

L-2

Supervised Learning I

MAIB7002
Fundamentals of AI Models and Technologies

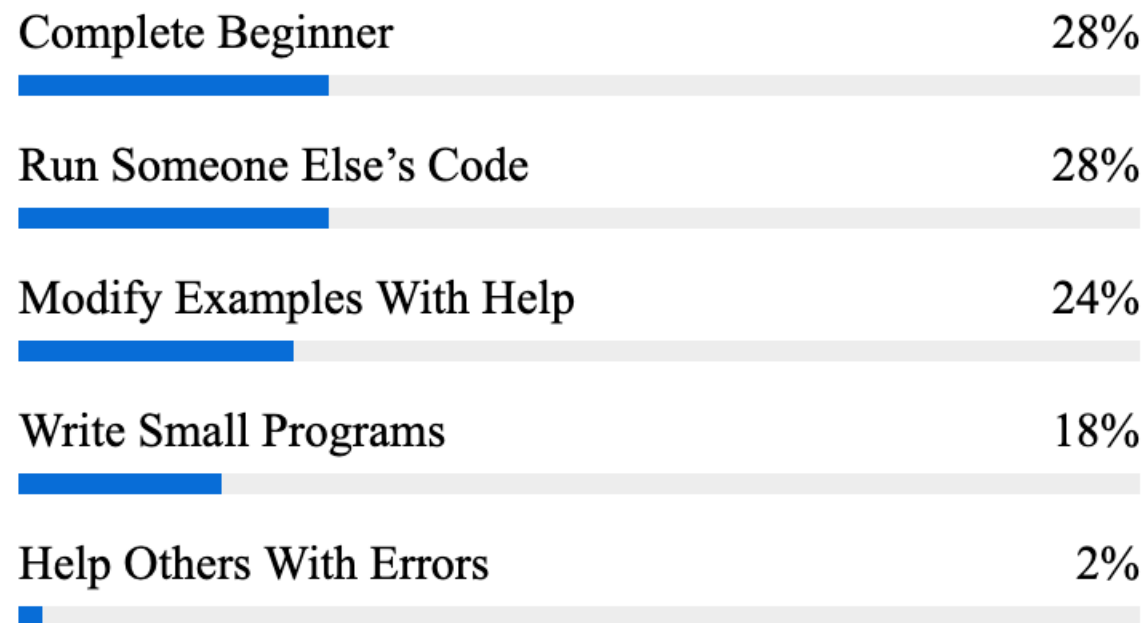


Sit with Your Project Group Members

Python and Math

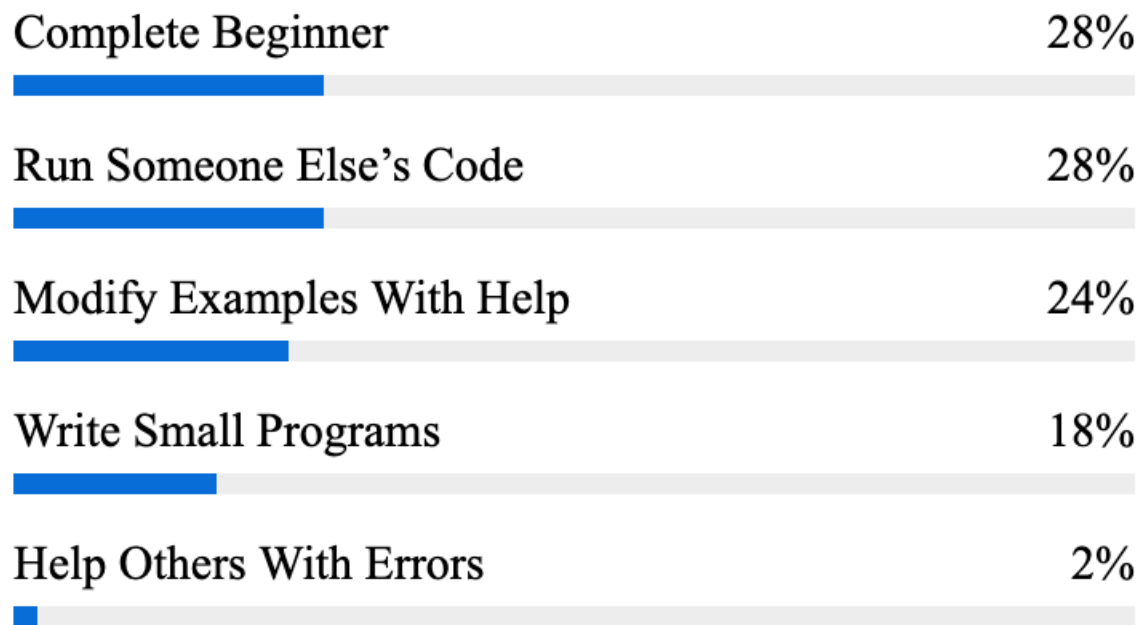
Python and Math

Python

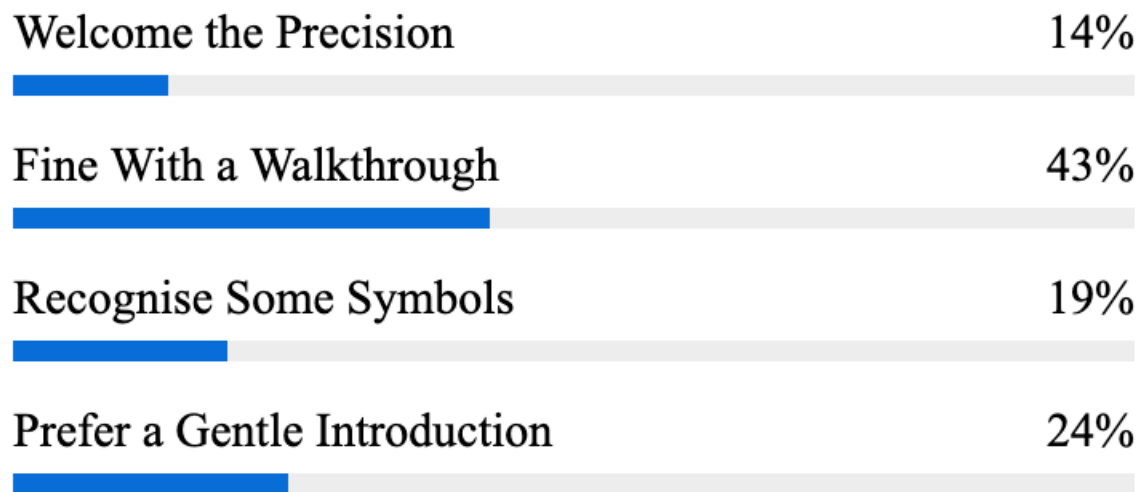


Python and Math

Python



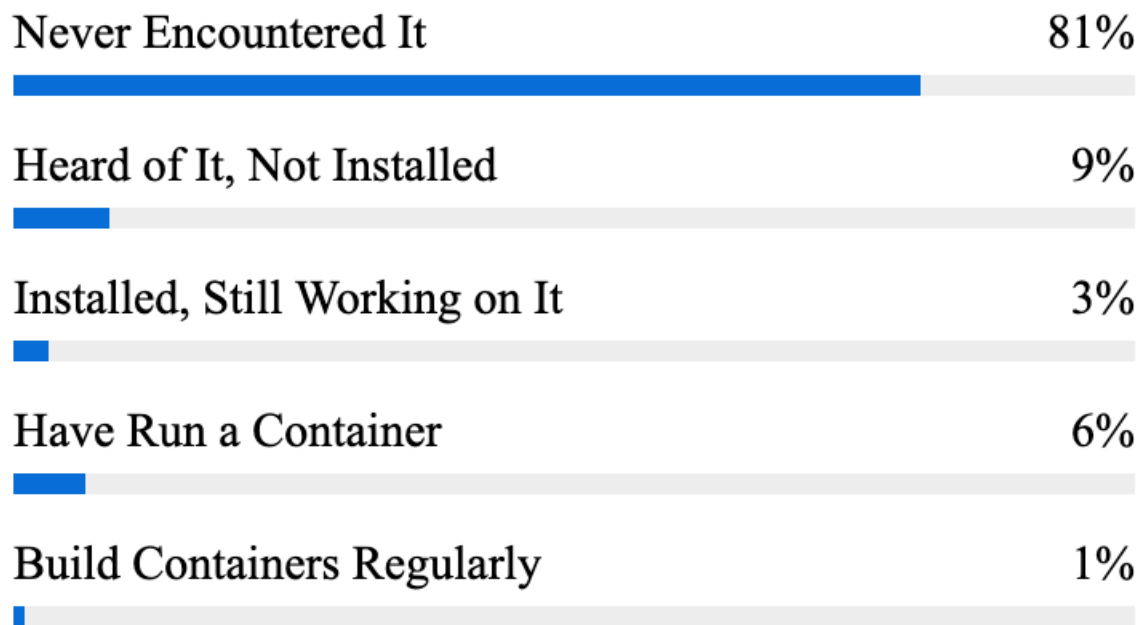
Math



Docker and Jupyter Notebook

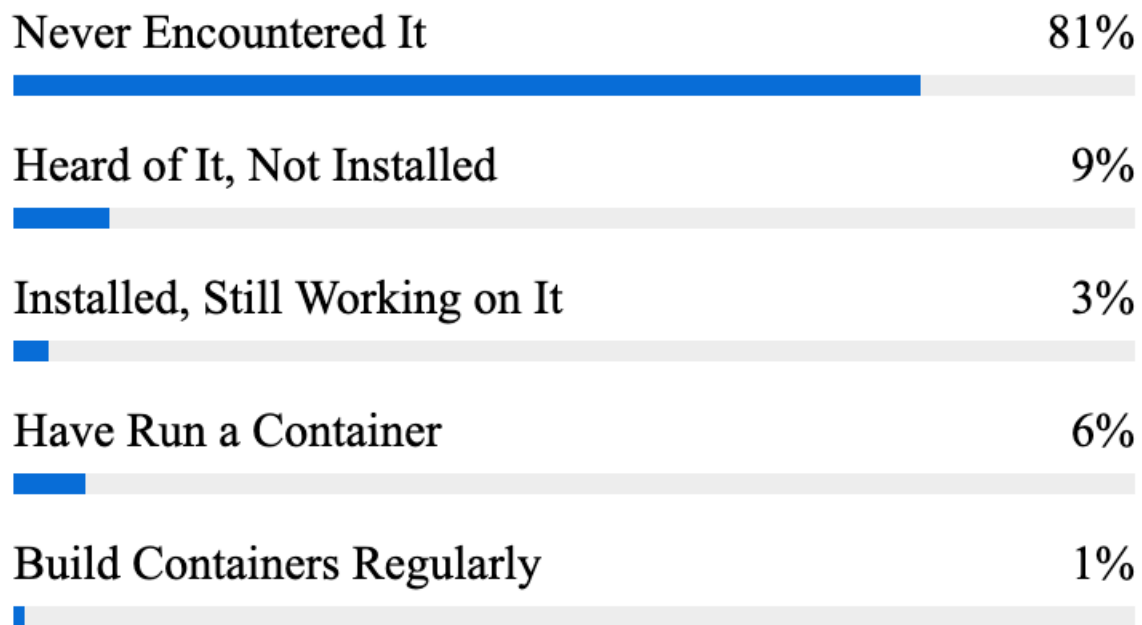
Docker and Jupyter Notebook

Docker

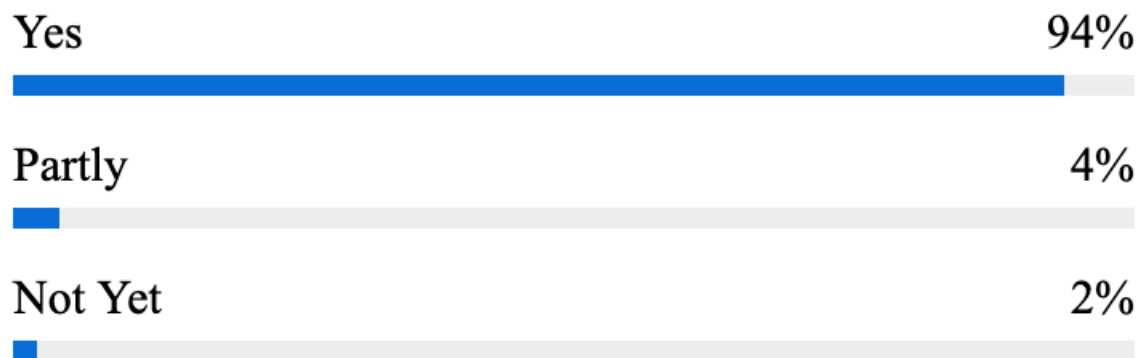


Docker and Jupyter Notebook

Docker



Jupyter Notebook



“

“When we install all the environment (new to me), I have no idea about the purpose of each step.”

“

“When we install all the environment (new to me), I have no idea about the purpose of each step.”

“

“how well do we need to know about the details of the technical stuff we worked on”

“

“When we install all the environment (new to me), I have no idea about the purpose of each step.”

“

“how would you recommend someone without a coding background to self-learn or study in between classes.”

“

“how well do we need to know about the details of the technical stuff we worked on”

“

“When we install all the environment (new to me), I have no idea about the purpose of each step.”

“

“how well do we need to know about the details of the technical stuff we worked on”

“

“how would you recommend someone without a coding background to self-learn or study in between classes.”

“

“How much time should i regularly spend to review all the materials especially regarding coding to make sure that i understand everything?”

“

“When we install all the environment (new to me), I have no idea about the purpose of each step.”

“

“how well do we need to know about the details of the technical stuff we worked on”

“

“how would you recommend someone without a coding background to self-learn or study in between classes.”

“

“How much time should i regularly spend to review all the materials especially regarding coding to make sure that i understand everything?”

“

“Can we have more sessions for practice?”

“

“All the AI models are quite mature, why we are not learning based on an open sourced model, like DeepSeek. Why do we still learn the basics?”

“

“All the AI models are quite mature, why we are not learning based on an open sourced model, like DeepSeek. Why do we still learn the basics?”

“

“I hope to be able to master the latest research methodologies of multimodal models and LLM by the end of the course, and also be proficient in the methods for building agents.”

