Artificial Neurons

Danna Gurari
University of Colorado Boulder
Spring 2025

https://dannagurari.colorado.edu/course/neural-networks-and-deep-learning-spring-2025/

Review

- Last lecture:
  - Deep learning applications
  - History of neural networks and deep learning
  - How does a machine learn?
  - Course logistics

- Please keep up with assigned readings posted to course website

- Questions?

Today’s Topics

- Supervised learning: approach to develop a model
- Artificial neuron model: basic unit of neural networks
- Evaluating classification models

Today’s Topics

- Supervised learning: approach to develop a model
- Artificial neuron model: basic unit of neural networks
- Evaluating classification models

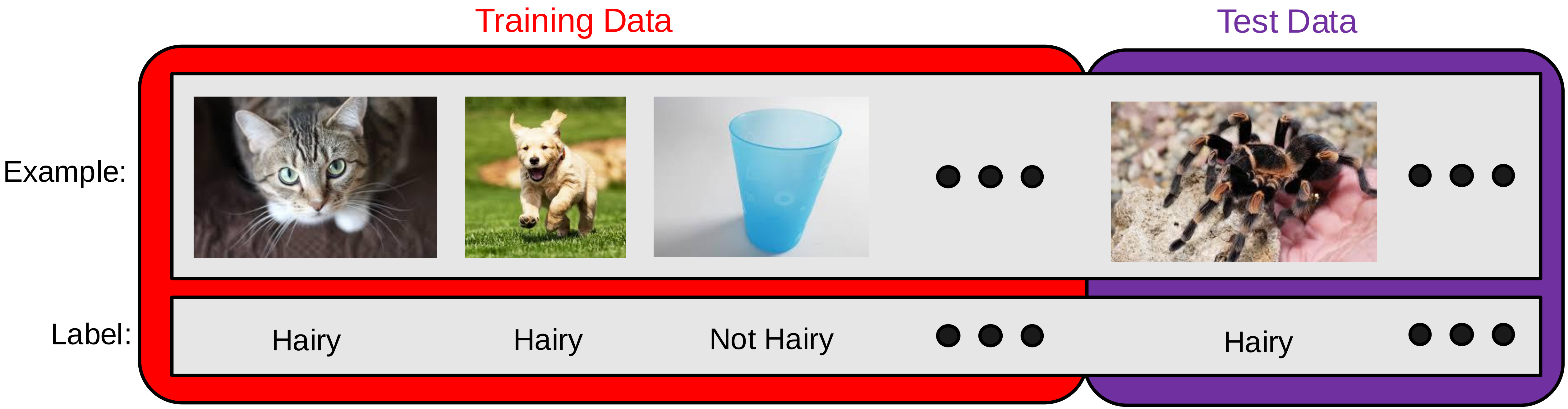
Goal: Design Models that **Generalize** Well to New, Previously Unseen Examples

Example:

Label: Hairy Hairy Not Hairy … Hairy …

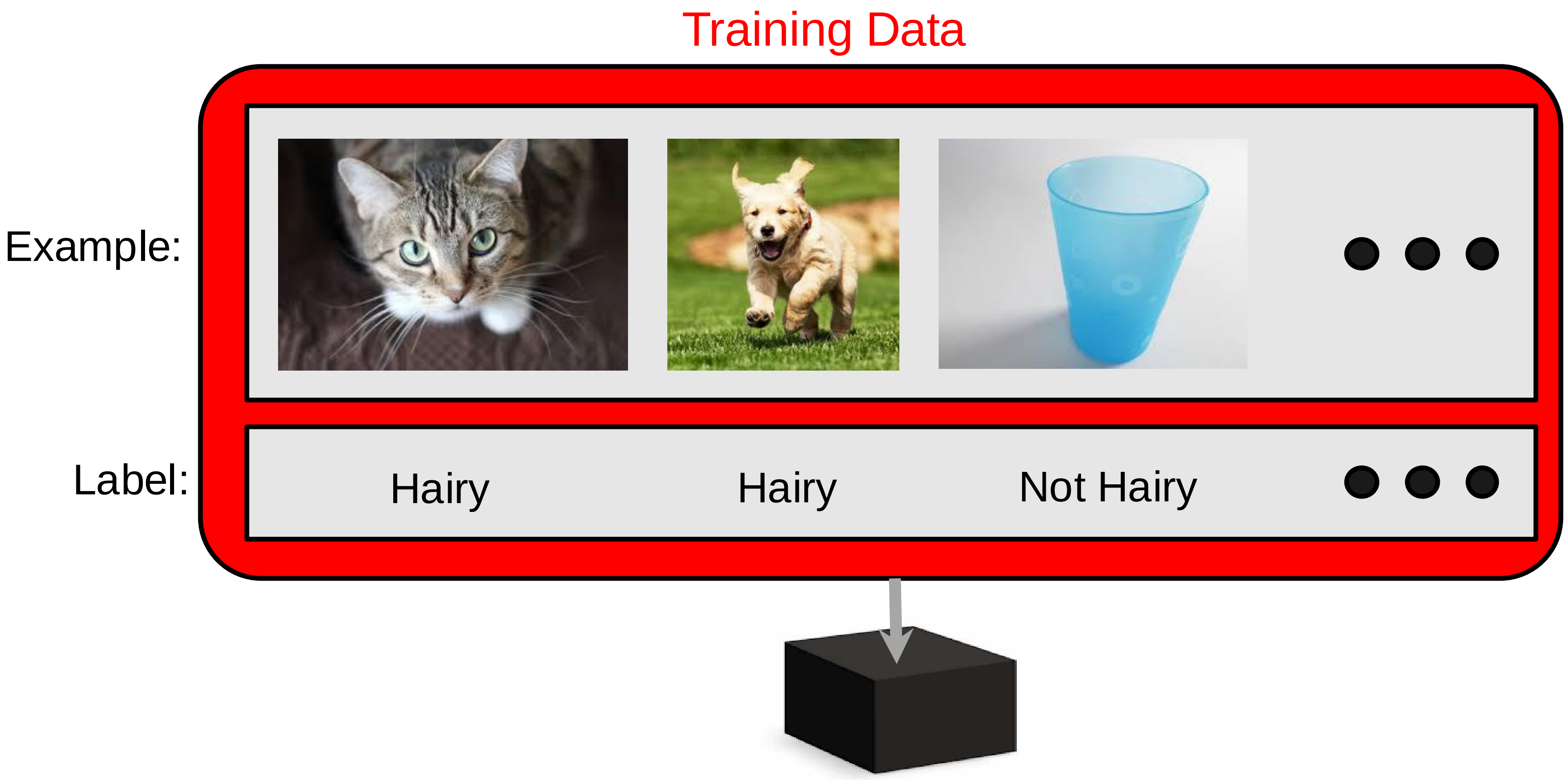
Goal: Design Models that **Generalize** Well to New, Previously Unseen Examples

1. Split data into a "training set" and "test set "



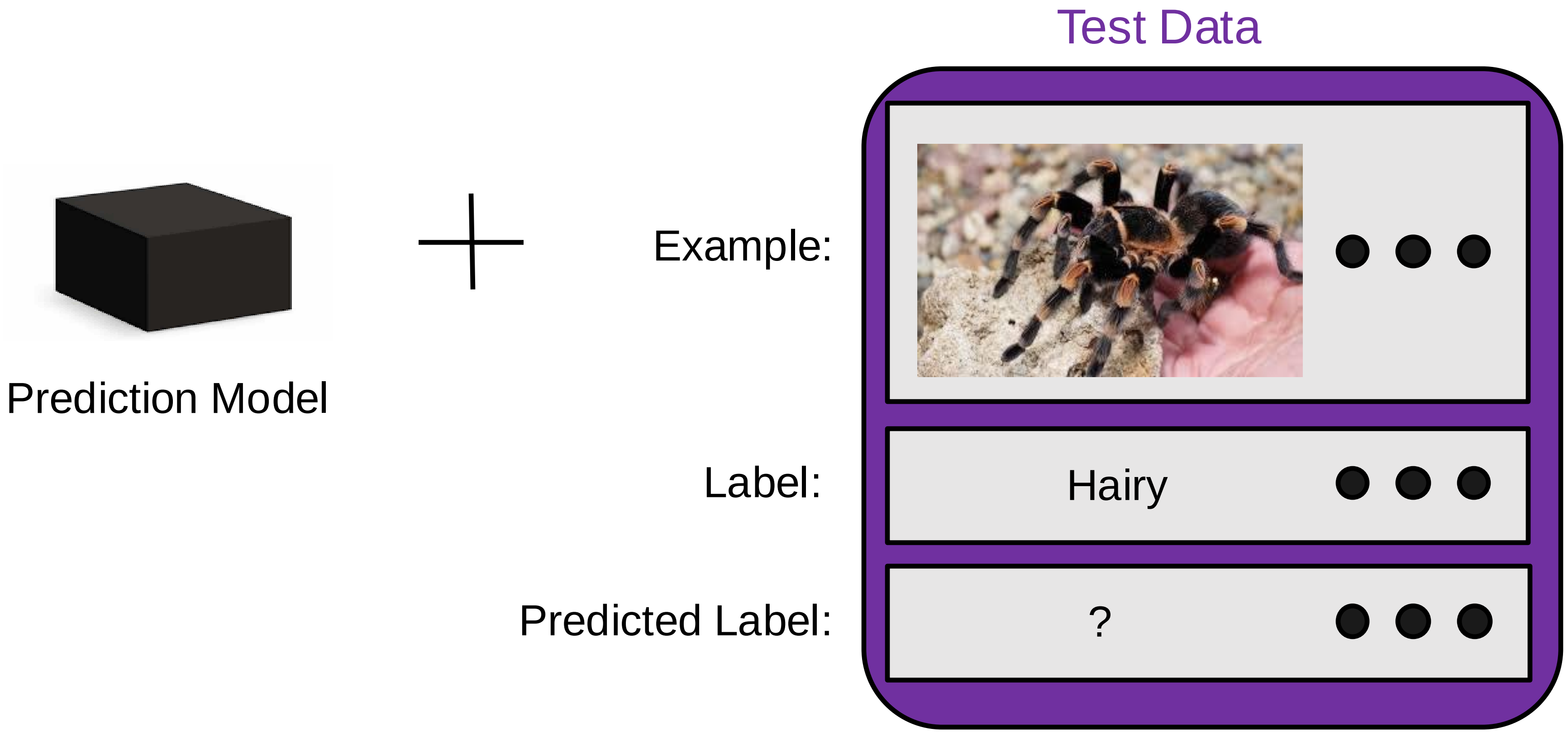
Goal: Design Models that **Generalize** Well to New, Previously Unseen Examples

