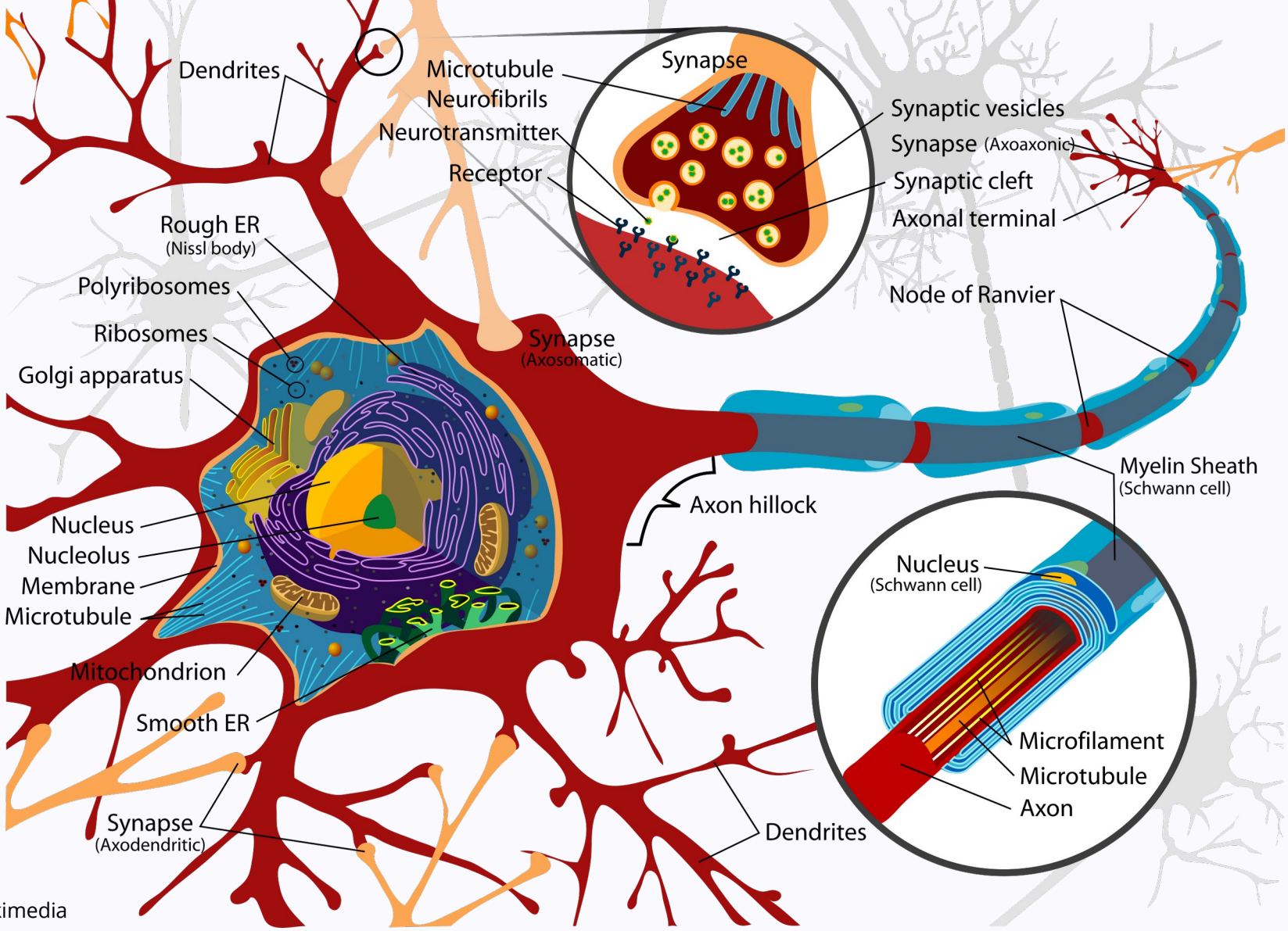
INFO5002: Intro to Python for Info Sys

Perceptron

DSfS 227-229

Slides created by: Zachary Doucet

LVX VERITAS VIRTVS
Northeastern University



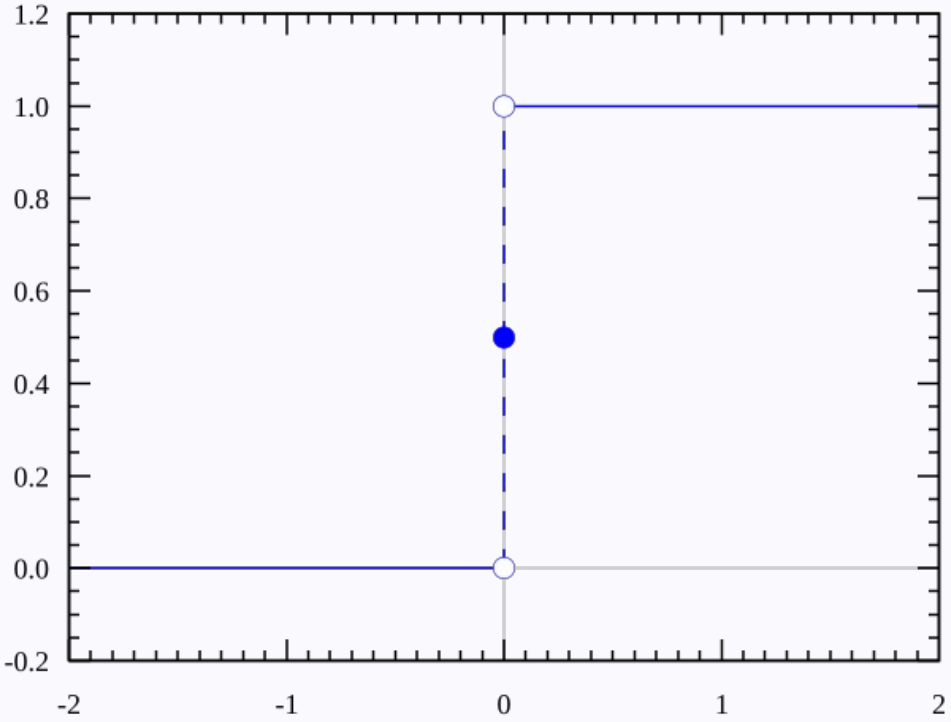
Source: Wikimedia

Perceptron

- Aims to simulate a single neuron.
  - Invented by Warren McCulloch & Walter Pitts
- Frank Rosenblatt simulated the first perceptron on an IBM 704.
- Can think of a logistic function but instead of logistic, you use a step function.

Step Function

- All or nothing function.