“

“All the AI models are quite mature, why we are not learning based on an open sourced model, like DeepSeek. Why do we still learn the basics?”

“

“Will we have opportunities to experiment with state-of-the-art LLMs or custom API integrations during the hands-on assignments?”

“

“I hope to be able to master the latest research methodologies of multimodal models and LLM by the end of the course, and also be proficient in the methods for building agents.”

“

“All the AI models are quite mature, why we are not learning based on an open sourced model, like DeepSeek. Why do we still learn the basics?”

“

“Will we have opportunities to experiment with state-of-the-art LLMs or custom API integrations during the hands-on assignments?”

“

“I hope to be able to master the latest research methodologies of multimodal models and LLM by the end of the course, and also be proficient in the methods for building agents.”

“

“How will the upcoming labs bridge the gap between abstract AI concepts and practical, production-ready implementation?”

“

“All the AI models are quite mature, why we are not learning based on an open sourced model, like DeepSeek. Why do we still learn the basics?”

“

“Will we have opportunities to experiment with state-of-the-art LLMs or custom API integrations during the hands-on assignments?”

“

“Mastering how to design practical AI agents and prompt engineering workflows that solve real-world ambiguity, rather than just knowing isolated theory.”

“

“I hope to be able to master the latest research methodologies of multimodal models and LLM by the end of the course, and also be proficient in the methods for building agents.”

“

“How will the upcoming labs bridge the gap between abstract AI concepts and practical, production-ready implementation?”

“

“All the AI models are quite mature, why we are not learning based on an open sourced model, like DeepSeek. Why do we still learn the basics?”

“

“Will we have opportunities to experiment with state-of-the-art LLMs or custom API integrations during the hands-on assignments?”

“

“Mastering how to design practical AI agents and prompt engineering workflows that solve real-world ambiguity, rather than just knowing isolated theory.”

“

“I hope to be able to master the latest research methodologies of multimodal models and LLM by the end of the course, and also be proficient in the methods for building agents.”

“

“How will the upcoming labs bridge the gap between abstract AI concepts and practical, production-ready implementation?”

“

“I would like to train a LLM”

“

“Why we should understand the technicals and how this knowledge will benefit us on an executive level”

“

“Why we should understand the technicals and how this knowledge will benefit us on an executive level”

“

“how well do we need to know about the details of the technical stuff we worked on”

“

“Why we should understand the technicals and how this knowledge will benefit us on an executive level”

“

“Do we need to write a program on our own using any coding languages?”

“

“how well do we need to know about the details of the technical stuff we worked on”

“

“Why we should understand the technicals and how this knowledge will benefit us on an executive level”

“

“how well do we need to know about the details of the technical stuff we worked on”

“

“Do we need to write a program on our own using any coding languages?”

“

“With AI, is it necessary to write code in paper lol. When I worked in LLM company. Employees used AI to write code A LOT lol.”

“

“If I am a complete beginner in Python and do not have any technical background, do I still stand a chance in scoring well (A range) in this course?”

“

“If I am a complete beginner in Python and do not have any technical background, do I still stand a chance in scoring well (A range) in this course?”

“

“Will I failed the exam if I do not have any python experience?”

“

“If I am a complete beginner in Python and do not have any technical background, do I still stand a chance in scoring well (A range) in this course?”

“

“Will I failed the exam if I do not have any python experience?”

“

“I hope the final closed book exam won't be that difficult...”

“

“If I am a complete beginner in Python and do not have any technical background, do I still stand a chance in scoring well (A range) in this course?”

“

“I hope the final closed book exam won't be that difficult...”

“

“Will I failed the exam if I do not have any python experience?”

“

“The 50% final is terrifying”

maib7002.dirk.hk/env_v01.zip

Demo 1: Notebook Workspace

maib7002.dirk.hk/L1.zip

Demo 2: 04_sklearn.ipynb

AI vs ML

Artificial Intelligence (AI)

- Search
- Planning
- Logical Reasoning
- Rule-Based Expert Systems
- Constraint Solving

AI vs ML

Artificial Intelligence (AI)

- Search
- Planning
- Logical Reasoning
- Rule-Based Expert Systems
- Constraint Solving

Machine Learning (ML)

Learn patterns from data

AI vs ML

Artificial Intelligence (AI)

- Search
- Planning
- Logical Reasoning
- Rule-Based Expert Systems
- Constraint Solving

Machine Learning (ML)

Learn patterns from data

ML is a subset of AI

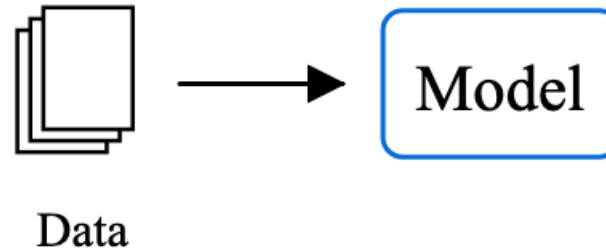
AI vs ML

Artificial Intelligence (AI)