2. Train model on "training set" to try to minimize prediction error on it



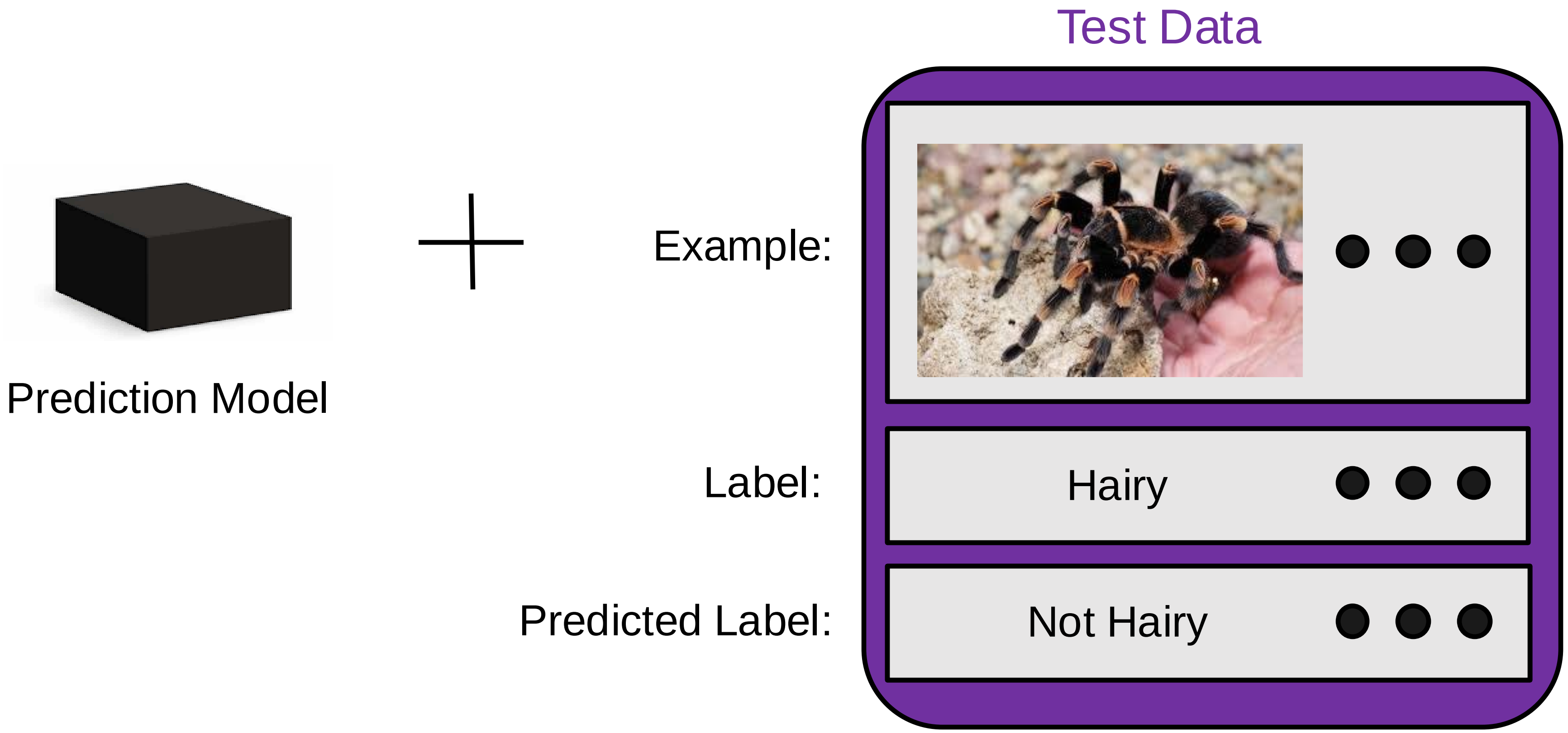
Goal: Design Models that **Generalize** Well to New, Previously Unseen Examples

3. Apply trained model on "test set" to measure generalization error



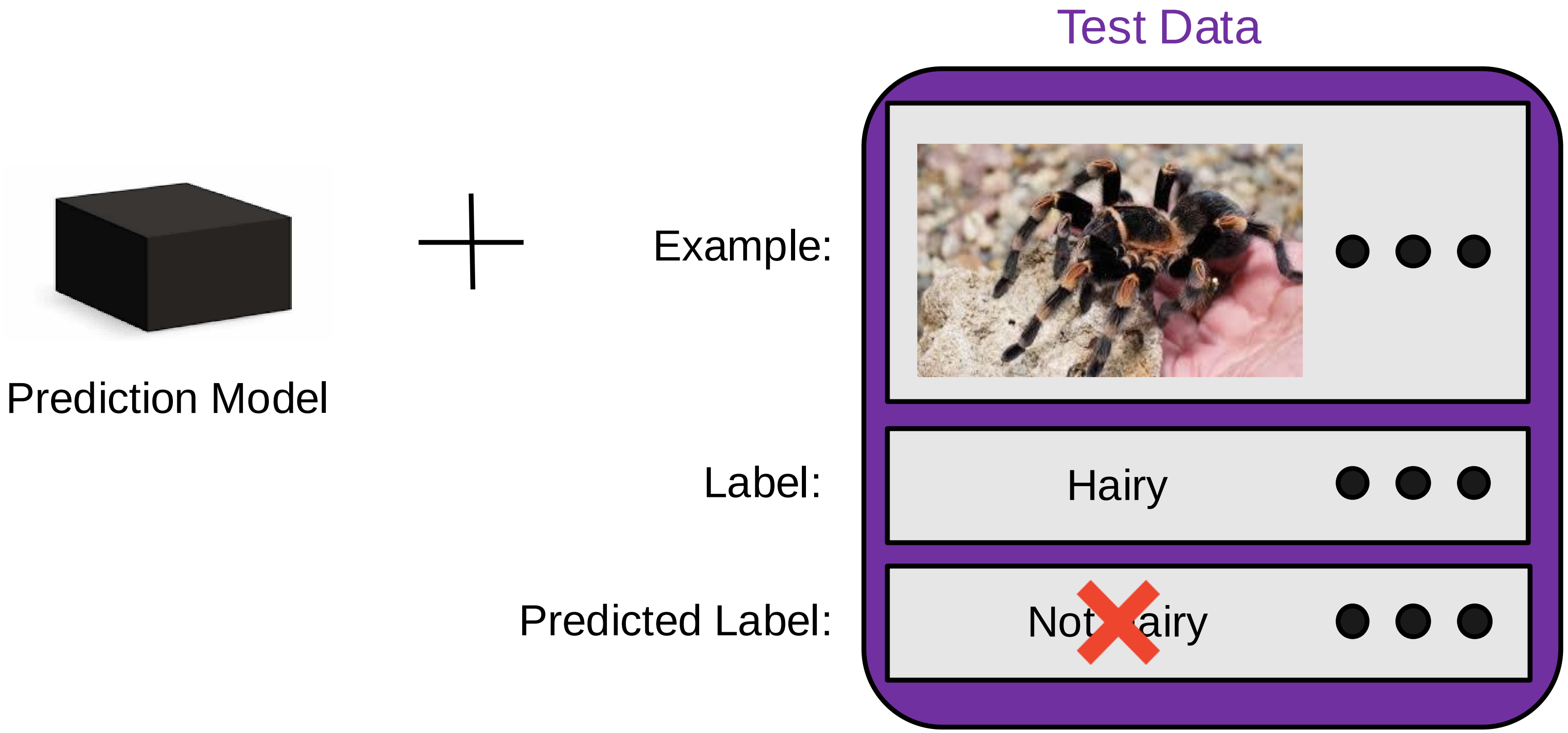
Goal: Design Models that **Generalize** Well to New, Previously Unseen Examples

3. Apply trained model on "test set" to measure generalization error



Goal: Design Models that **Generalize** Well to New, Previously Unseen Examples

3. Apply trained model on "test set" to measure generalization error



Supervised Learning Mimics One of the Ways that Humans Can Learn; e.g.,

Supervised Learning

(identify patterns from *structured* data with labels of target outputs)

Unsupervised Learning

(identify patterns in *unstructured* data)

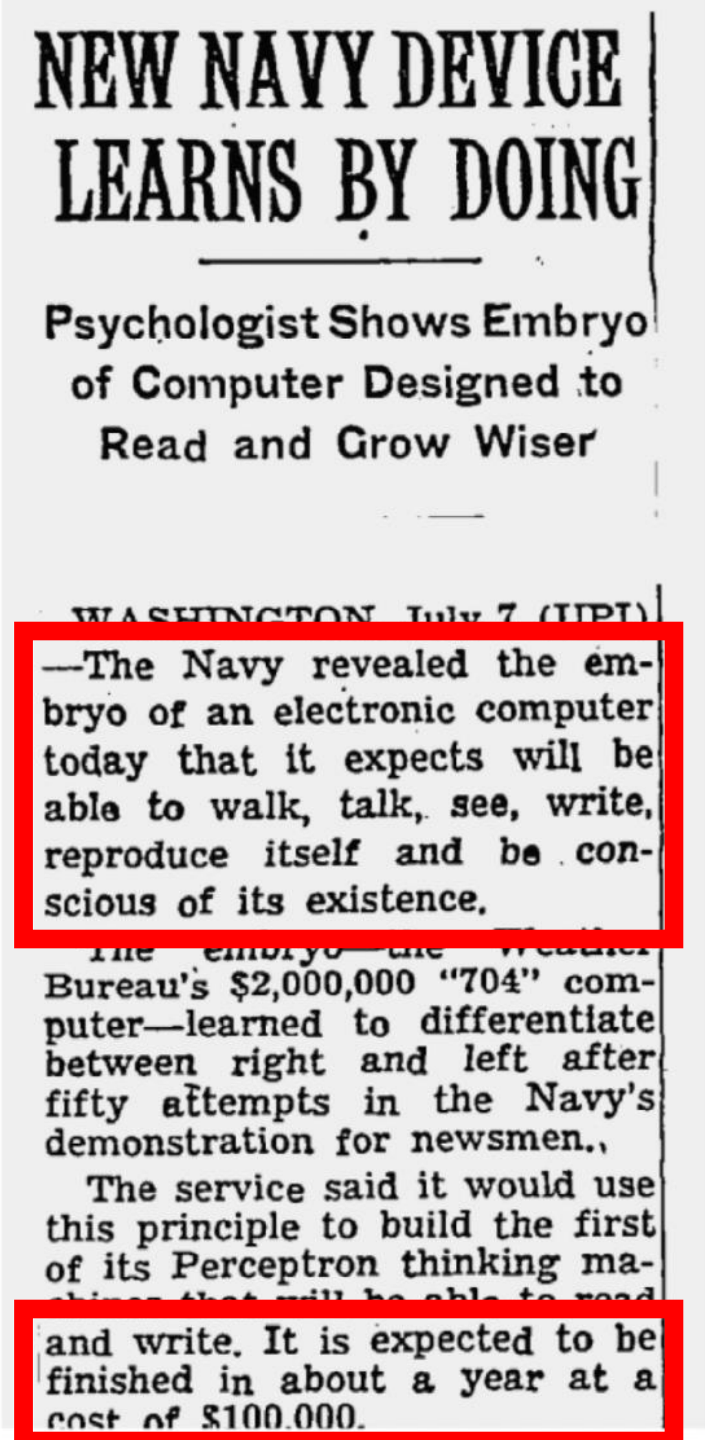
Today’s Topics

- Supervised learning: approach to develop a model
- Artificial neuron model: basic unit of neural networks
- Evaluating classification models

Vision

New York Times article, July 8, 1958 :

https://www.nytimes.com/1958/07/08/archives/new-navy-device-learns-by-doing-psychologist-shows-embryo-of.html



NEW NAVY DEVICE LEARNS BY DOING

Psychologist Shows Embryo of Computer Designed to Read and Grow Wiser

WASHINGTON, July 7 (UPI) —The Navy revealed the embryo of an electronic computer today that it expects will be able to walk, talk, see, write, reproduce itself and be conscious of its existence.

The embryo—the Weather Bureau's $2,000,000 "704" computer—learned to differentiate between right and left after fifty attempts in the Navy's demonstration for newsmen.

The service said it would use this principle to build the first of its Perceptron thinking machines that will be able to read and write. It is expected to be finished in about a year at a cost of $100,000.

Idea: Mimic Human Machinery

Neuron: basic computing machinery that enables human behavior!