Source: Wikimedia

>= 0 then 1
< 0 then 0

Called the
Heaviside function

The perceptron

$$f(x) = h(w \cdot x + b)$$

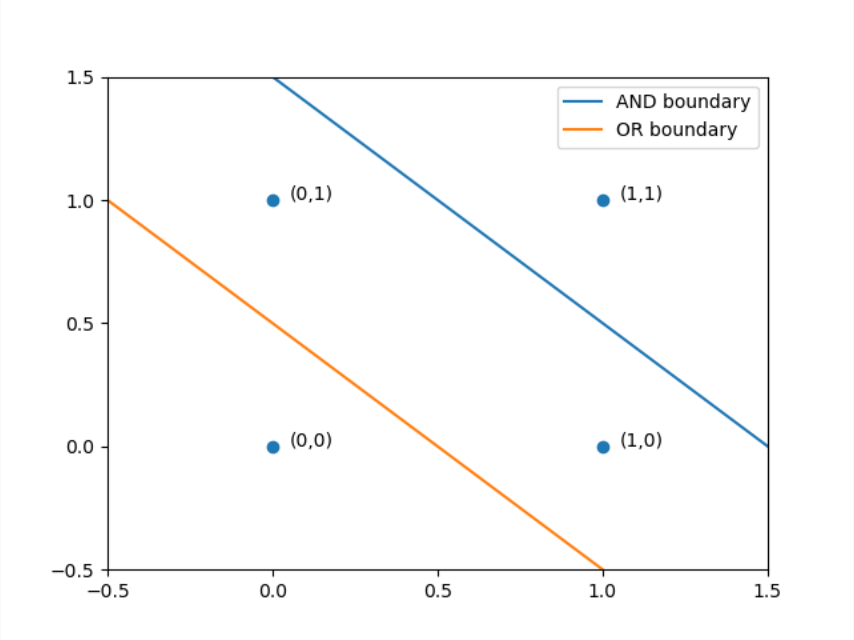
You can pass the bias in like we did w/ MLR