- Search
- Planning
- Logical Reasoning
- Rule-Based Expert Systems
- Constraint Solving

Machine Learning (ML)

Learn patterns from data



ML is a subset of AI

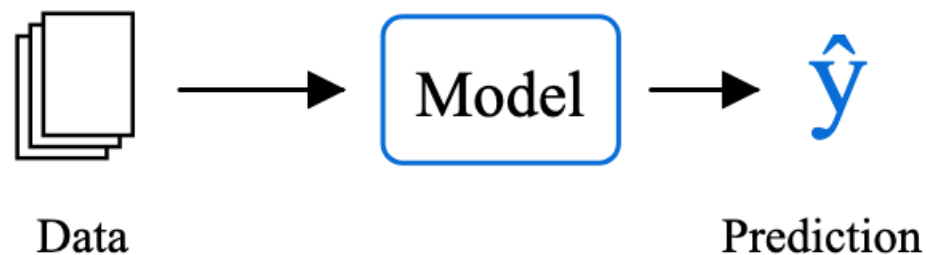
AI vs ML

Artificial Intelligence (AI)

- Search
- Planning
- Logical Reasoning
- Rule-Based Expert Systems
- Constraint Solving

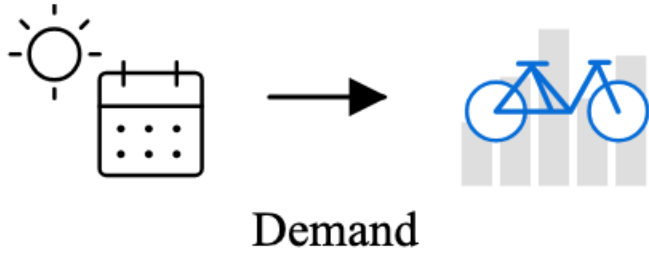
Machine Learning (ML)

Learn patterns from data

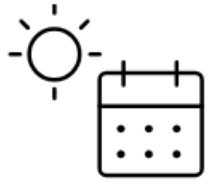
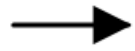


ML is a subset of AI

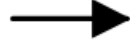
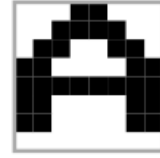
What Can We Predict?



What Can We Predict?

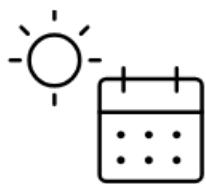
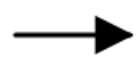


Demand

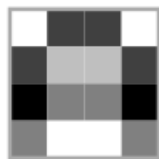
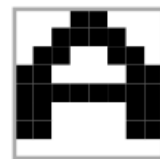


Super-Resolution

What Can We Predict?



Demand

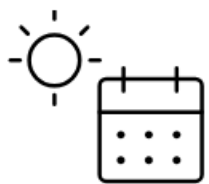
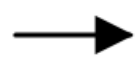


Super-Resolution

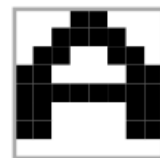


Text

What Can We Predict?



Demand



Super-Resolution

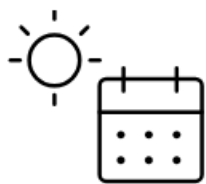
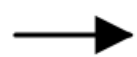


Text

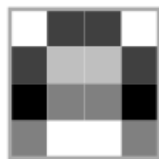
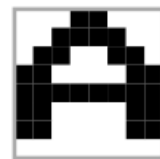


Image

What Can We Predict?



Demand



Super-Resolution



Text

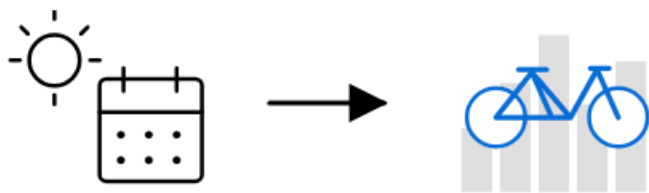


Image

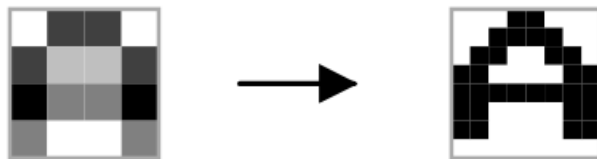
What Can We Predict?