- receives, processes, and transmits information, including:

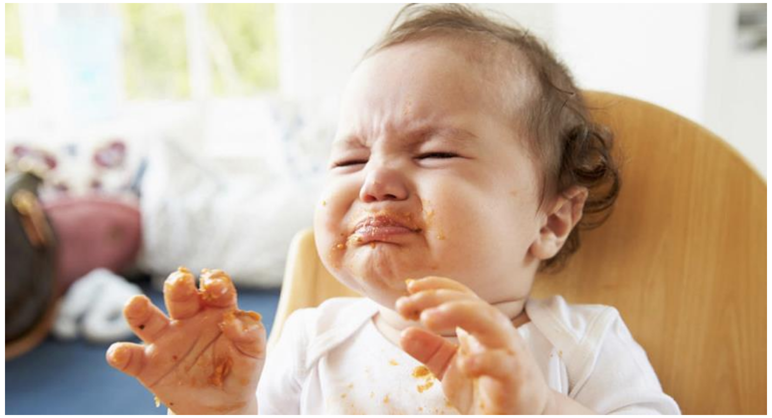
“hot”

https://www.clipart.email/clipart/dont-touch-hot-stove-clipart-73647.html

“loud”

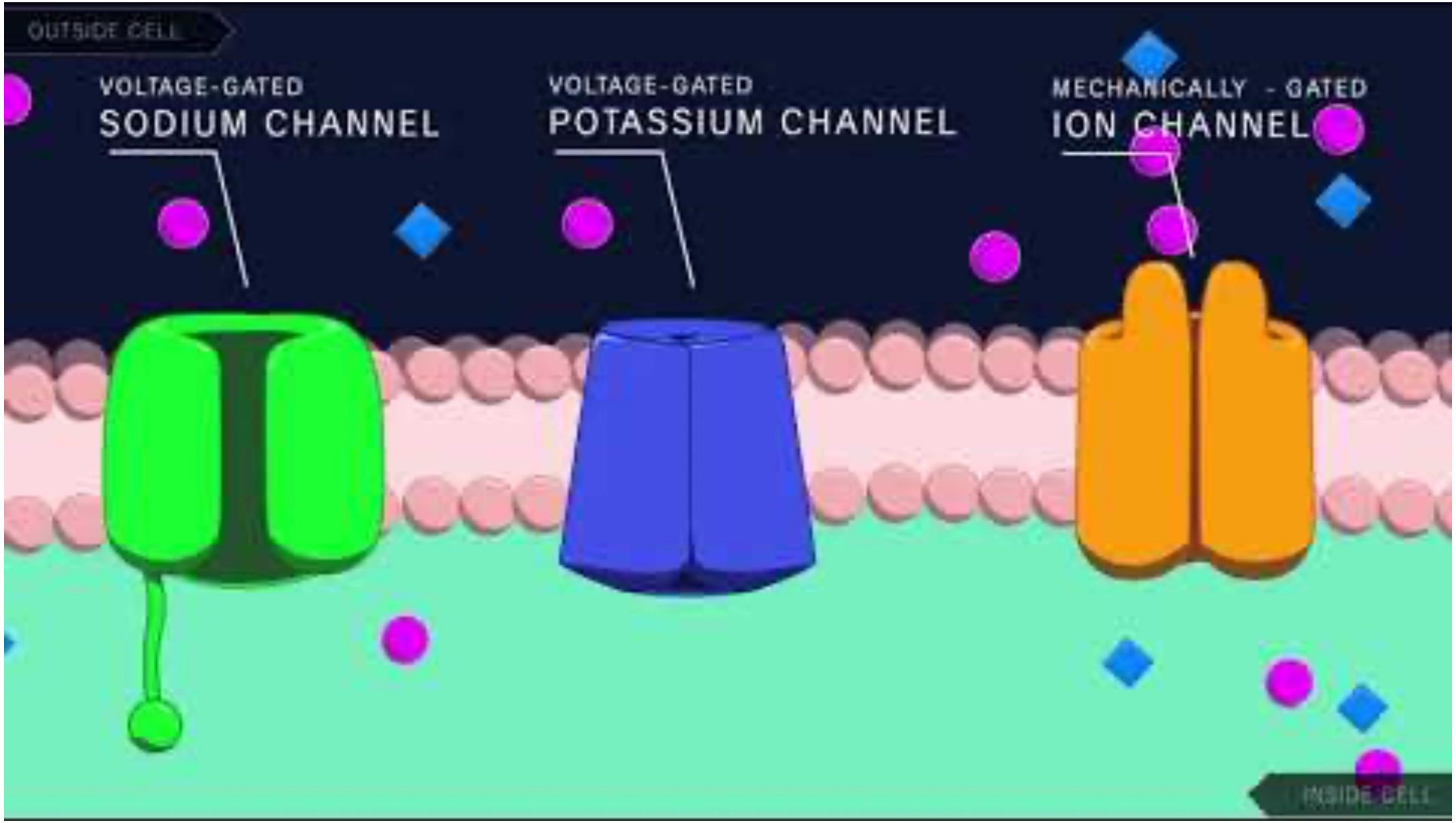
https://kisselpaso.com/if-the-sun-city-music-fest-gets-too-loud-there-is-a-phone-number-you-can-call-to-complain/

“spicy”



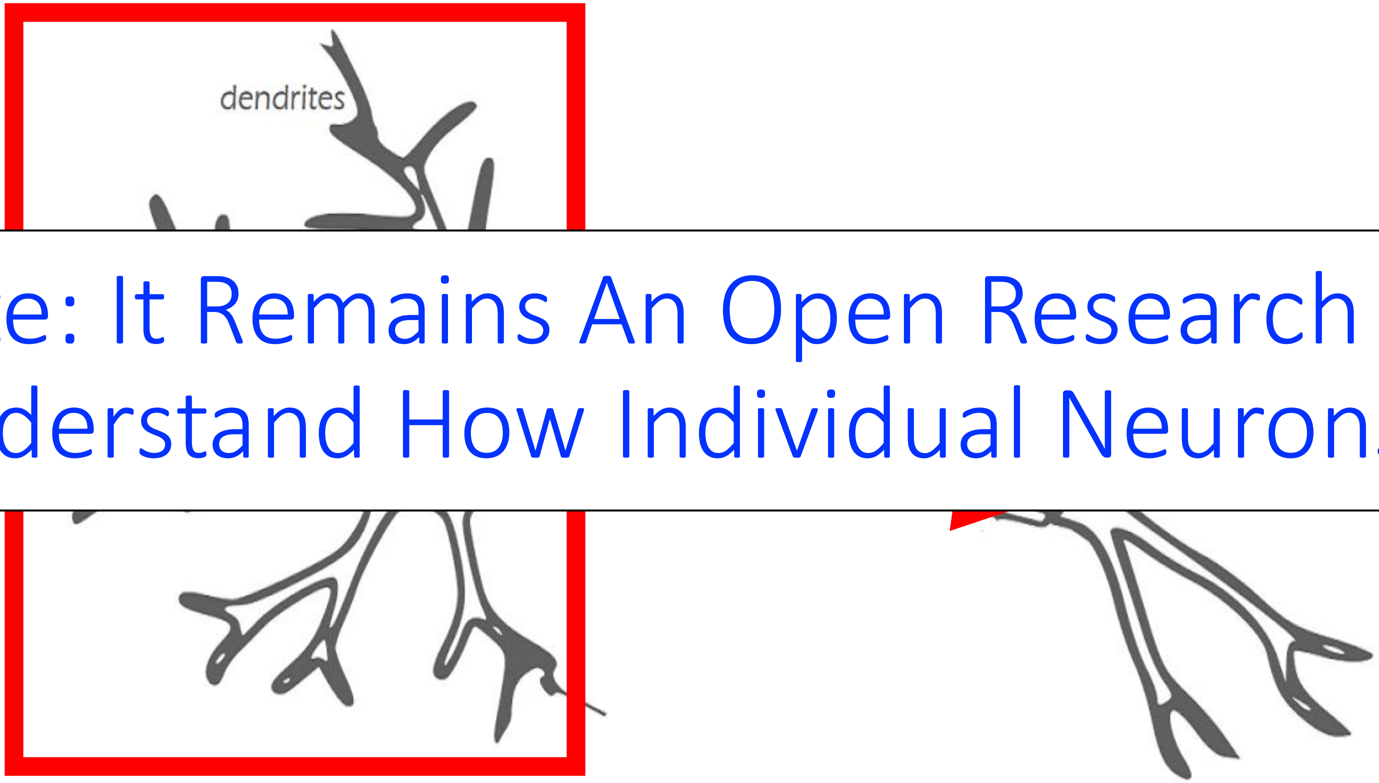
https://www.babycenter.com/404_when-can-my-baby-eat-spicy-foods_1368539.bc

Idea: Mimic Human Machinery



https://www.youtube.com/watch?v=oa6rvUJlg7o

Idea: Mimic Human Machinery



Sidenote: It Remains An Open Research Problem to Understand How Individual Neurons Work

- When the input signals exceed a certain threshold within a short period of time, a neuron "fires"
- Neuron "firing" is an "all-or-none" process, where either a signal is sent or nothing happens

https://becominghuman.ai/introduction-to-neural-networks-bd042ebf2653

Historical Context: Artificial Neurons

First mathematical model of neuron

1943 1945 1950 1956 1959

First programmable machine

Turing test

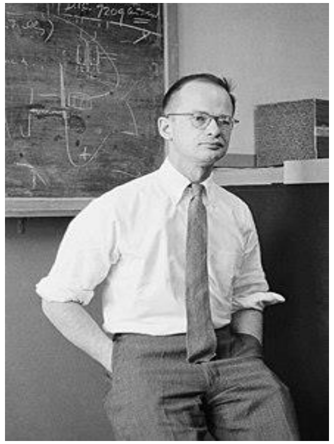
AI

Machine learning

Warren McCulloch (Neurophysiologist)

http://web.csulb.edu/~cwallis/artificialn/warren_mcculloch.html

Emerged from interdisciplinary collaboration



Walter Pitts (Mathematician)

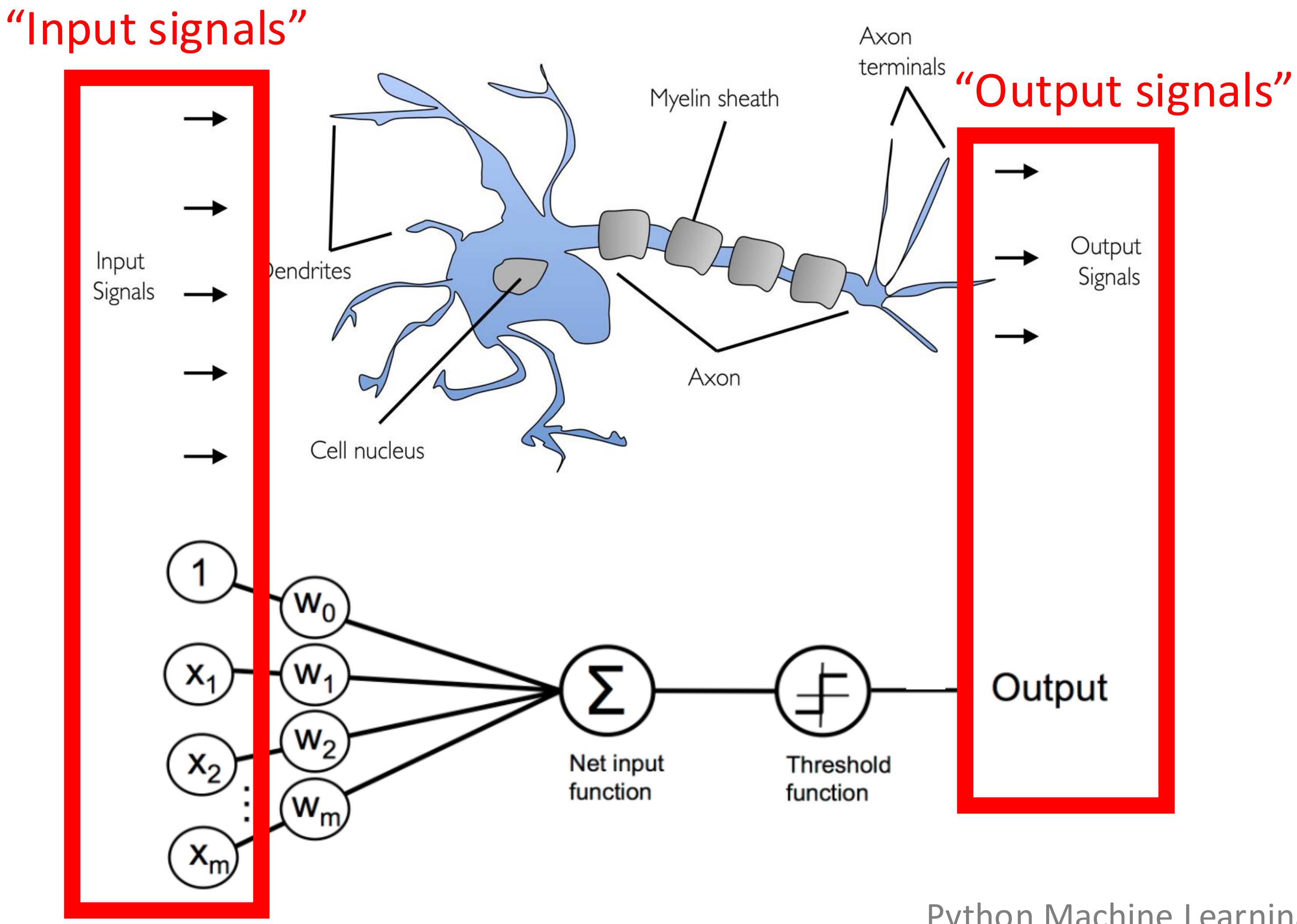
https://en.wikipedia.org/wiki/Walter_Pitts

