Beyond CS106A

Juliette Woodrow

CS106A, Stanford University

Slides from Piech + Sahami

There is something going on
in the world of AI

[suspense]

Self Driving Cars

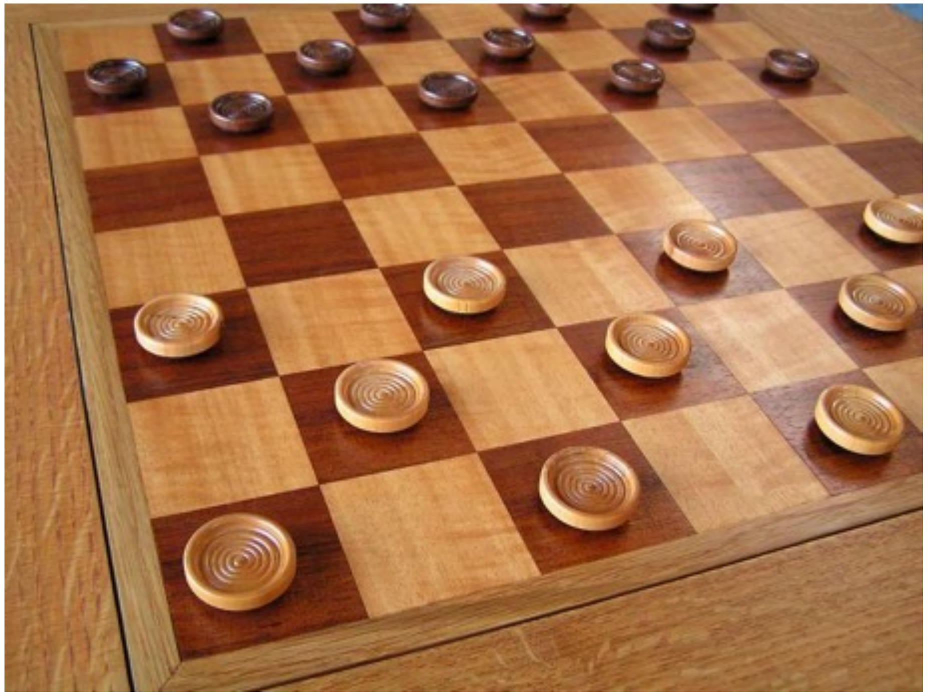
Computers Making Art

Slides from Piech + Sahami

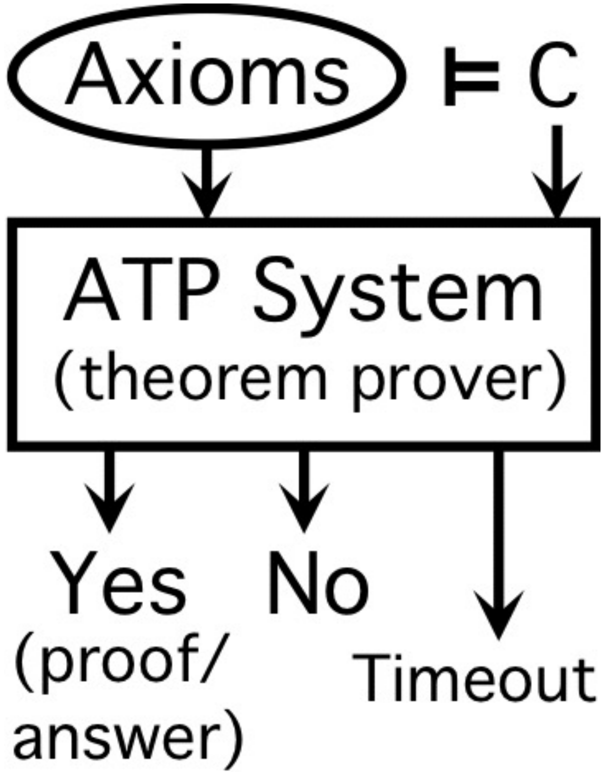
The Last Remaining Board Game

Early Optimism 1950

1952

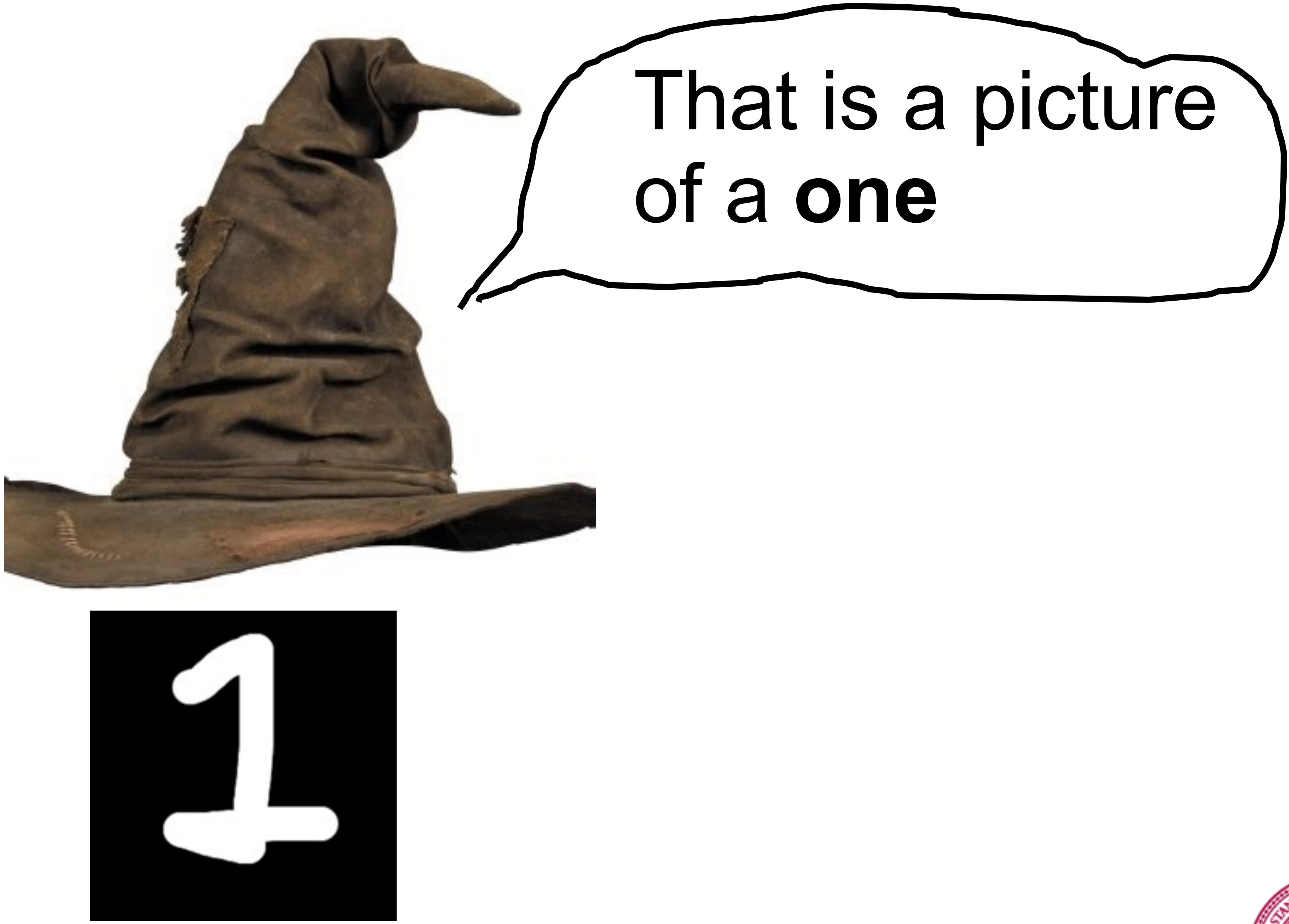


1955