Speech



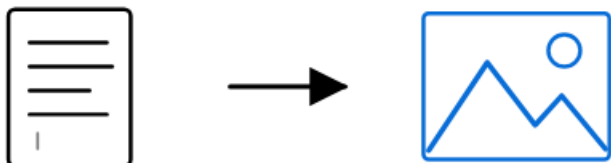
Demand



Super-Resolution



Text



Image

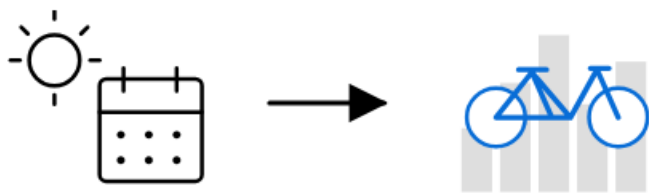
What Can We Predict?



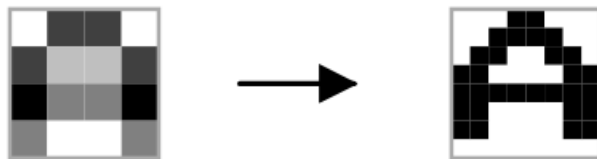
Speech



Fraud



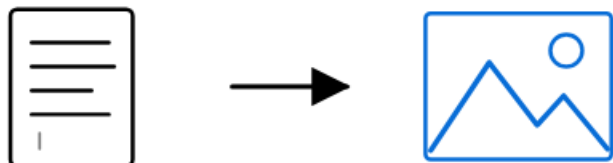
Demand



Super-Resolution



Text



Image

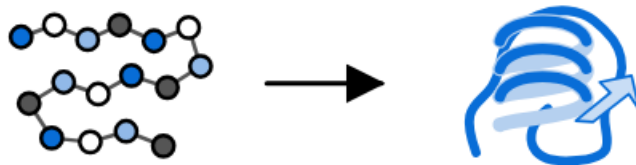
What Can We Predict?



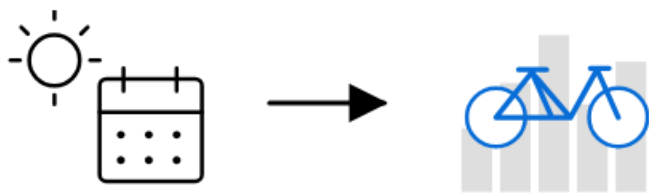
Speech



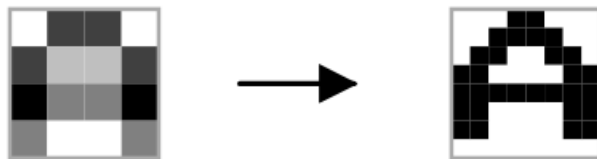
Fraud



Structure



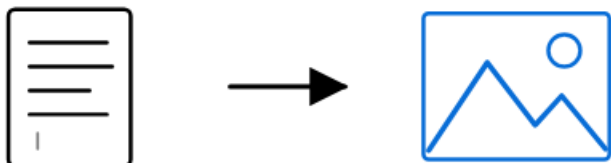
Demand



Super-Resolution



Text



Image

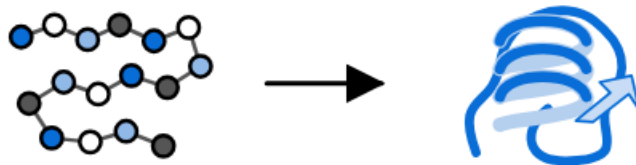
What Can We Predict?