$$f(x) = h(w' \cdot x')$$

$$h(y) = \begin{cases} 0, if\ y < 0 \\ 1, if\ y \geq 0 \end{cases}$$

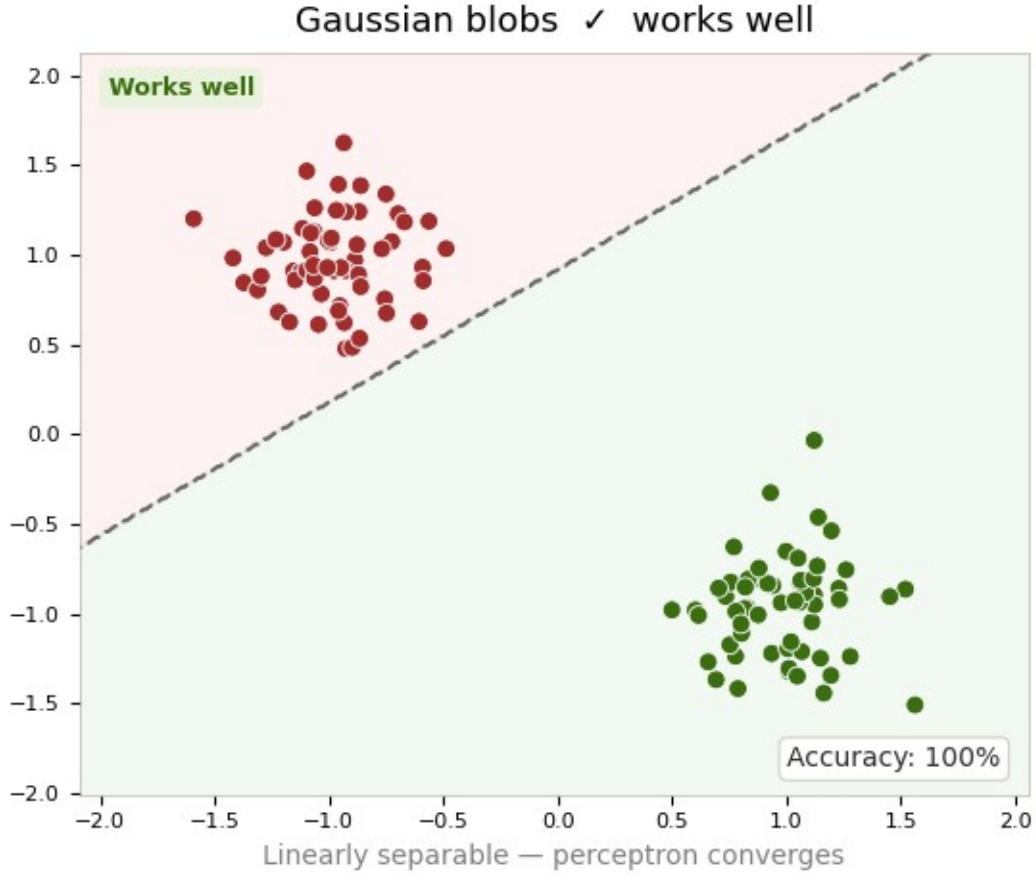
We can solve many problems

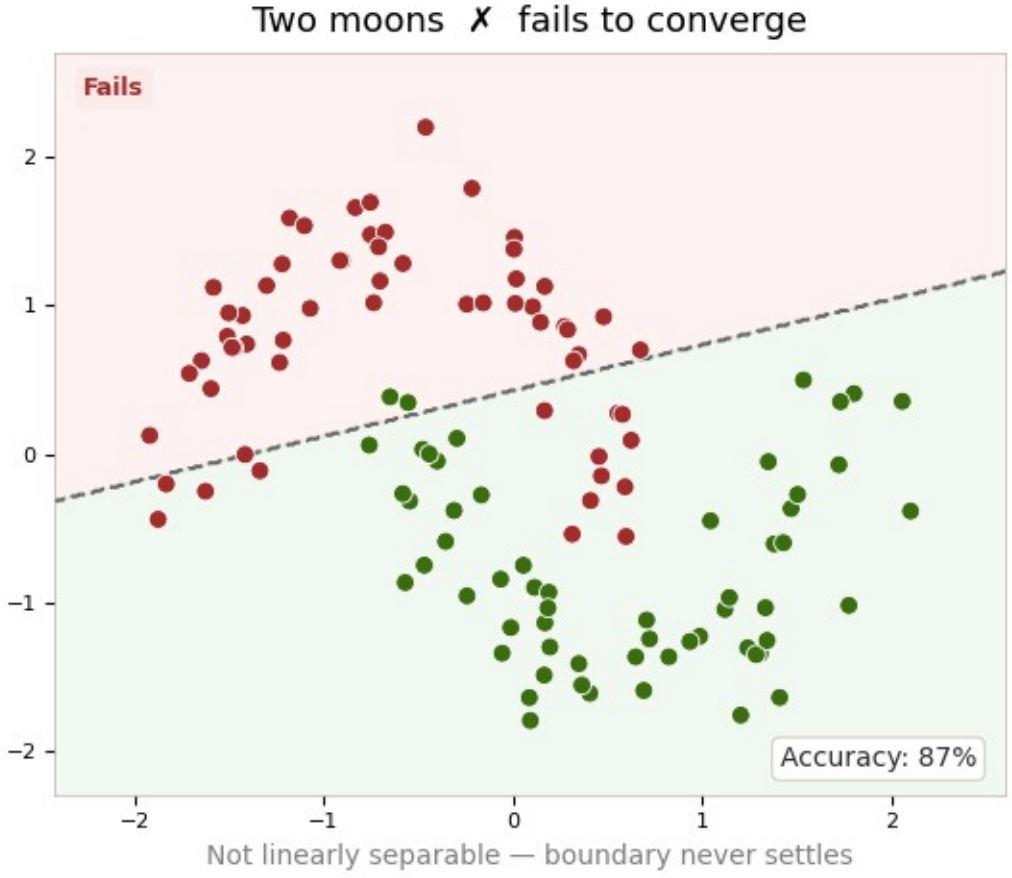
- As long as they are linearly separable.
- You can represent an AND gate, an OR gate, a NOT gate.