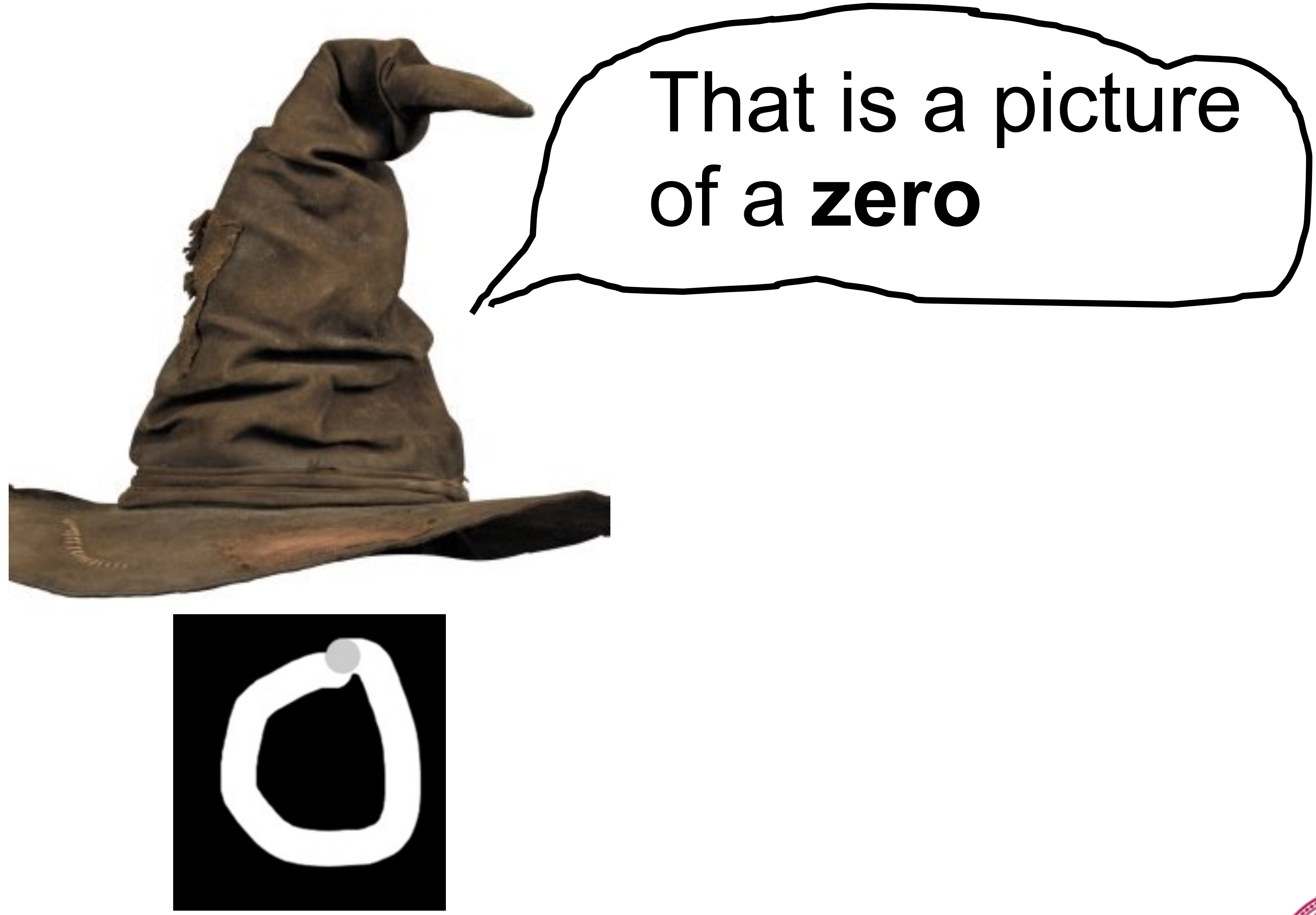


Computer Vision

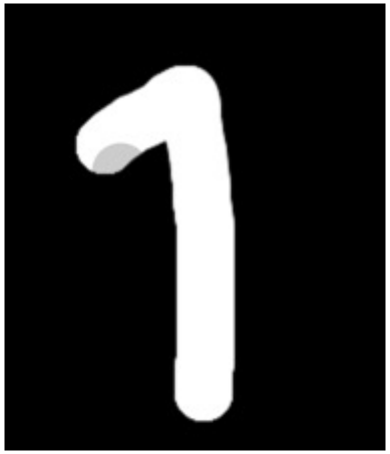
Classification



Classification



Classification



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

Identifying Cats

Here's one way you might code this…

```
def is_cat(image):
  if contains_two_eyes(image):
    if has_whiskers(image):
      if has_pointy_ears(image):
        return True
  return False
```