Speech



Fraud



Structure



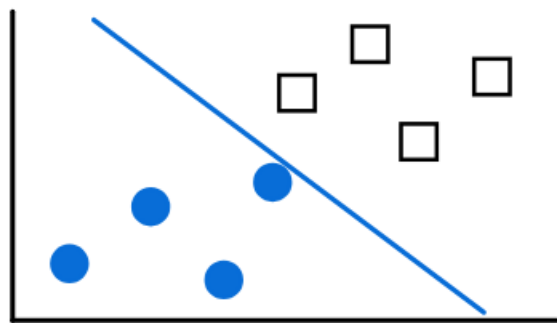
Recommendation

Three Ways to Learn

Three Ways to Learn

Supervised

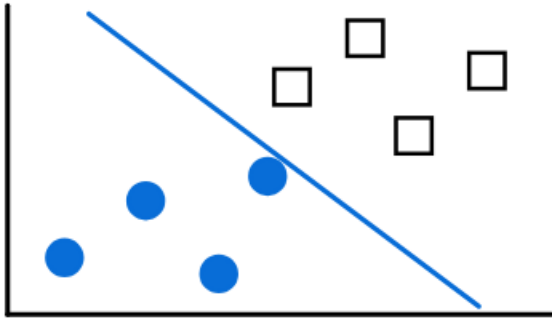
Known answers



Three Ways to Learn

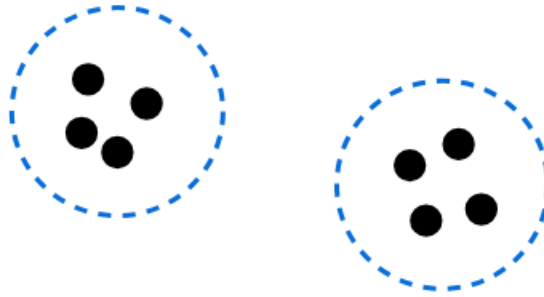
Supervised

Known answers



Unsupervised

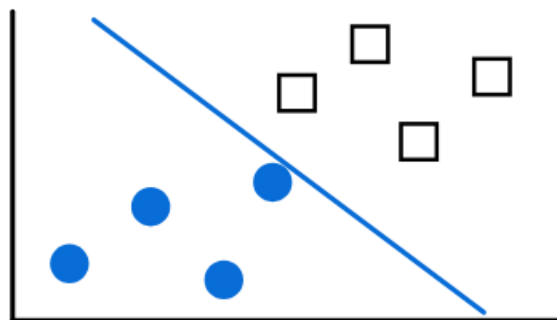
No supplied answers



Three Ways to Learn

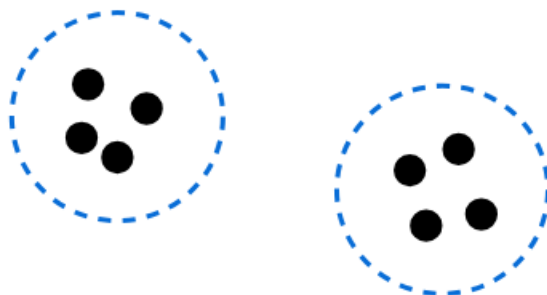
Supervised

Known answers



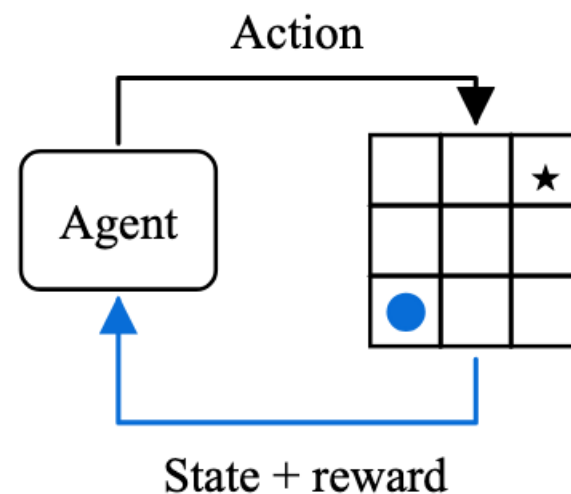
Unsupervised

No supplied answers



Reinforcement

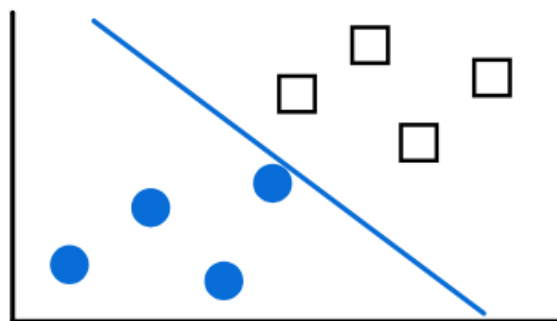
Reward feedback



Three Ways to Learn

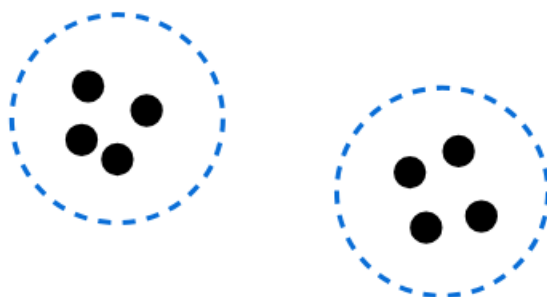
Supervised

Known answers



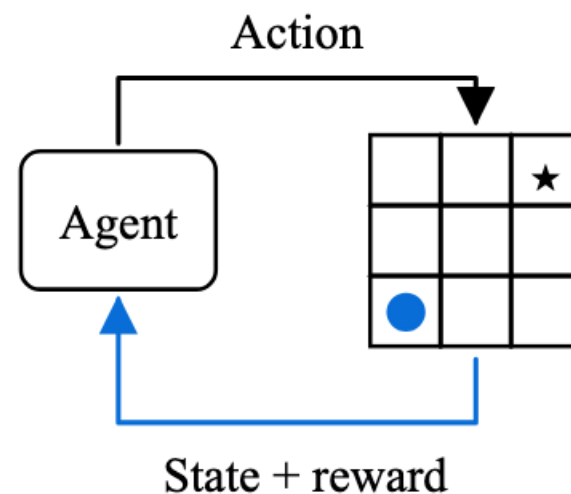
Unsupervised

No supplied answers



Reinforcement

Reward feedback

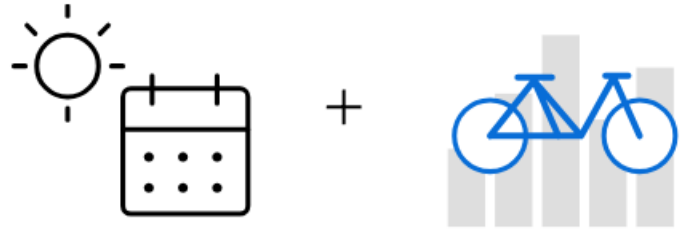


Today's focus

Training vs Prediction