Warren McCulloch and Walter Pitts, A Logical Calculus of Ideas Immanent in Nervous Activity, 1943

Artificial Neuron: McCulloch-Pitts Neuron

Biological Neuron:

Artificial Neurons
(e.g., Perceptron):

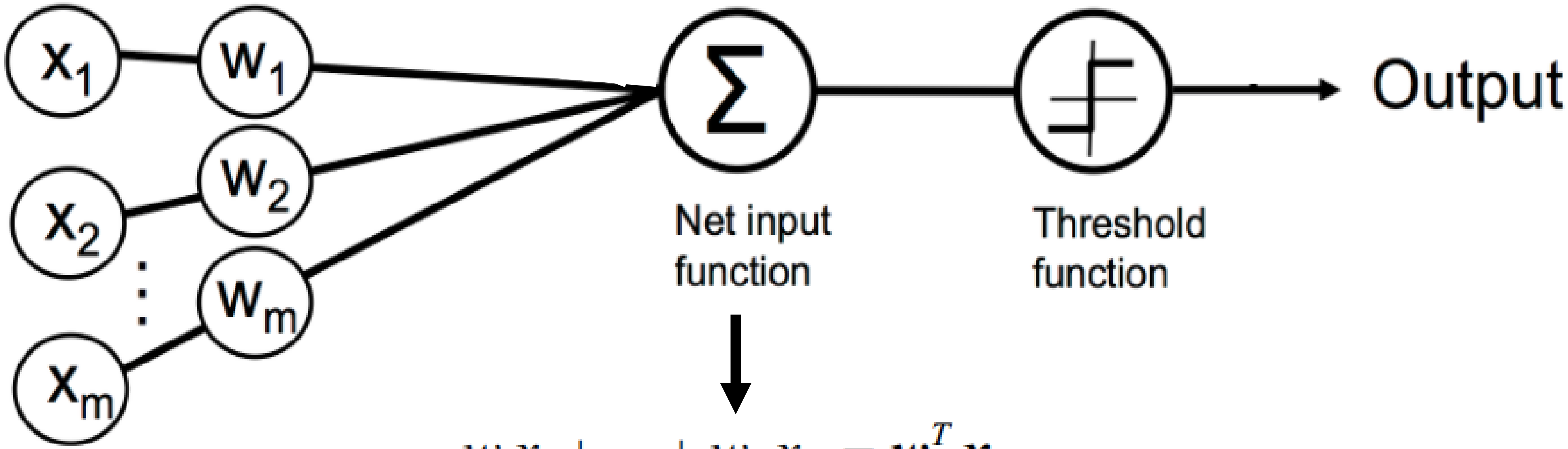


Python Machine Learning; Raschka & Mirjalili
https://becominghuman.ai/introduction-to-neural-networks-bd042ebf2653

Artificial Neuron: McCulloch-Pitts Neuron

- inputs (x) and weights (w) can be 0 or 1
- weights (w) and threshold values are fixed

- outputs 1 or 0 (mimics neurons by "firing" only when aggregate value exceeds threshold)

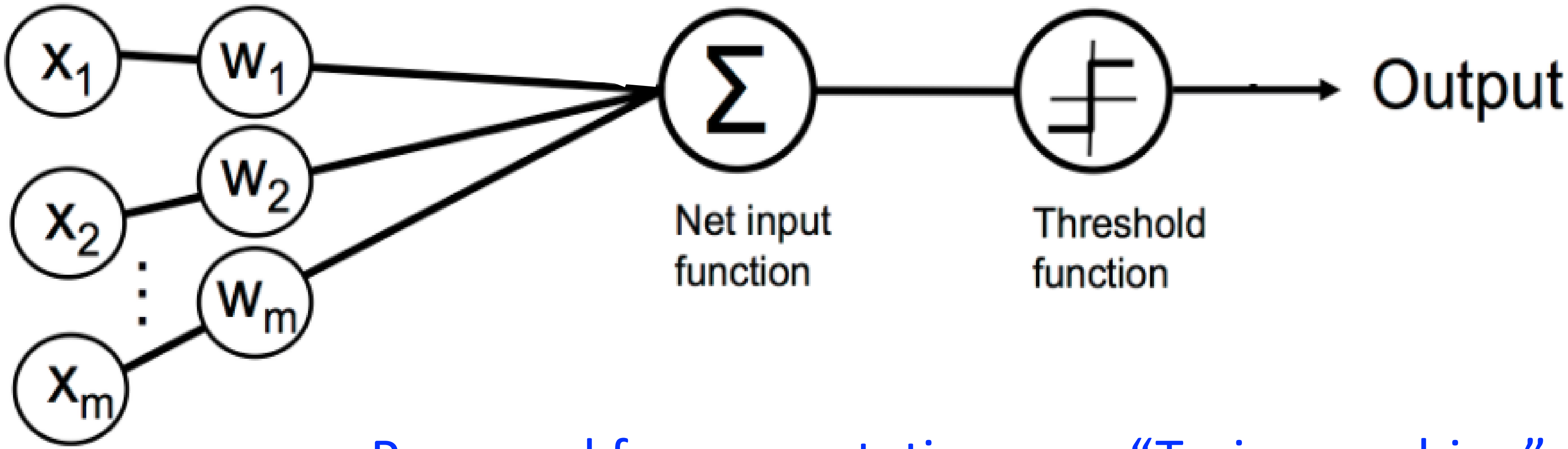


$$w_1x_1 + \ldots + w_mx_m = \boldsymbol{w}^T\boldsymbol{x}$$

Artificial Neuron: McCulloch-Pitts Neuron

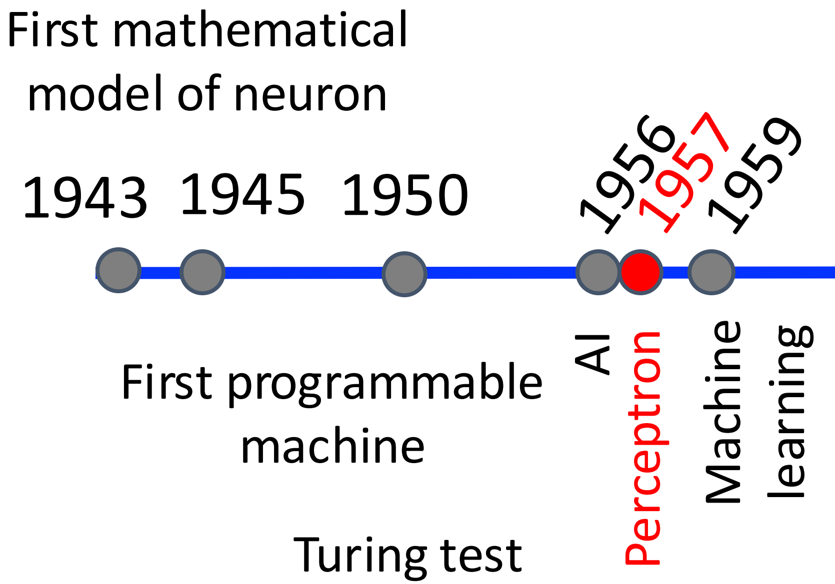
- inputs (x) and weights (w) can be 0 or 1
- weights (w) and threshold values are fixed

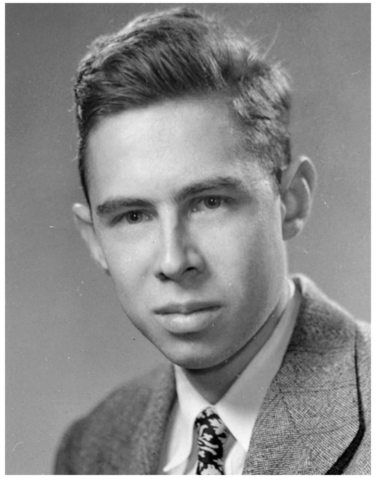
- outputs 1 or 0 (mimics neurons by “firing” only when aggregate value exceeds threshold)



Proposed for computation on a “Turing machine”

Perceptron: Model Mimicking Human Machinery



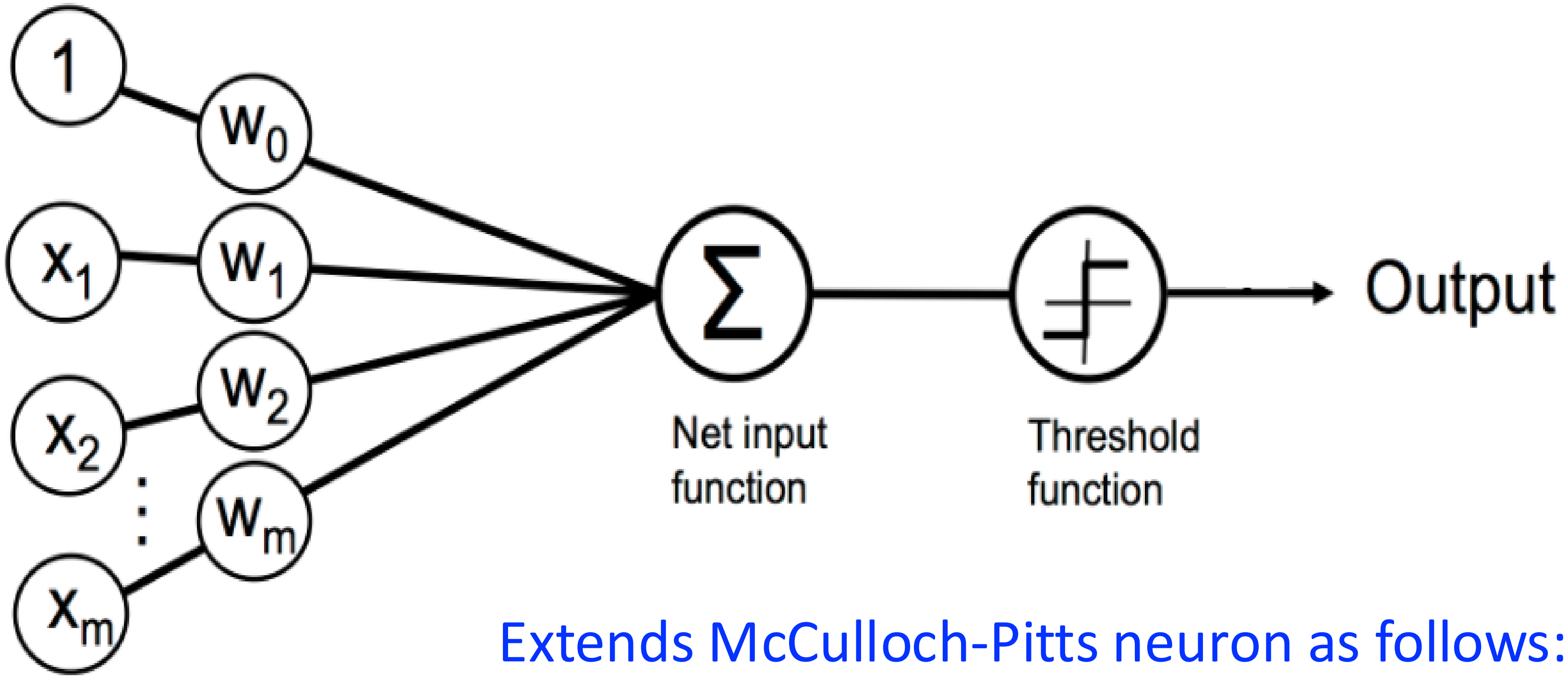


Frank Rosenblatt
(Psychologist)

https://news.cornell.edu/stories/2019/09/professors-perceptron-paved-way-ai-60-years-too-soon