Identifying Cats

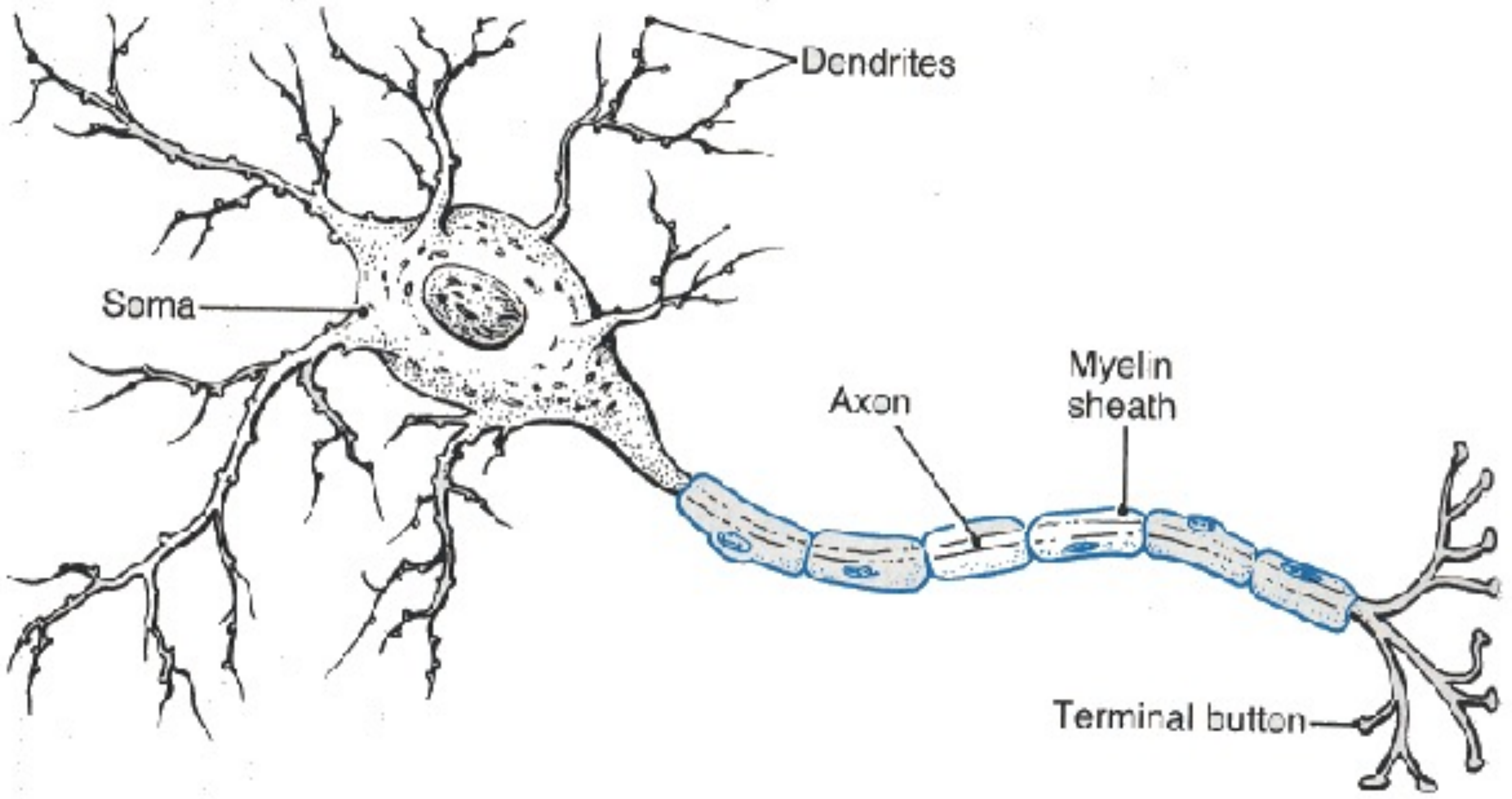
Here's one way you might code this…

```
def is_cat(image):
  if not contains_two_eyes(image):
    return False
  if not has_whiskers(image):
    return False
  if not has_pointy_ears(image):
    return False
  return True
```

