Prediction: The person who solves one shot learning problem will use core probability

Closest thing to magic you can learn.
Now is the time, Stanford is the place.