Frank Rosenblatt, The perceptron, a perceiving and recognizing automaton Project Para. Cornell Aeronautical Laboratory, 1957

Perceptron: Architecture (Linear Threshold Unit)



Extends McCulloch-Pitts neuron as follows:

- inputs and weights can be any value
- weights (W) are learned

Perceptron: Architecture (Linear Threshold Unit)

- Function deciding output value ("fire" or not):

$$\phi(z) = \begin{cases} 1 & if\ \boxed{z \geq \theta} \\ -1 & otherwise \end{cases}$$

- Rewriting function:

$$\phi(z) = \begin{cases} 1 & if\ \boxed{z \geq 0} \\ -1 & otherwise \end{cases}$$

  - Where:

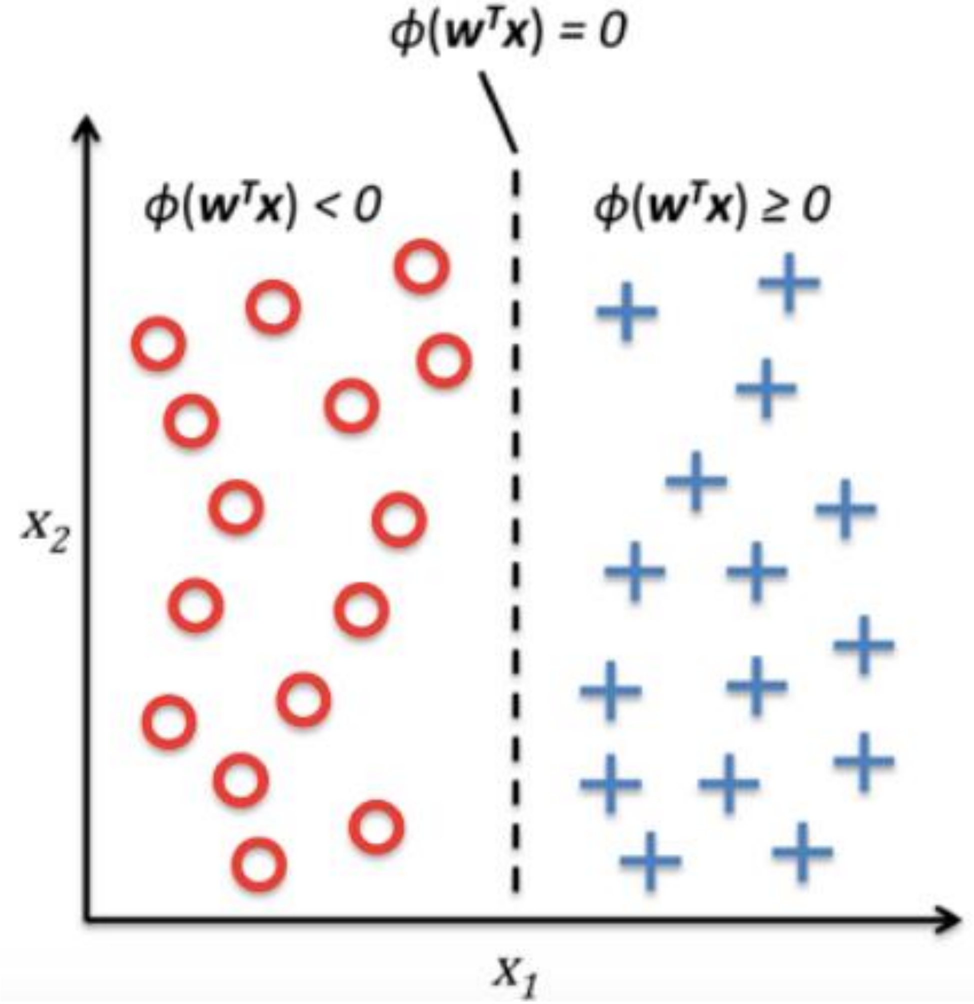
$$z = \boxed{w_0}\boxed{x_0} + w_1x_1 + \ldots + w_mx_m = \boldsymbol{w}^T\boldsymbol{x}$$

Bias $\boxed{-\theta}$ $\boxed{1}$

*** Note: Kamath textbook offers two common conventions for Perceptrons of using two possible output values of {-1, 1} and {0, 1}, in Chapters 2.5 and 4. The output choice dictates whether the threshold should be set to 0.5 or 0.**

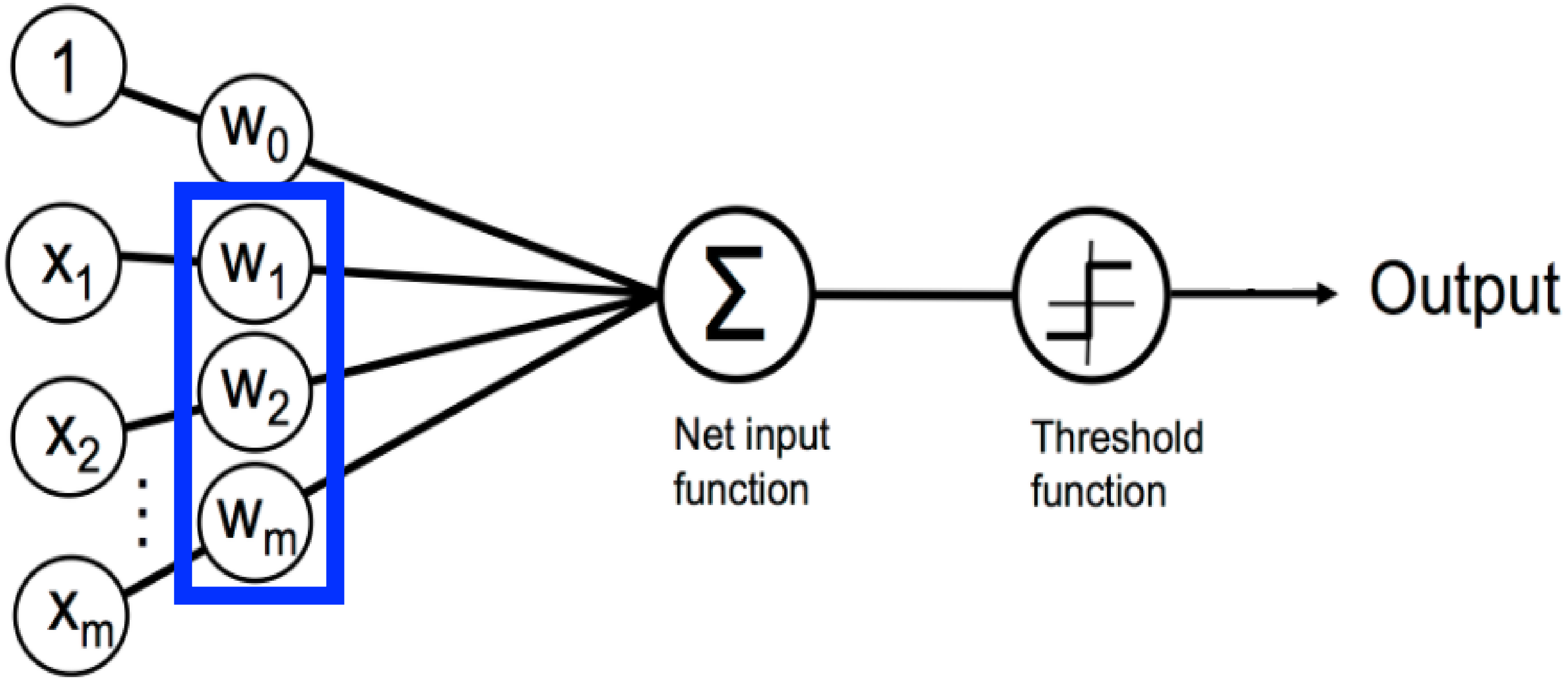
Perceptron: Architecture (Linear Threshold Unit)

Graphical representation when there are two features:



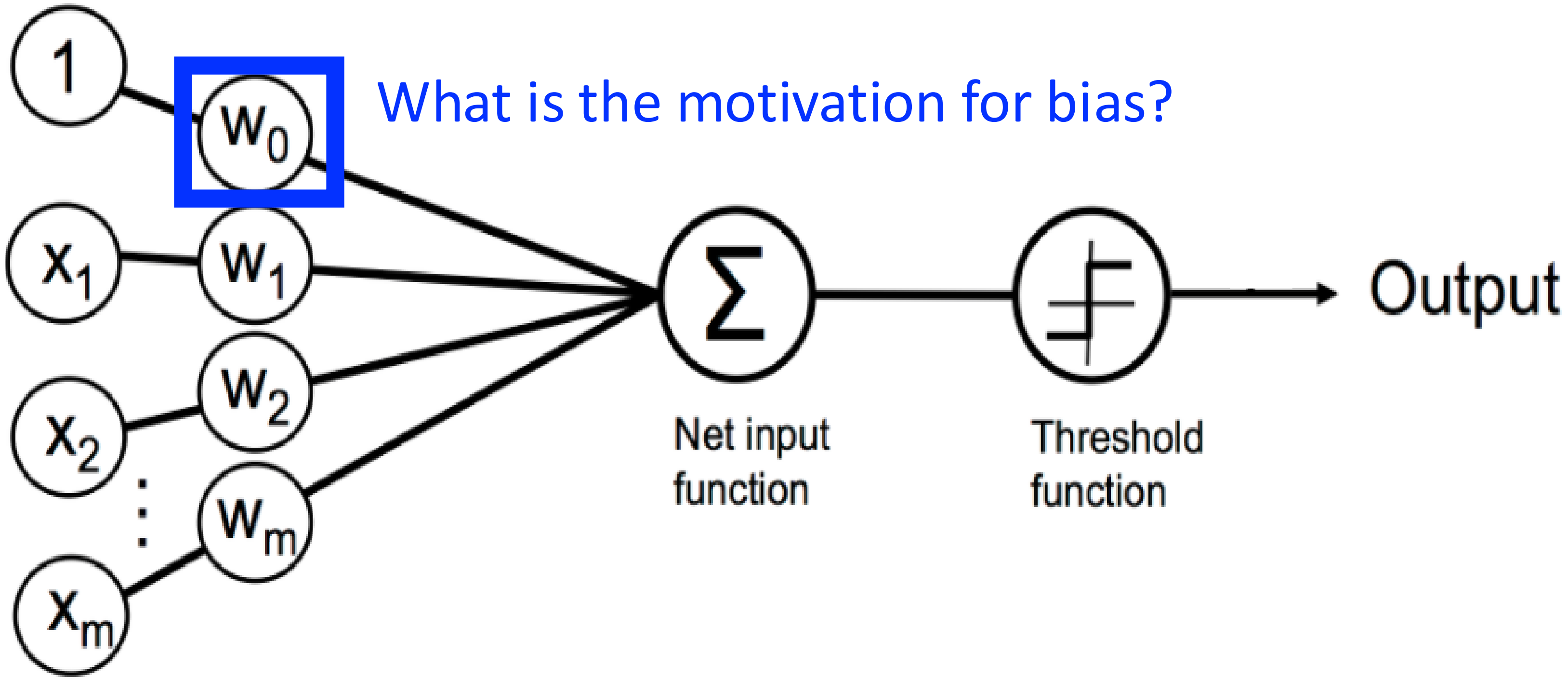
Python Machine Learning; Raschka & Mirjalili

Perceptron: Architecture (Linear Threshold Unit)



What is the motivation for weights?
e.g., predicting if you will like a movie?

Perceptron: Architecture (Linear Threshold Unit)



What is the motivation for bias?