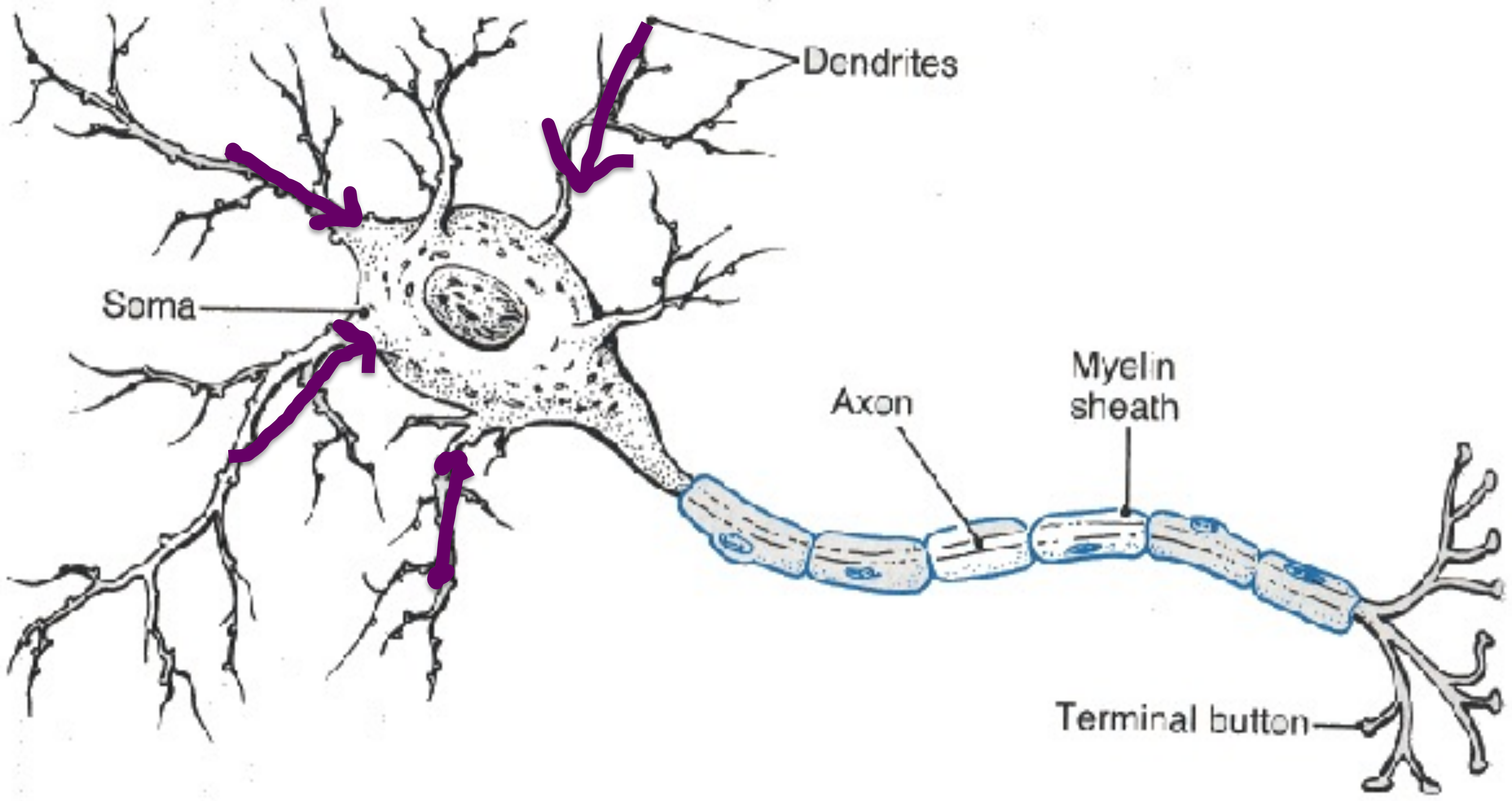
Some Tricky Cases

Great idea inspired by biology

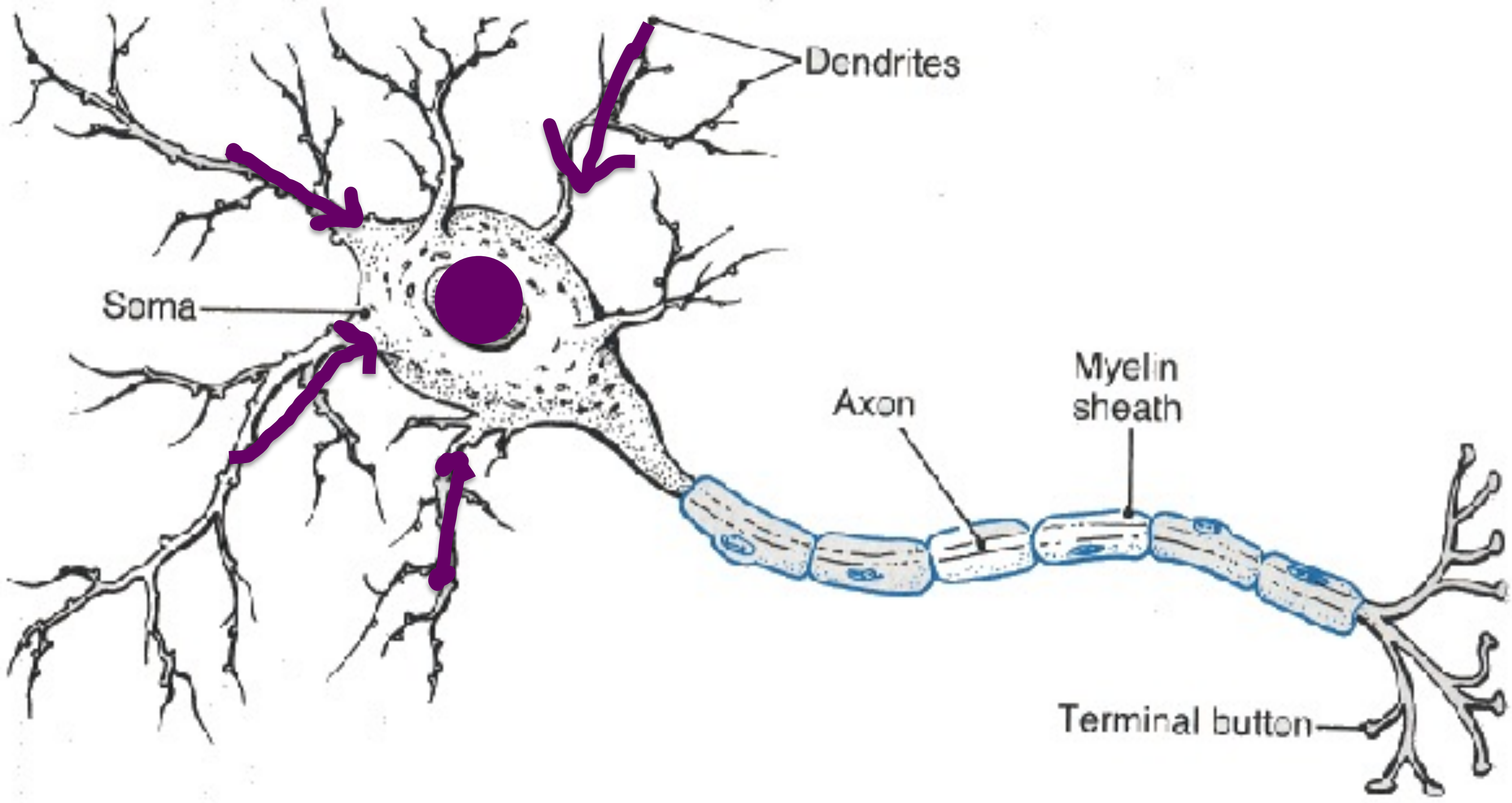
Neuron



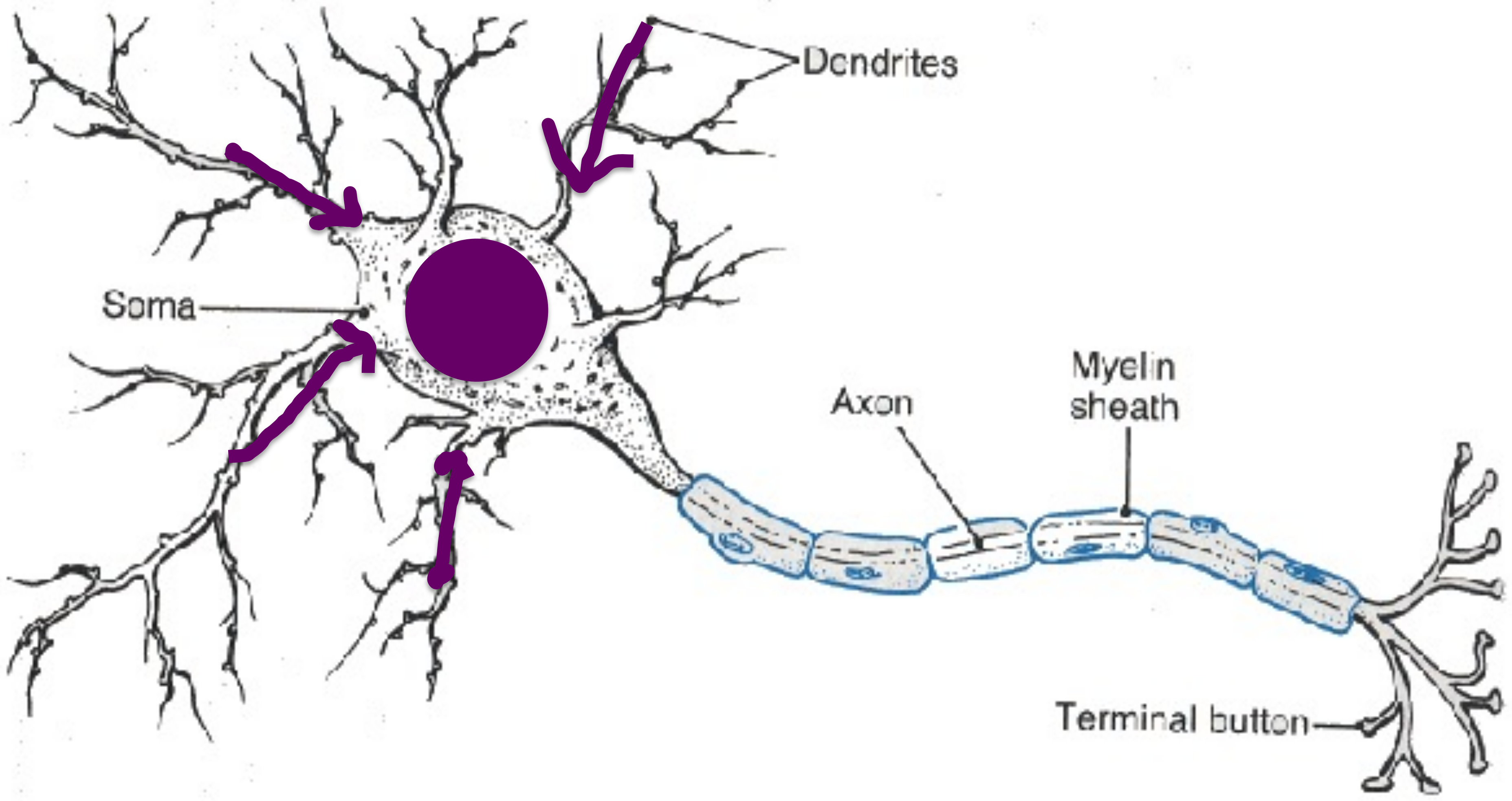
Neuron



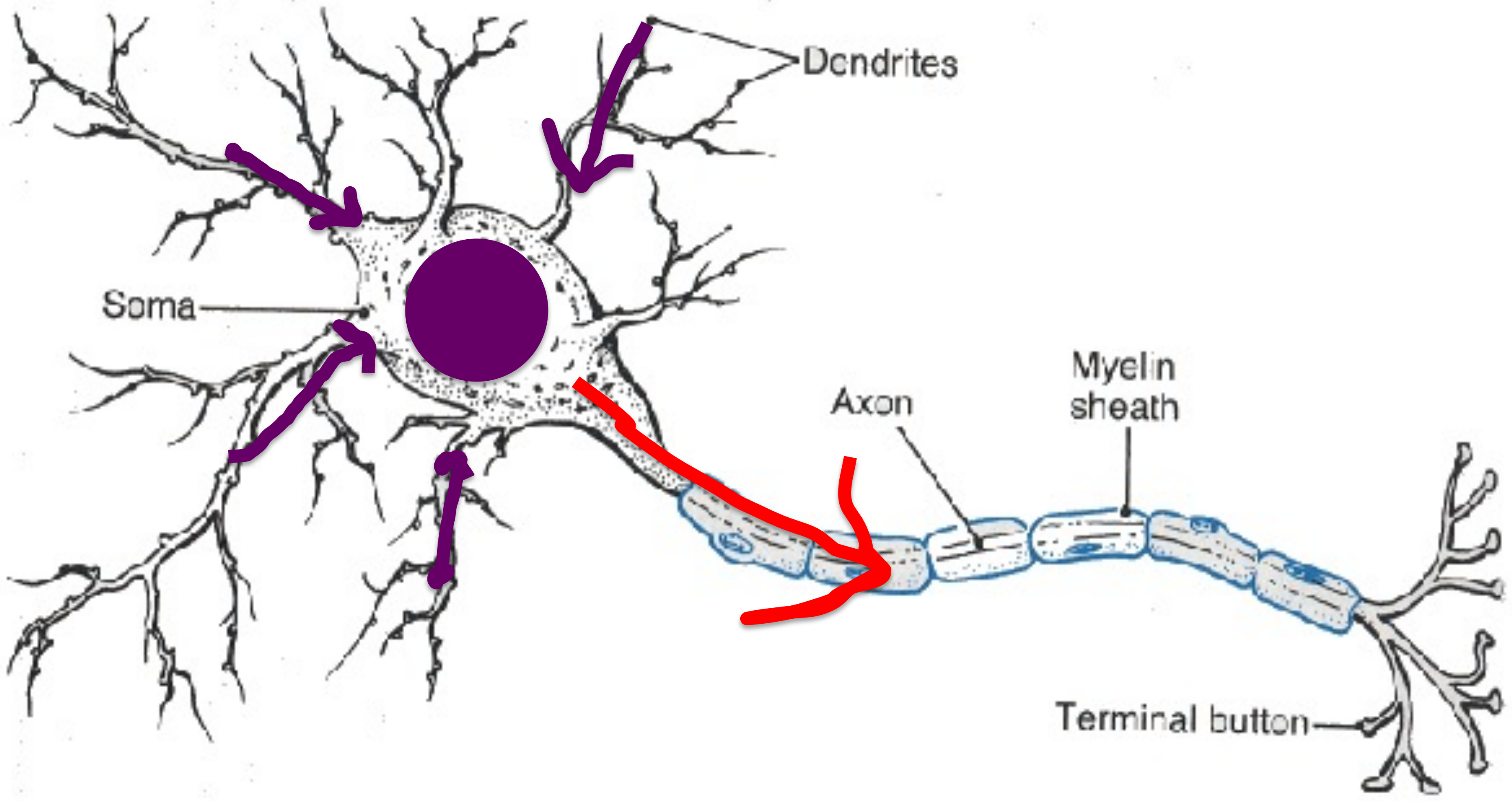
Neuron



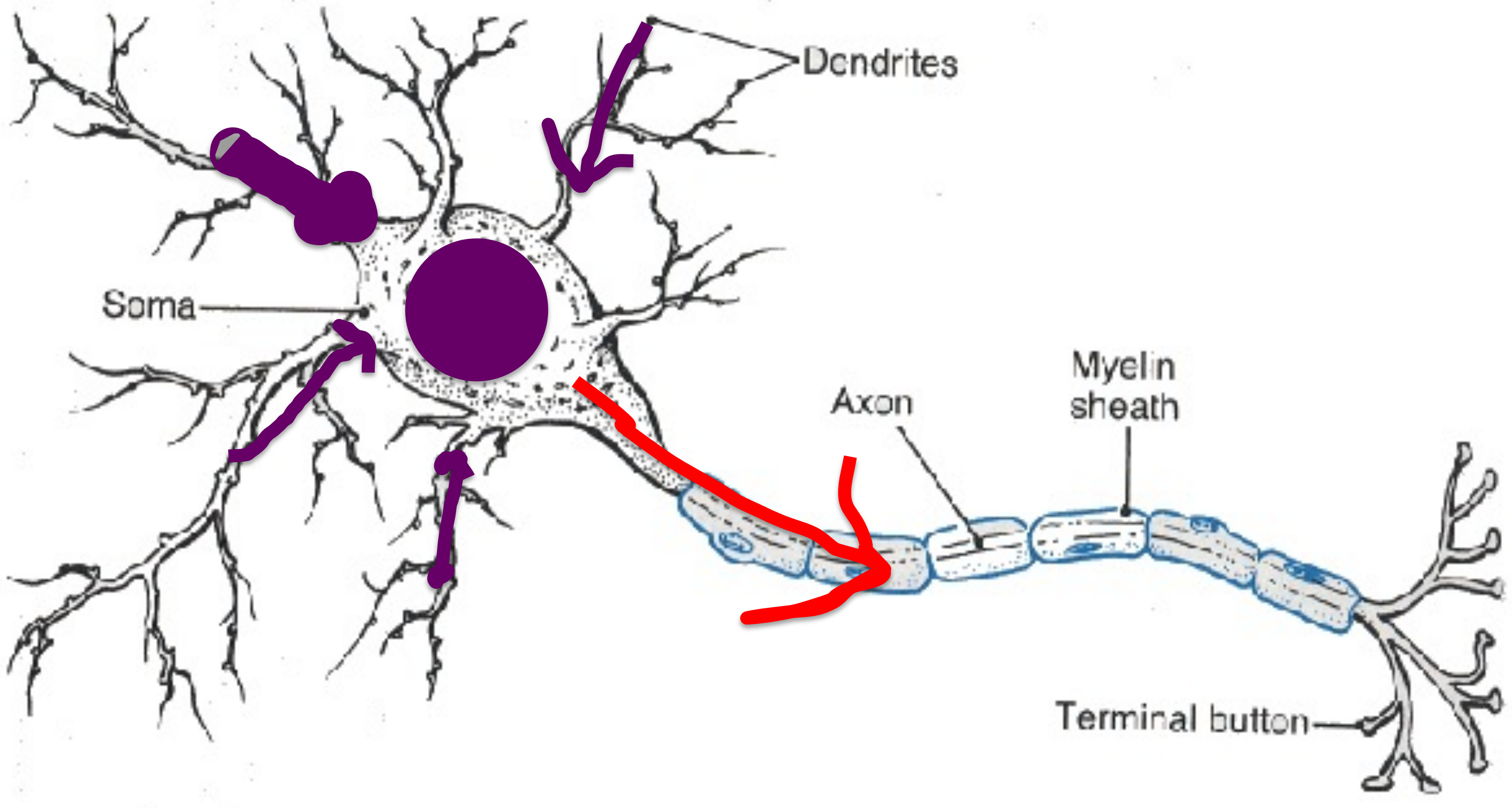
Neuron



Neuron

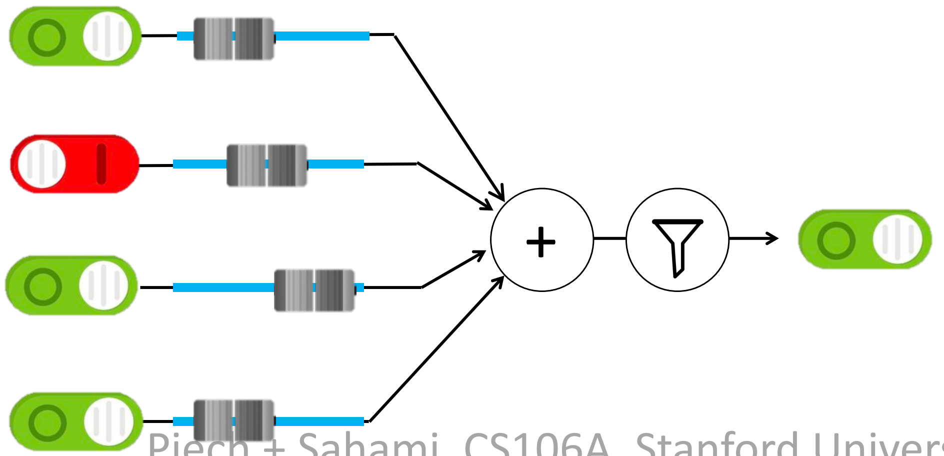


Some Inputs are More Important



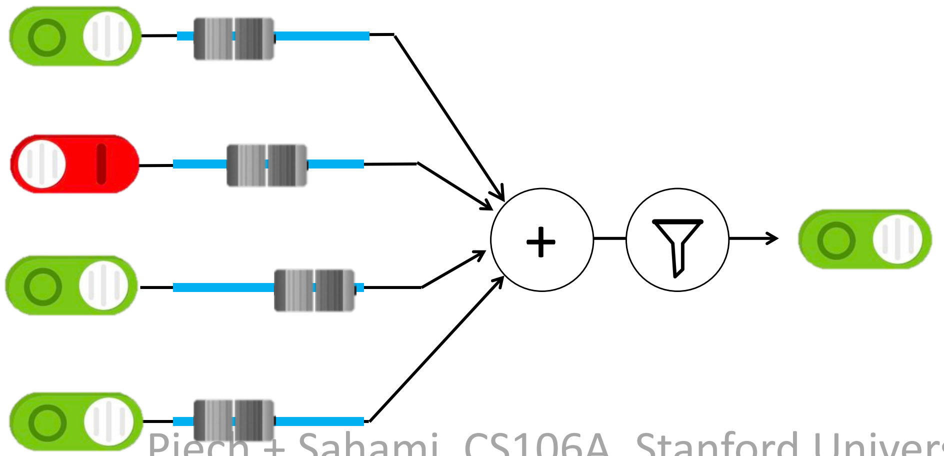
Artificial Neuron