But can we solve XOR?

```python
from sklearn.linear_model import Perceptron

model = Perceptron(penalty=None, alpha=0.0001, l1_ratio=0.15,
    fit_intercept=True, max_iter=1000, tol=0.001, shuffle=True,
    verbose=0, eta0=1.0, n_jobs=None, random_state=0,
    early_stopping=False, validation_fraction=0.1,
    n_iter_no_change=5, class_weight=None, warm_start=False)

# To train the model
model.fit(X_train, y_train)

# Number weight updates
model.t_

# Predict confidence scores (n_samples, n_classes)
scores = model.decision_function(X_test)

...
```

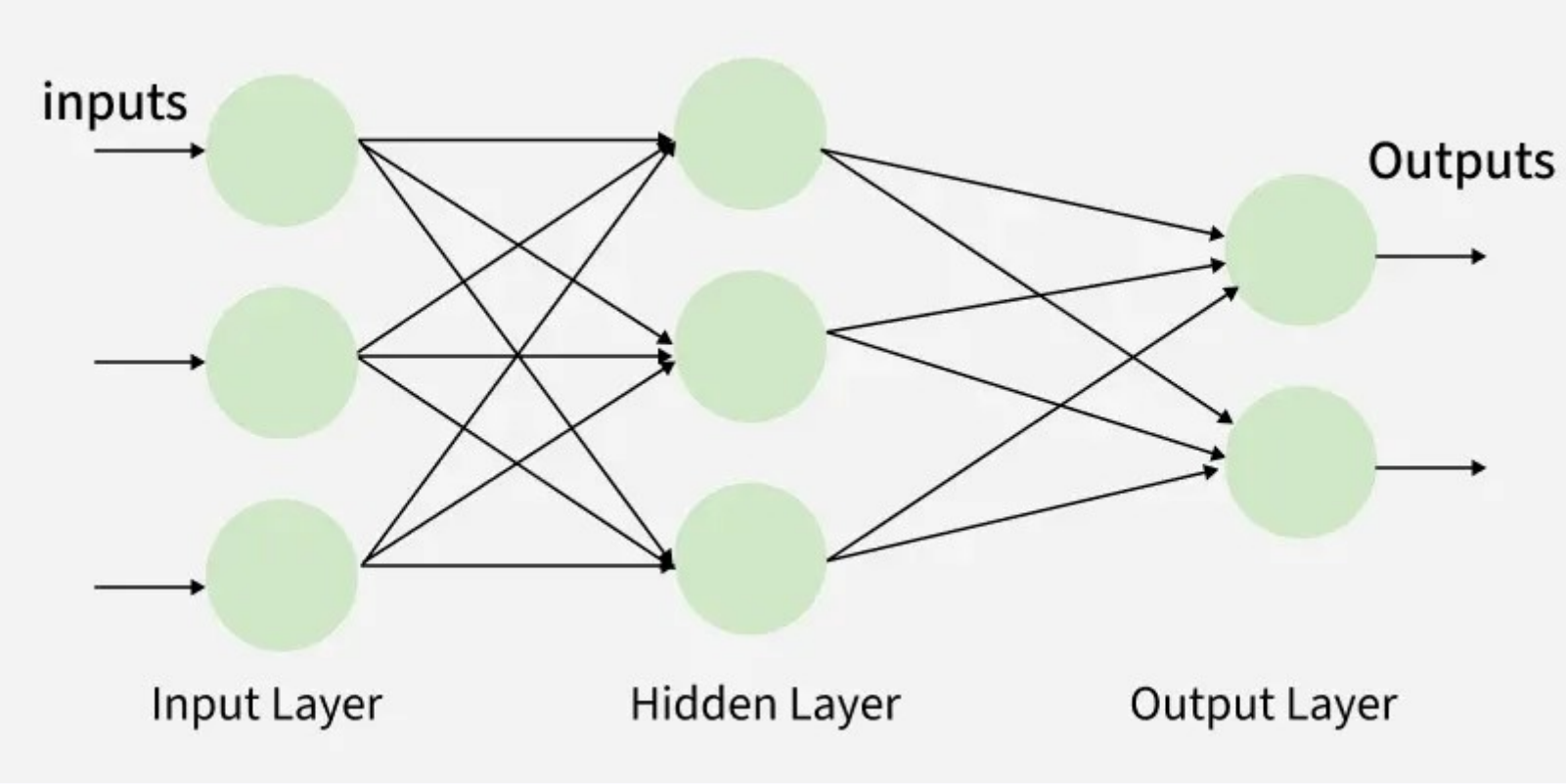

Gaussian blobs ✓ works well
Works well
Accuracy: 100%
Linearly separable — perceptron converges