Perceptron: Architecture (Linear Threshold Unit)

The model (in 2D, a line) must go through the origin without a **bias:**

$$z = \boxed{w_0}\boxed{x_0} + w_1 x_1 + \ldots + w_m x_m = \boldsymbol{w}^T \boldsymbol{x}$$

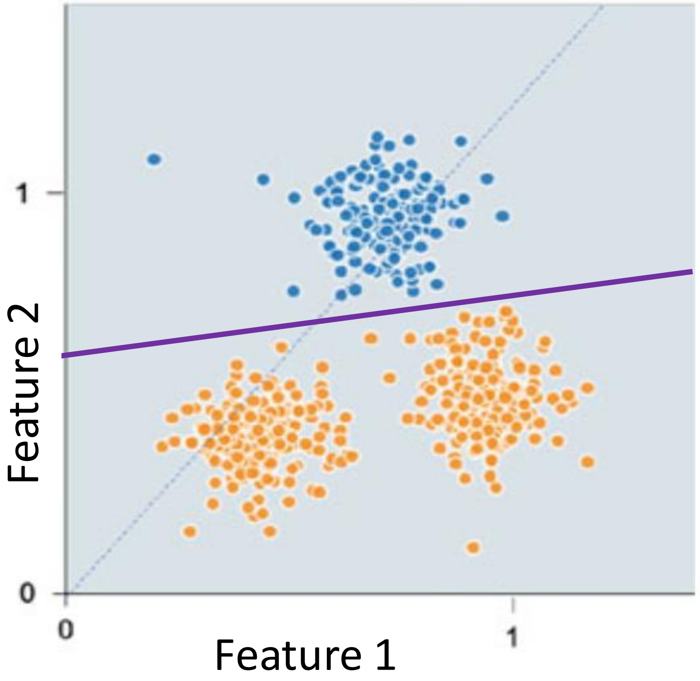
Bias $\boxed{-\theta}$ $\boxed{1}$

Binary classification problems (separate blue and orange points):

Kamath et al. Deep Learning for NLP and Speech Recognition. 2019.

Perceptron: Architecture (Linear Threshold Unit)

Binary classification problems (separate blue and orange points):



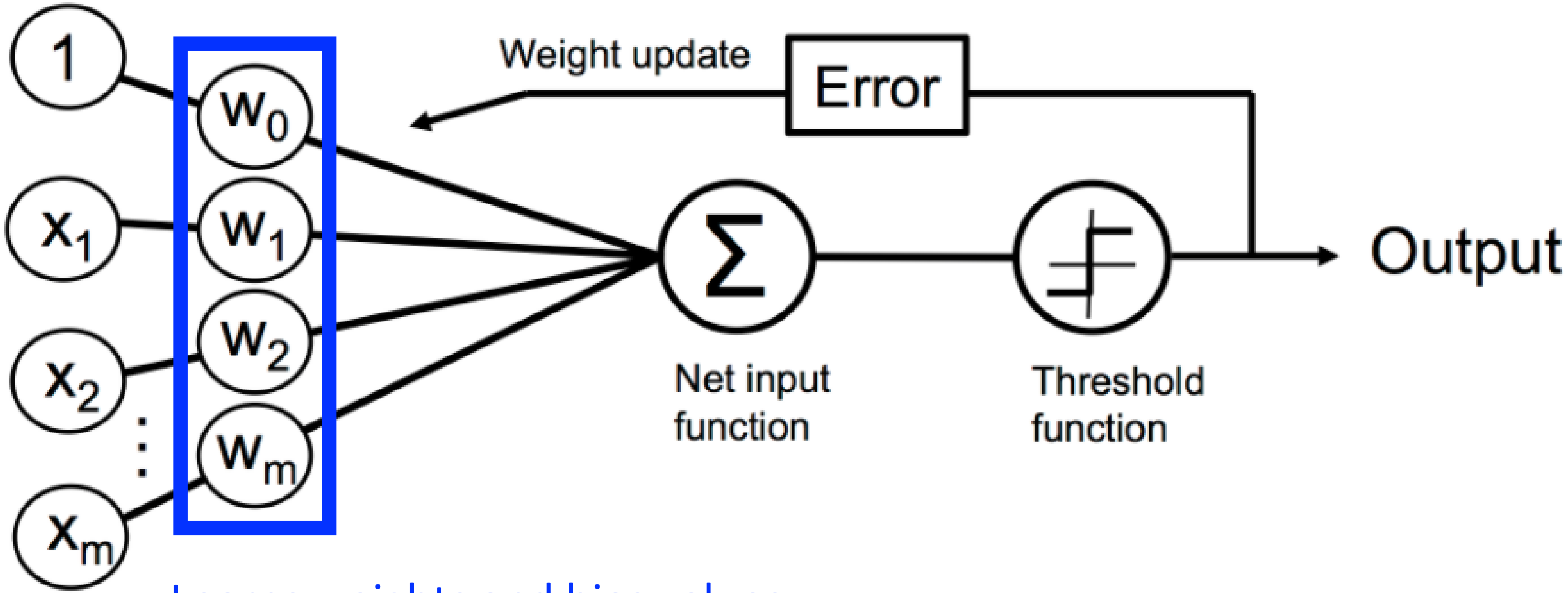
The model (in 2D, a line) doesn't have to pass through the origin with **bias:**

$$z = \boxed{w_0}\boxed{x_0} + w_1 x_1 + \ldots + w_m x_m = \boldsymbol{w}^T \boldsymbol{x}$$

Bias $\boxed{-\theta}$ $\boxed{1}$

Kamath et al. Deep Learning for NLP and Speech Recognition. 2019.

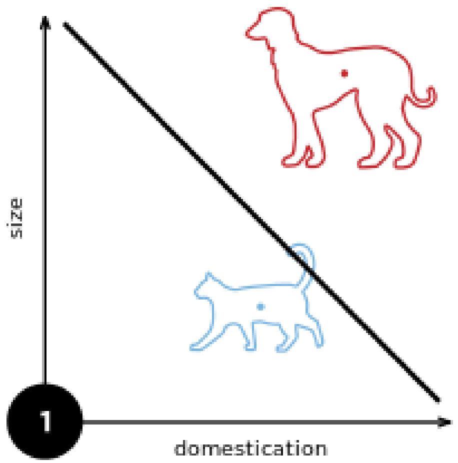
Perceptron: Learning Algorithm



Learns weights and bias values

Python Machine Learning; Raschka & Mirjalili

Perceptron: Learning Algorithm

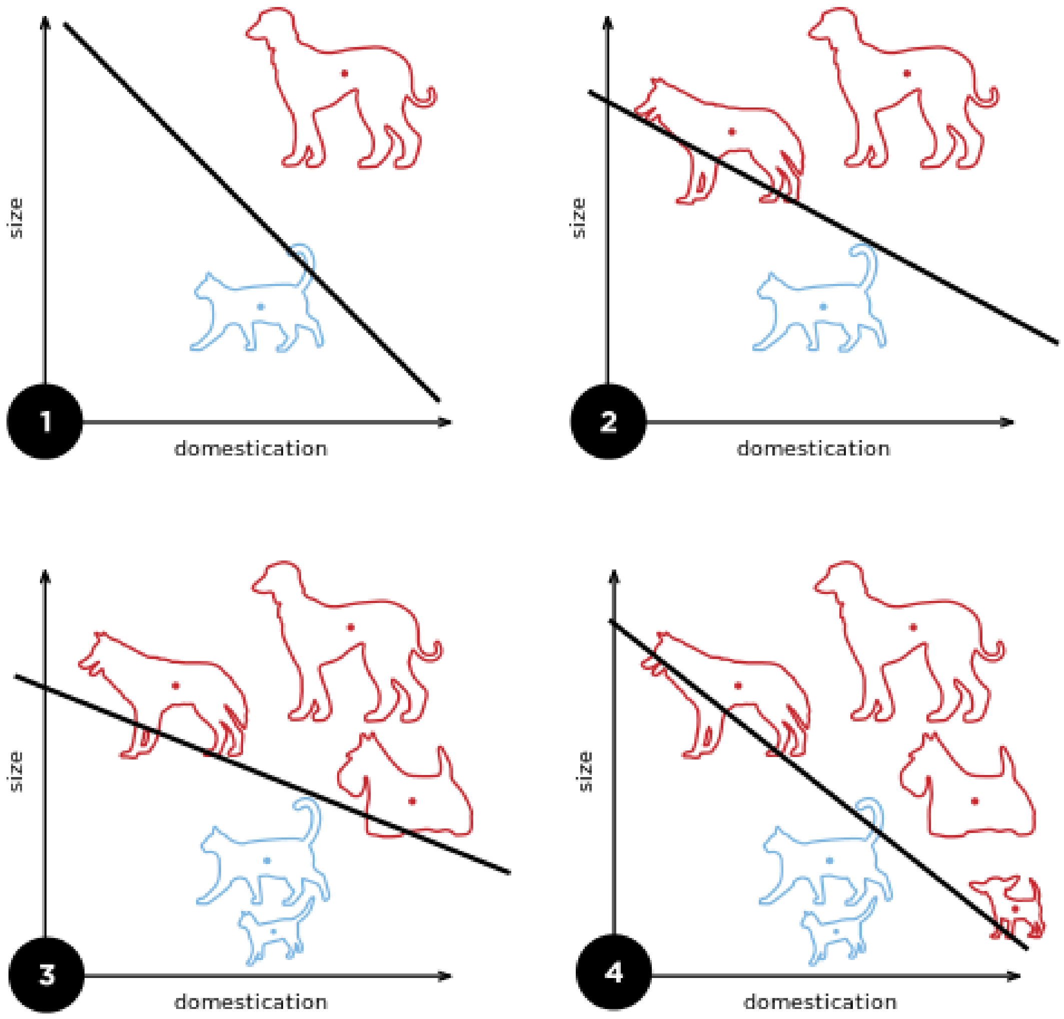


Process: iteratively update boundary with observation of each additional example:

https://en.wikipedia.org/wiki/Perceptron

Perceptron: Learning Algorithm

Process: iteratively update boundary with observation of each additional example:



https://en.wikipedia.org/wiki/Perceptron

Perceptron: Learning Algorithm

1. Initialize weights/bias to 0 or small random numbers
2. Repeat until stopping criterion met:
   1. Compute predicted value (i.e., {-1, 1}): $\sum_{j=0}^{m} \boldsymbol{x}_j \boldsymbol{w}_j = \boldsymbol{w}^T \boldsymbol{x}$
   2. Update parameters based on prediction success: $w_j := w_j + \boxed{\Delta w_j}$

$$\boxed{\Delta w_j = \eta \left(\text{target}^{(i)} - \text{output}^{(i)}\right) x_j^{(i)}}$$

(for *i-th* training example)

Learning Rate

(set a priori and held constant)

True Class Label

Predicted Class Label

Perceptron: Learning Algorithm

1. Initialize weights/bias to 0 or small random numbers
2. Repeat until stopping criterion met:
    1. Compute predicted value (i.e., {-1, 1}): $\sum_{j=0}^{m} \boldsymbol{x}_j \boldsymbol{w}_j = \boldsymbol{w}^T \boldsymbol{x}$
    2. Update parameters based on prediction success: $w_j := w_j + \boxed{\Delta w_j}$

$$\Delta w_j = \eta \boxed{(\text{target}^{(i)} - \text{output}^{(i)})} x_j^{(i)}$$

(for *i-th* training example)

What happens to the parameters when
the model predicts the **correct** class label?

- no update since result is 0

https://sebastianraschka.com/faq/docs/diff-perceptron-adaline-neuralnet.html

Perceptron: Learning Algorithm

1. Initialize weights/bias to 0 or small random numbers
2. Repeat until stopping criterion met:
    1. Compute predicted value (i.e., {-1, 1}): $\sum_{j=0}^{m} \boldsymbol{x}_j \boldsymbol{w}_j = \boldsymbol{w}^T \boldsymbol{x}$
    2. Update parameters based on prediction success: $w_j := w_j + \boxed{\Delta w_j}$

$$\Delta w_j = \eta \boxed{(\text{target}^{(i)} - \text{output}^{(i)})} x_j^{(i)}$$

(for *i-th* training example)

What happens to the parameters when
the model predicts the **wrong** class label?