```python
# calculate the activation of a neuron
def activate(weights_list, inputs_list):
    n = len(inputs_list)
    weighted_sum = 0
    for i in range(n):
       weighted_sum += weights_list[i] * inputs_list[i]

    return squash(weighted_sum)



# the sigmoid function forces a value to be between 0 and 1
def squash(value):
    return 1 / (1 + math.exp(-value));
```



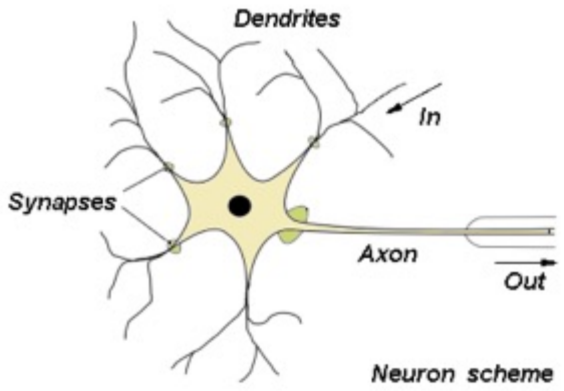
Artificial Neuron

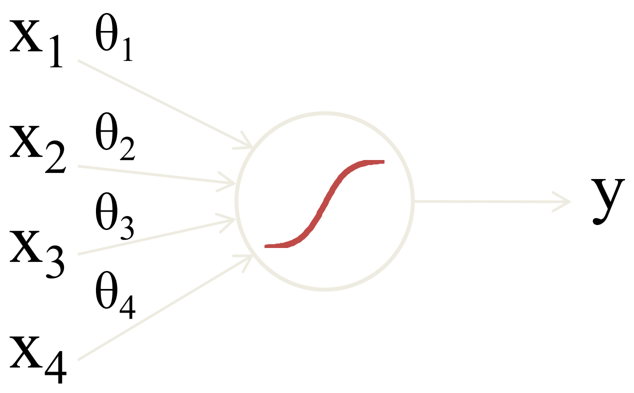
```
# calculate the activation of a neuron
def activate(weights_list, inputs_list):
    n = len(inputs_list)
    # using list comprehensions
    weighted = [weights_list[i] * inputs_list[i] for i in range(n)]
    weighted_sum = sum(weighted)
    return squash(weighted_sum)



# the sigmoid function forces a value to be between 0 and 1
def squash(value):
    return 1 / (1 + math.exp(-value));
```