Two moons ✗ fails to converge
Fails
Accuracy: 87%
Not linearly separable — boundary never settles

MLP

DSfS 227-229

We can solve by stacking



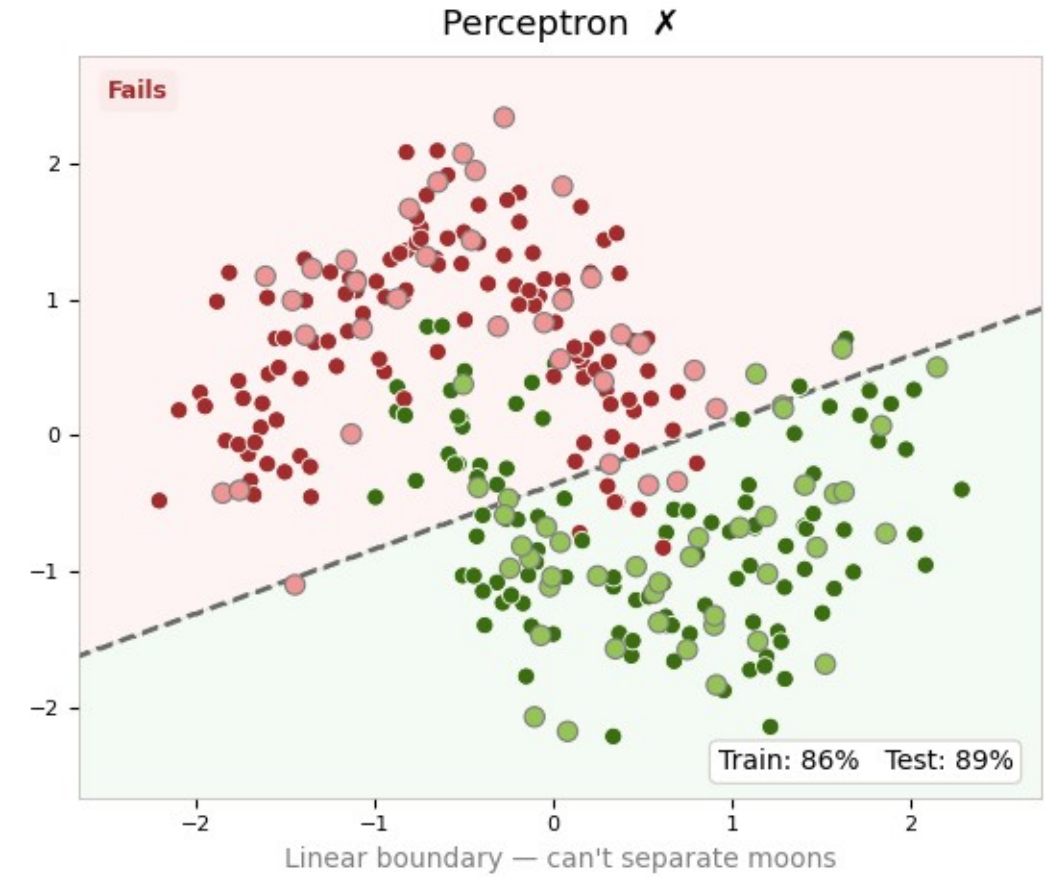
Source: GeekForGeek

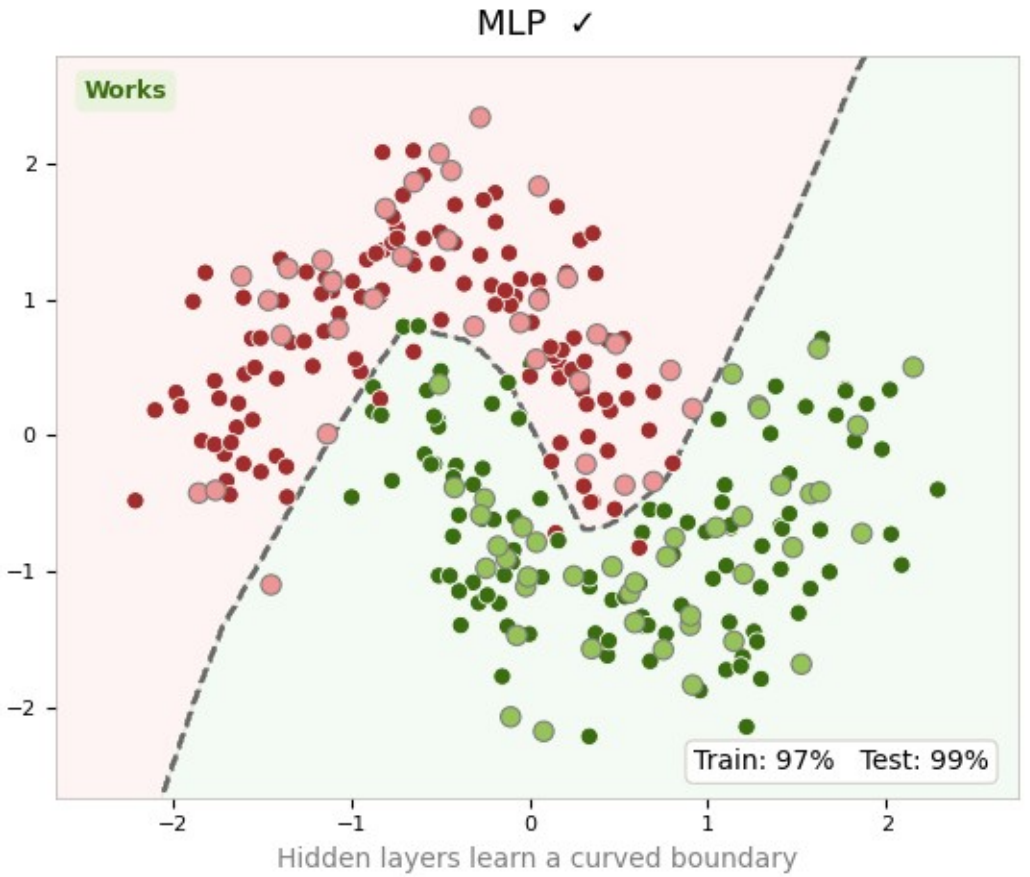
- By combining perceptrons we can create an MLP

Combine

- The perceptron of one can be fed into the next one
- To allow for learning (which requires derivatives) we need to use a continuous function.
  - We can mimic heaviside function w/ sigmoid function
- This simple idea is the foundation of modern ML

```
from sklearn.neural_network import MLPClassifier

model = MLPClassifier(hidden_layer_size=(100,), activation="relu",
    solver="adam", 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=0.001, verbose=False, warm_start=False, momentum=0.9,
    nesterovs_momentum=True, early_stopping=False,
    validation_fraction=0.1, beta_1=0.9, beta_2=0.999,
    epsilon=1e-08, n_iter_no_change=10, max_fun=15000)

# To train the model
model.fit(X_train, y_train)

 ...
```


Perceptron ✗
Fails
Train: 86% Test: 89%
Linear boundary — can't separate moons


MLP ✓
Works
Train: 97% Test: 99%
Hidden layers learn a curved boundary

That’s it!

This is all the material for the class! It has been a pleasure teaching you all!

Reminder

- Next week exam @ 10 am followed by project OH
  - 2 hour closed book exam like previous ones
- Projects due Sun Apr 26th by 11:59pm PST