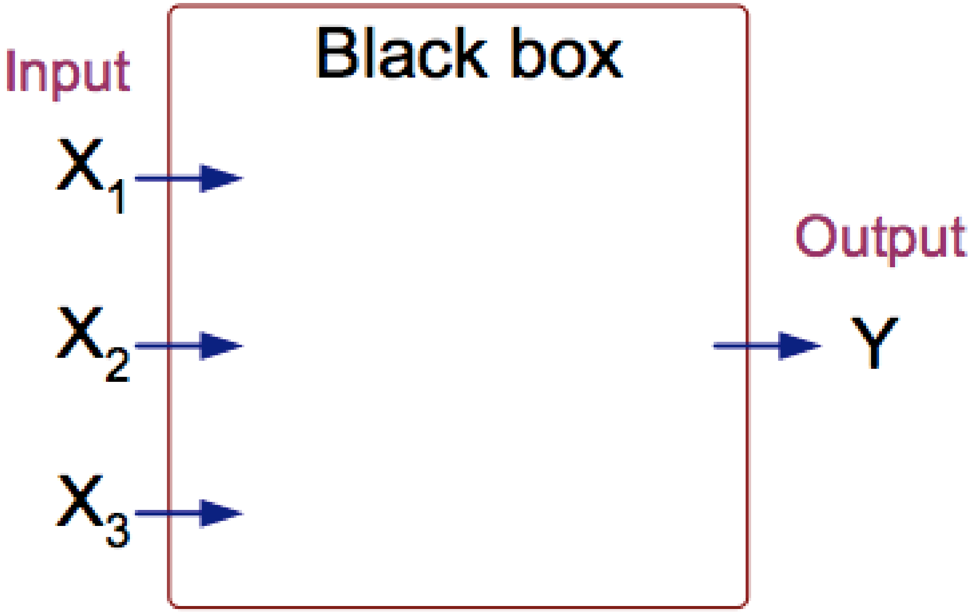
- updates since result is "2" or "-2"

https://sebastianraschka.com/faq/docs/diff-perceptron-adaline-neuralnet.html

Perceptron: Example

- True Model: Y is 1 if at least 2 of the 3 inputs are 1, and -1 otherwise

X_1	X_2	X_3	Y
1	0	0	
1	0	1	
1	1	0	
1	1	1	?
0	0	1	
0	1	0	
0	1	1	
0	0	0	

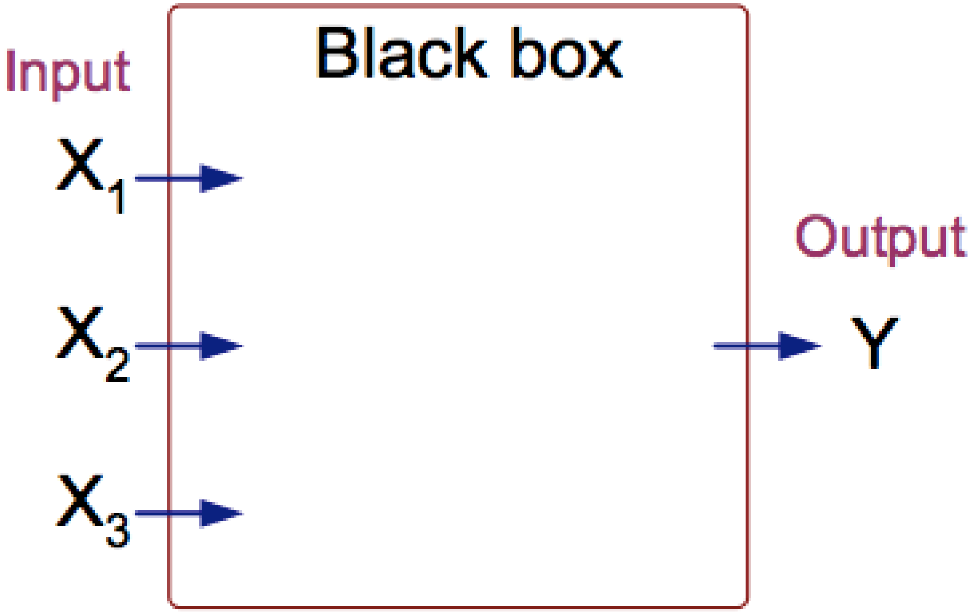


https://www-users.cs.umn.edu/~kumar001/dmbook/slides/chap4_ann.pdf

Perceptron: Example

- True Model: Y is 1 if at least 2 of the 3 inputs are 1, and -1 otherwise

<table>
<tr><th>X_1</th><th>X_2</th><th>X_3</th><th>Y</th></tr>
<tr><td>1</td><td>0</td><td>0</td><td>-1</td></tr>
<tr><td>1</td><td>0</td><td>1</td><td rowspan="7">?</td></tr>
<tr><td>1</td><td>1</td><td>0</td></tr>
<tr><td>1</td><td>1</td><td>1</td></tr>
<tr><td>0</td><td>0</td><td>1</td></tr>
<tr><td>0</td><td>1</td><td>0</td></tr>
<tr><td>0</td><td>1</td><td>1</td></tr>
<tr><td>0</td><td>0</td><td>0</td></tr>
</table>

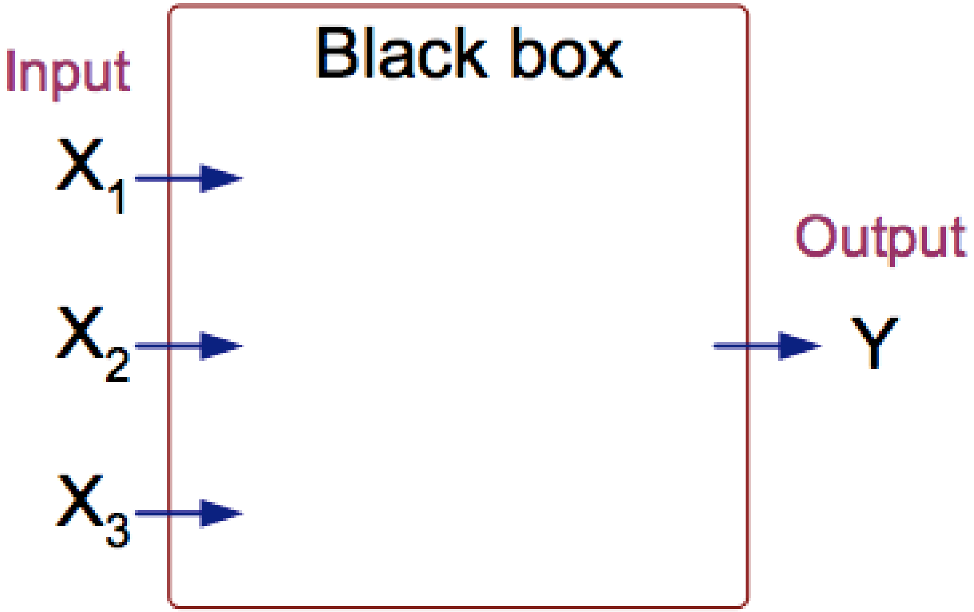


https://www-users.cs.umn.edu/~kumar001/dmbook/slides/chap4_ann.pdf

Perceptron: Example

- True Model: Y is 1 if at least 2 of the 3 inputs are 1, and -1 otherwise

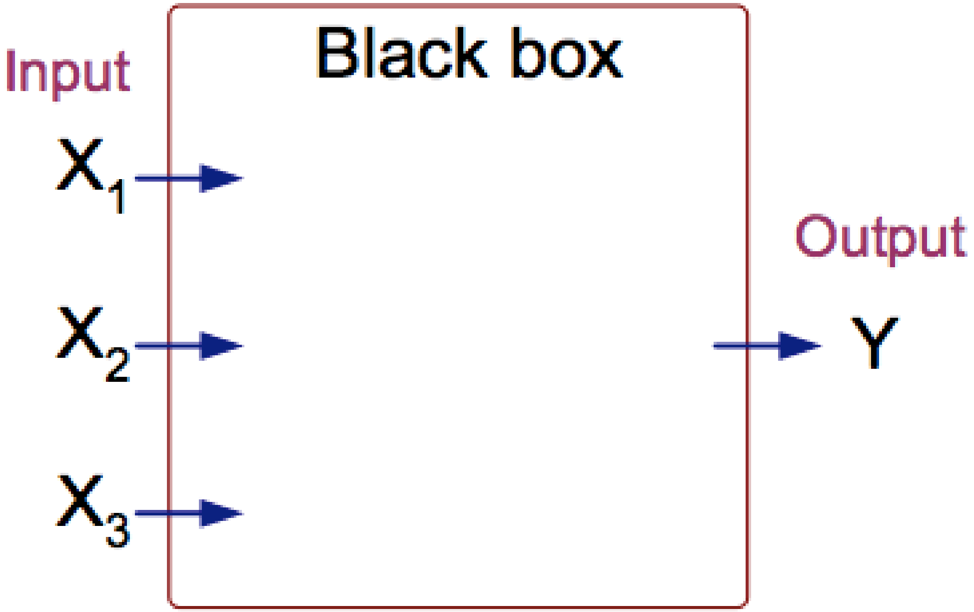
X_1	X_2	X_3	Y
1	0	0	-1
1	0	1	1
1	1	0	
1	1	1	
0	0	1	?
0	1	0	
0	1	1	
0	0	0	



Perceptron: Example

- True Model: Y is 1 if at least 2 of the 3 inputs are 1, and -1 otherwise

X_1	X_2	X_3	Y
1	0	0	-1
1	0	1	1
1	1	0	1
1	1	1	1
0	0	1	-1
0	1	0	-1
0	1	1	1
0	0	0	-1



https://www-users.cs.umn.edu/~kumar001/dmbook/slides/chap4_ann.pdf

Perceptron: Example (Training with 1rst Sample)

- Compute predicted value: $\sum_{j=0}^{m} x_j w_j = w^T x$; $\phi(w^T x)$ = $\begin{cases} 1 \text{ if } \phi(w^T x) \geq 0 \\ -1 \text{ otherwise} \end{cases}$

X_1	X_2	X_3	Y
1	0	0	-1

Predicted
?

w_0	w_1	w_2	w_3
0	0	0	0

https://www-users.cs.umn.edu/~kumar001/dmbook/slides/chap4_ann.pdf

Perceptron: Example (Training with 1rst Sample)

- Update params: $w_j = w_j + \eta\,(\text{target}^{(i)} - \text{output}^{(i)})\,x_j^{(i)}$; learning rate = 0.1

X_1	X_2	X_3	Y
1	0	0	-1

Predicted
1

w_0	w_1	w_2	w_3
0	0	0	0
?	?	?	?

$$\Delta w_0 = \eta\,(\text{target}^{(i)} - \text{output}^{(i)})$$

$$\Delta w_1 = \eta\,(\text{target}^{(i)} - \text{output}^{(i)})\,x_1^{(i)}$$

$$\Delta w_2 = \eta\,(\text{target}^{(i)} - \text{output}^{(i)})\,x_2^{(i)}$$

$$\Delta w_3 = \eta\,(\text{target}^{(i)} - \text{output}^{(i)})\,x_3^{(i)}$$

https://www-users.cs.umn.edu/~kumar001/dmbook/slides/chap4_ann.pdf

Perceptron: Example (Training with 1rst Sample)

- Update params: $w_j = w_j + \eta\,(\text{target}^{(i)} - \text{output}^{(i)})\,x_j^{(i)}$; learning rate = 0.1

X_1	X_2	X_3	Y
1	0	0	-1

Predicted
1

w_0	w_1	w_2	w_3
0	0	0	0
?	?	?	?