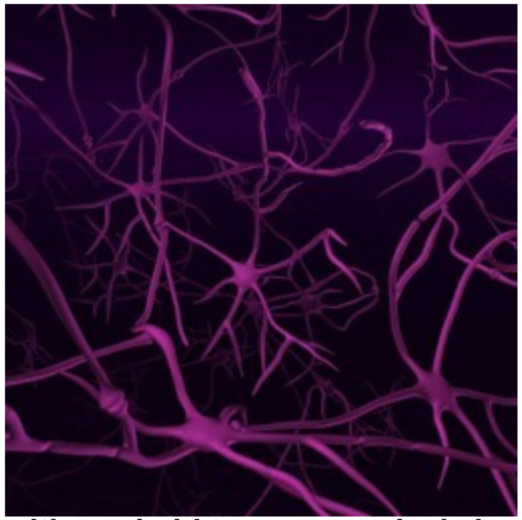
Biological Basis for Neural Networks

- A neuron

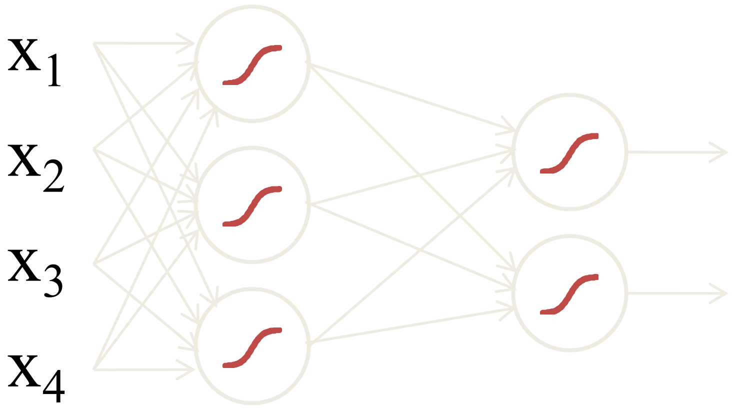




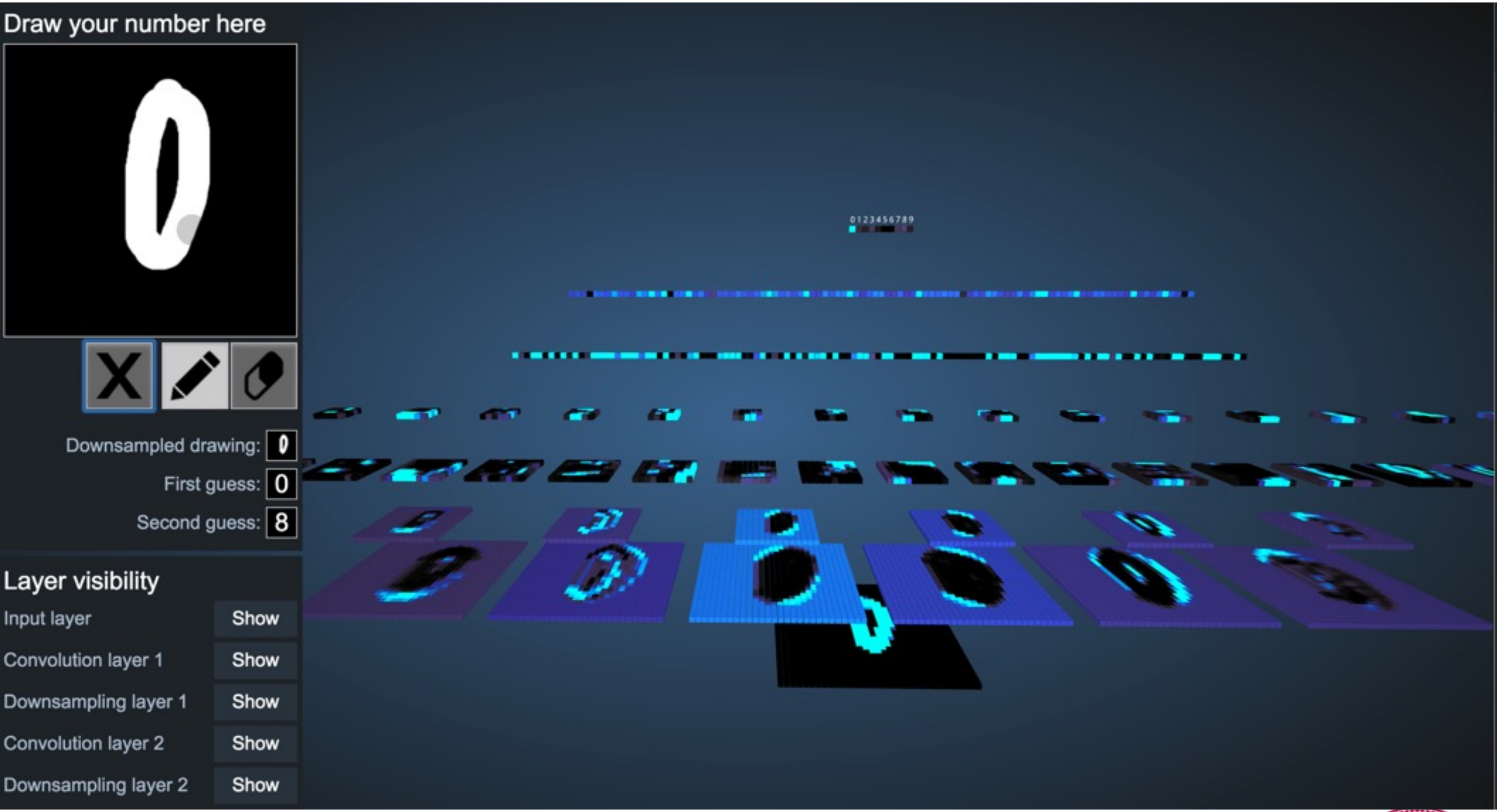
- Your brain



Actually, it's probably someone else's brain



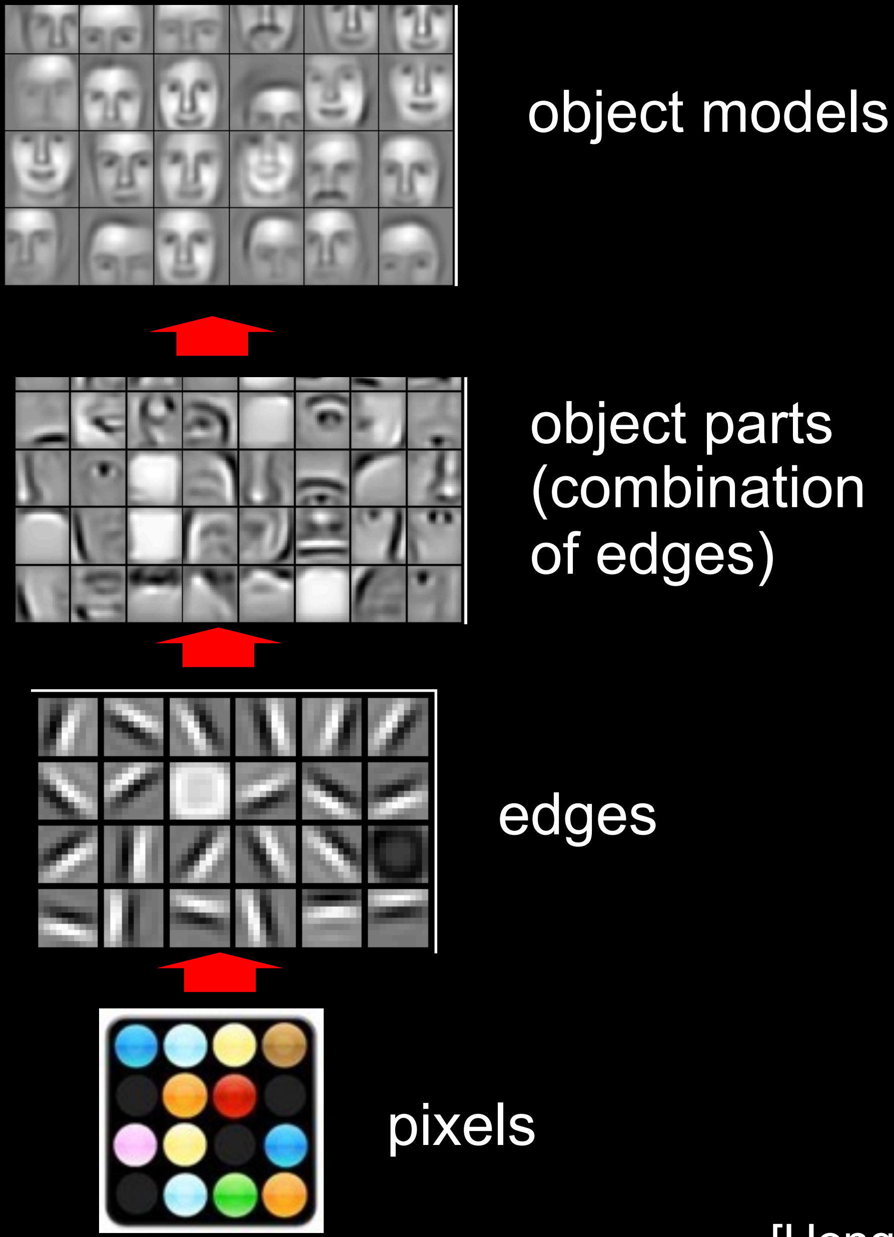
Demonstration



http://scs.ryerson.ca/~aharley/vis/conv/

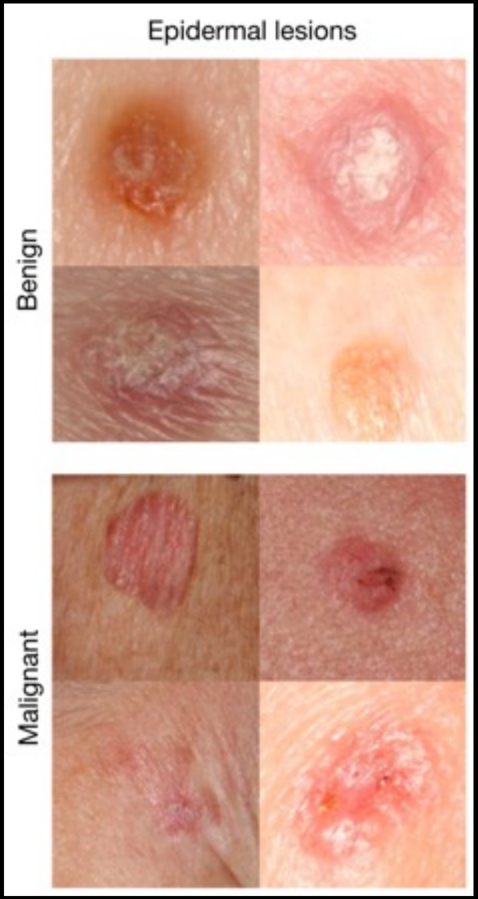
Visualize the Weights

Training set: Aligned images of faces.