Δw_0 = 0.1(-1-1)*1 = -0.2

Δw_1 = 0.1(-1-1)*1 = -0.2

Δw_2 = 0.1(-1-1)*0 = 0

Δw_3 = 0.1(-1-1)*0 = 0

When predicted value is greater than true value (e.g., 1 > -1), the updates *decrease* parameter values to increase the likelihood of classifying the sample as -1 next time

https://www-users.cs.umn.edu/~kumar001/dmbook/slides/chap4_ann.pdf

Perceptron: Example (Training with 2nd Sample)

- Compute output value: $\sum_{j=0}^{m} x_j w_j = w^T x$; $\phi(w^T x) = \begin{cases} 1 \text{ if } \phi(w^T x) \geq 0 \\ -1 \text{ otherwise} \end{cases}$

X_1	X_2	X_3	Y
1	0	0	-1
1	0	1	1

Predicted
1
?

w_0	w_1	w_2	w_3
0	0	0	0
-0.2	-0.2	0	0

Perceptron: Example (Training with 2nd Sample)

- Update params: $w_j = w_j + \eta\ (\text{target}^{(i)} - \text{output}^{(i)})\ x_j^{(i)}$; learning rate = 0.1

X_1	X_2	X_3	Y
1	0	0	-1
1	0	1	1

Predicted
1
-1

w_0	w_1	w_2	w_3
0	0	0	0
-0.2	-0.2	0	0
?	?	?	?

$$\Delta w_0 = \eta\ (\text{target}^{(i)} - \text{output}^{(i)})$$

$$\Delta w_1 = \eta\ (\text{target}^{(i)} - \text{output}^{(i)})\ x_1^{(i)}$$

$$\Delta w_2 = \eta\ (\text{target}^{(i)} - \text{output}^{(i)})\ x_2^{(i)}$$

$$\Delta w_3 = \eta\ (\text{target}^{(i)} - \text{output}^{(i)})\ x_3^{(i)}$$

https://www-users.cs.umn.edu/~kumar001/dmbook/slides/chap4_ann.pdf

Perceptron: Example (Training with 2nd Sample)

- Update params: $w_j = w_j + \eta\ (\text{target}^{(i)} - \text{output}^{(i)})\ x_j^{(i)}$; learning rate = 0.1

X_1	X_2	X_3	Y
1	0	0	-1
1	0	1	1

Predicted
1
-1

w_0	w_1	w_2	w_3
0	0	0	0
-0.2	-0.2	0	0
?	?	?	?

Δw_0 = 0.1(1--1)*1 = 0.2

Δw_1 = 0.1(1--1)*1 = 0.2

Δw_2 = 0.1(1--1)*0 = 0

Δw_3 = 0.1(1--1)*1 = 0.2

When predicted value is less than true value (e.g., -1 < 1), the updates *increase* parameter values to increase the likelihood of classifying the sample as 1 next time

https://www-users.cs.umn.edu/~kumar001/dmbook/slides/chap4_ann.pdf

Perceptron: Example (Training with 2nd Sample)

- Update params: $w_j = w_j + \eta\,(\text{target}^{(i)} - \text{output}^{(i)})\,x_j^{(i)}$; learning rate = 0.1

X_1	X_2	X_3	Y
1	0	0	-1
1	0	1	1

Predicted
1
-1

w_0	w_1	w_2	w_3
0	0	0	0
-0.2	-0.2	0	0
0	0	0	0.2

Δw_0 = 0.1(1--1)*1 = 0.2

Δw_1 = 0.1(1--1)*1 = 0.2

Δw_2 = 0.1(1--1)*0 = 0

Δw_3 = 0.1(1--1)*1 = 0.2

What is the influence of the learning rate? i.e., what would happen if the value was larger/smaller?

https://www-users.cs.umn.edu/~kumar001/dmbook/slides/chap4_ann.pdf

Perceptron: Example – One Epoch (Training with All Samples)

- $w_j = w_j + \eta\,(\text{target}^{(i)} - \text{output}^{(i)})\,x_j^{(i)}$; learning rate = 0.1

X_1	X_2	X_3	Y
1	0	0	-1
1	0	1	1
1	1	0	1
1	1	1	1
0	0	1	-1
0	1	0	-1
0	1	1	1
0	0	0	-1

	w_0	w_1	w_2	w_3
0	0	0	0	0
1	-0.2	-0.2	0	0
2	0	0	0	0.2
3	0	0	0	0.2
4	0	0	0	0.2
5	-0.2	0	0	0
6	-0.2	0	0	0
7	0	0	0.2	0.2
8	-0.2	0	0.2	0.2

https://www-users.cs.umn.edu/~kumar001/dmbook/slides/chap4_ann.pdf

Perceptron: Example – Six Epochs

- $w_j = w_j + \eta\,(\text{target}^{(i)} - \text{output}^{(i)})\,x_j^{(i)}$; learning rate = 0.1

X_1	X_2	X_3	Y
1	0	0	-1
1	0	1	1
1	1	0	1
1	1	1	1
0	0	1	-1
0	1	0	-1
0	1	1	1
0	0	0	-1

Epoch	w_0	w_1	w_2	w_3
0	0	0	0	0

Perceptron: Example – Six Epochs

- $w_j = w_j + \eta\,(\text{target}^{(i)} - \text{output}^{(i)})\,x_j^{(i)}$; learning rate = 0.1

X_1	X_2	X_3	Y
1	0	0	-1
1	0	1	1
1	1	0	1
1	1	1	1
0	0	1	-1
0	1	0	-1
0	1	1	1
0	0	0	-1

	w_0	w_1	w_2	w_3
0	0	0	0	0
1	-0.2	-0.2	0	0
2	0	0	0	0.2
3	0	0	0	0.2
4	0	0	0	0.2
5	-0.2	0	0	0
6	-0.2	0	0	0
7	0	0	0.2	0.2
8	-0.2	0	0.2	0.2

Epoch	w_0	w_1	w_2	w_3
0	0	0	0	0
1	-0.2	0	0.2	0.2
2	-0.2	0	0.4	0.2
3	-0.4	0	0.4	0.2
4	-0.4	0.2	0.4	0.4
5	-0.6	0.2	0.4	0.2
6	-0.6	0.4	0.4	0.2

https://www-users.cs.umn.edu/~kumar001/dmbook/slides/chap4_ann.pdf

Perceptron: Learning Algorithm Choices

1. $\boxed{\text{Initialize weights and bias}}$ **1. Values**
2. Repeat until $\boxed{\text{stopping criterion}}$ met: **2.**
    1. Compute predicted value (i.e., {-1, 1}): $\sum_{j=0}^{m} \boldsymbol{x}_j \boldsymbol{w}_j = \boldsymbol{w}^T \boldsymbol{x}$
    2. Update parameters based on prediction success: $w_j := w_j + \boxed{\Delta w_j}$

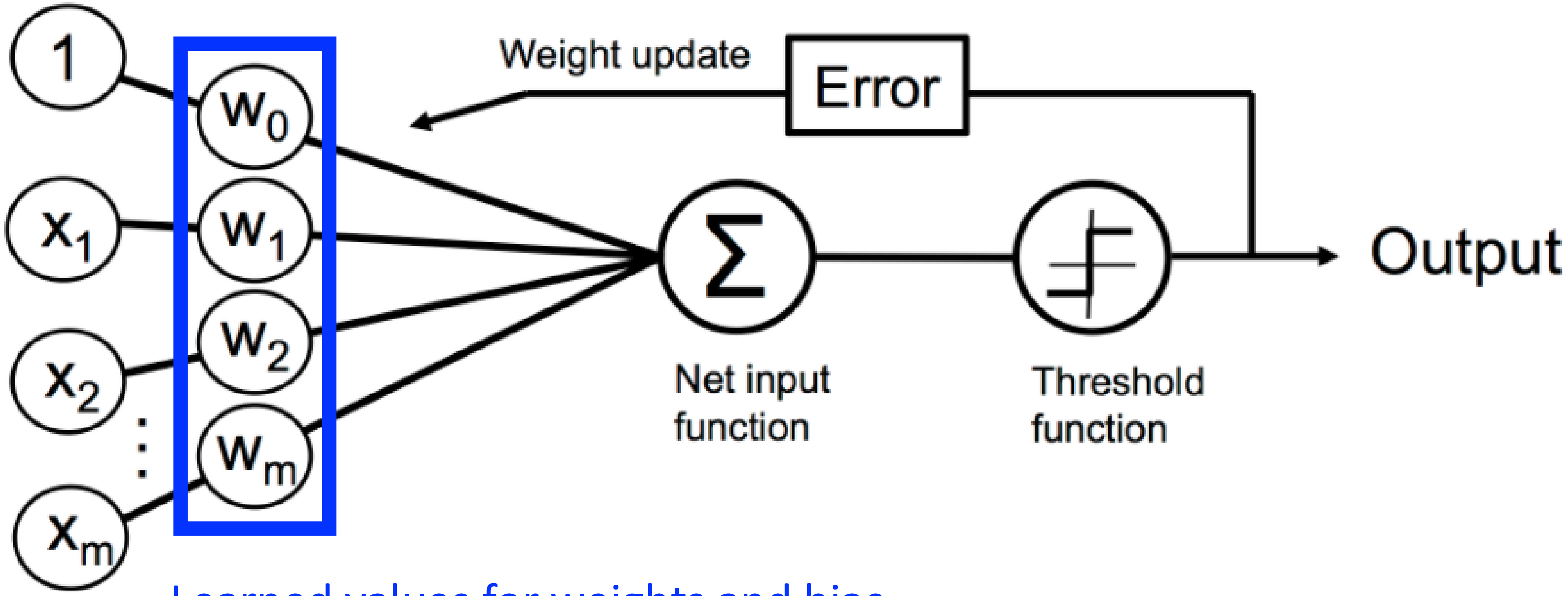
$$\Delta w_j = \boxed{\eta} \left(\text{target}^{(i)} - \text{output}^{(i)}\right) x_j^{(i)}$$

3. Learning Rate

Today’s Topics

- Supervised learning: approach to develop a model
- Artificial neuron model: basic unit of neural networks
- Evaluating classification models

Evaluation: How Good Is Our Model?



Learned values for weights and bias

Evaluation Methods: Confusion Matrix

		Actual	
		Hairy	Not hairy
Predicted	Hairy	TP	FP
	Not hairy	FN	TN

TP = true positive
TN = true negative
FP = false positive
FN = false negative

Evaluation Methods: Descriptive Statistics

e.g.,

		Actual: Hairy	Actual: Not hairy
Predicted	Hairy	50	10
Predicted	Not hairy	15	100

Commonly-used statistical descriptions:

- How many ***actual hairy*** results are there? - 65
- How many ***actual not hairy*** results are there? - 110
- How many ***correctly classified instances***? - 150/175 ~ 86%
- How many ***incorrectly classified instances***? - 25/175 ~ 14%
- What is the ***precision (aka – sensitivity)?*** **(proportion of predicted instances actually hairy)** $\frac{TP}{TP + FP}$
  - 50/(50+10) ~ 83%

- What is the ***recall***? **(proportion of hairy instances predicted as hairy)** $\frac{TP}{TP + FN}$
  - 50/(50+15) ~ 77%

Evaluation Methods: Descriptive Statistics

(a)

		Actual: Z	Actual: Not Z
Predicted	Z	100	20
	Not Z	20	100

(b)

		Actual: Z	Actual: Not Z
Predicted	Z	80	0
	Not Z	120	40

(c)

		Actual: Z	Actual: Not Z
Predicted	Z	100	100
	Not Z	0	40

Which confusion matrix reflects the best-performing model?

Evaluation Methods: Descriptive Statistics

(a)

Predicted \ Actual	Z	Not Z
Z	100	20
Not Z	20	100

(b)

A ***cautious*** model that is correct whenever it predicts a positive label

Predicted \ Actual	Z	Not Z
Z	80	0
Not Z	120	40

However, it can incorrectly predict many positives have negative labels

(c)

Predicted \ Actual	Z	Not Z
Z	100	100
Not Z	0	40

Which confusion matrix reflects the model with the highest precision? $\frac{TP}{TP+FP}$

Evaluation Methods: Descriptive Statistics

(a)

		Actual: Z	Actual: Not Z
Predicted	Z	100	20
	Not Z	20	100

(b)

		Actual: Z	Actual: Not Z
Predicted	Z	80	0
	Not Z	120	40

(c)

However, many negatives can be deemed positive

		Actual: Z	Actual: Not Z
Predicted	Z	100	100
	Not Z	$\boxed{0}$	40

An ***ambitious*** model that tries to identify every positive label

Which confusion matrix reflects the model with the highest recall? $\frac{TP}{TP + FN}$

Class Discussion

- Which of these evaluation metrics would you use versus not use and why?
  - Accuracy (percentage of correctly classified examples)
  - Precision (percentage of relevant instances among retrieved instances)
  - Recall (percentage of relevant instances retrieved)

- Scenario 1: Medical test for a rare disease affecting one in every million people.

- Scenario 2: Deciding which emails to flag as spam.

Today’s Topics

- Supervised learning: approach to develop a model
- Artificial neuron model: basic unit of neural networks
- Evaluating classification models

The End