[Honglak Lee]

Where is this useful?

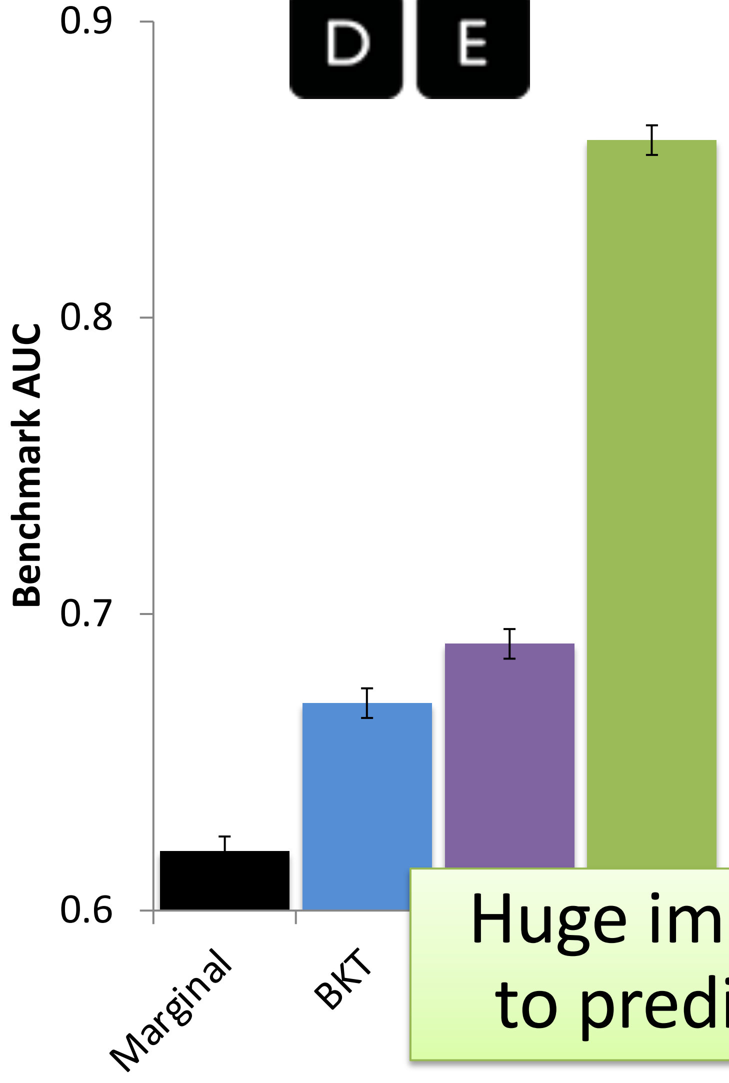


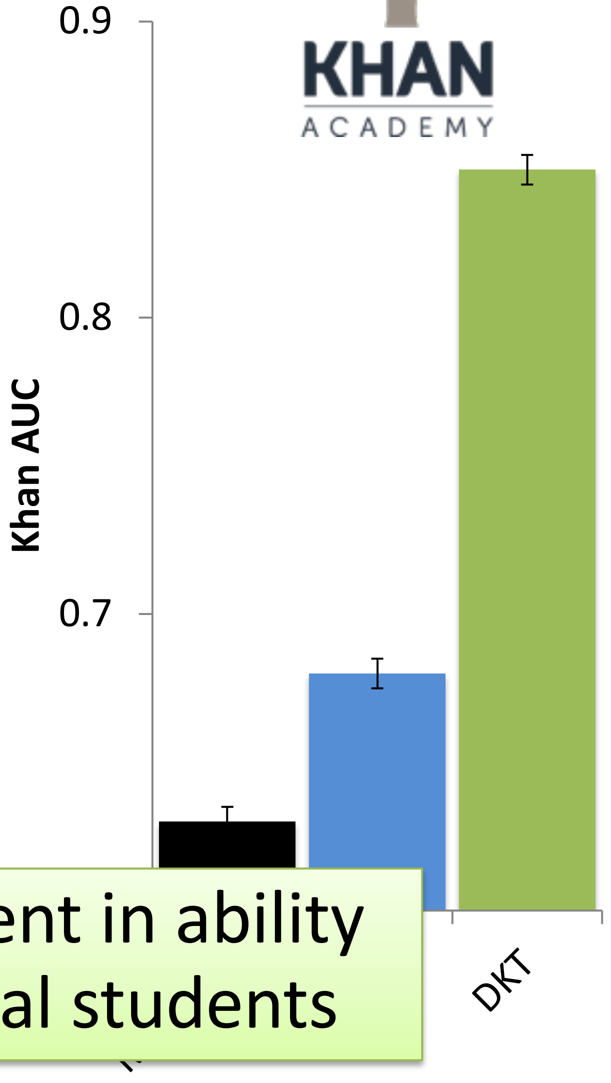
A machine learning algorithm performs **better than** the best dermatologists.

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.

Understanding Students





Huge improvement in ability to predict for real students

1. How to make your own project
2. What other languages look like
3. Deep Learning in Python