Training

Past days with known answers



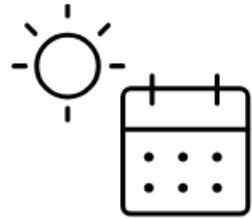
Weather + calendar

Recorded rentals

Training vs Prediction

Training

Past days with known answers



Weather + calendar

+



Recorded rentals

`fit()`

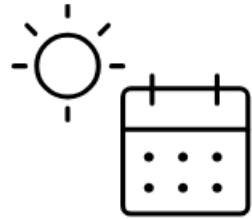


Trained Model

Training vs Prediction

Training

Past days with known answers



+



fit()

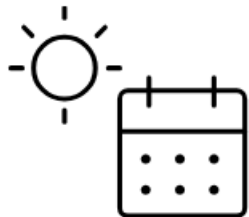


Trained Model

Weather + calendar

Recorded rentals

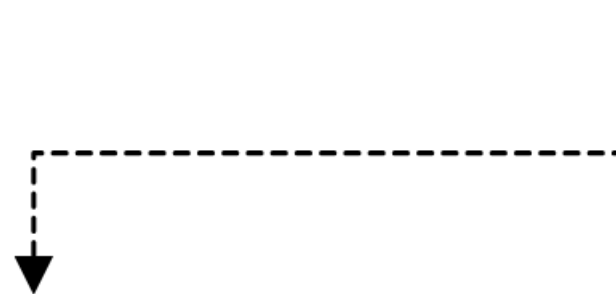
Prediction



Unseen day



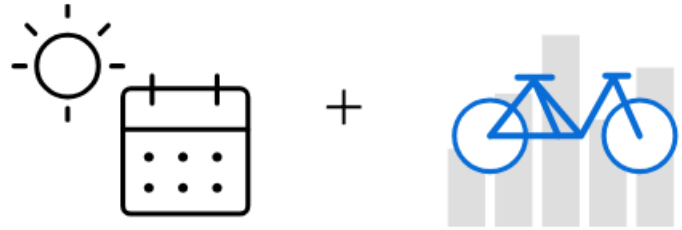
Trained Model



Training vs Prediction

Training

Past days with known answers



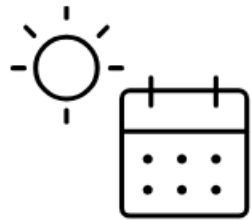
Weather + calendar

Recorded rentals

fit()

Trained Model

Prediction



Unseen day

Trained Model

predict()

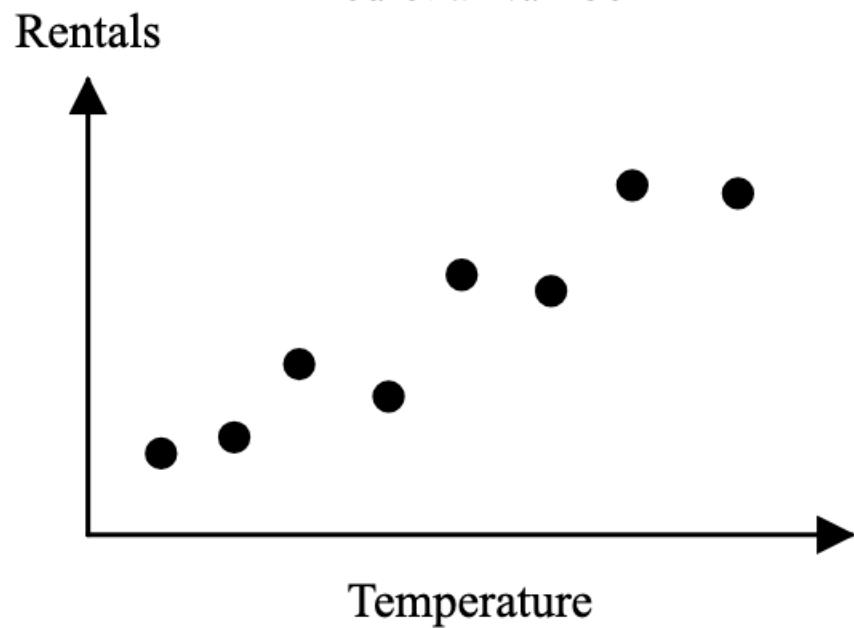


1,467 rentals

What Kind of Answer?

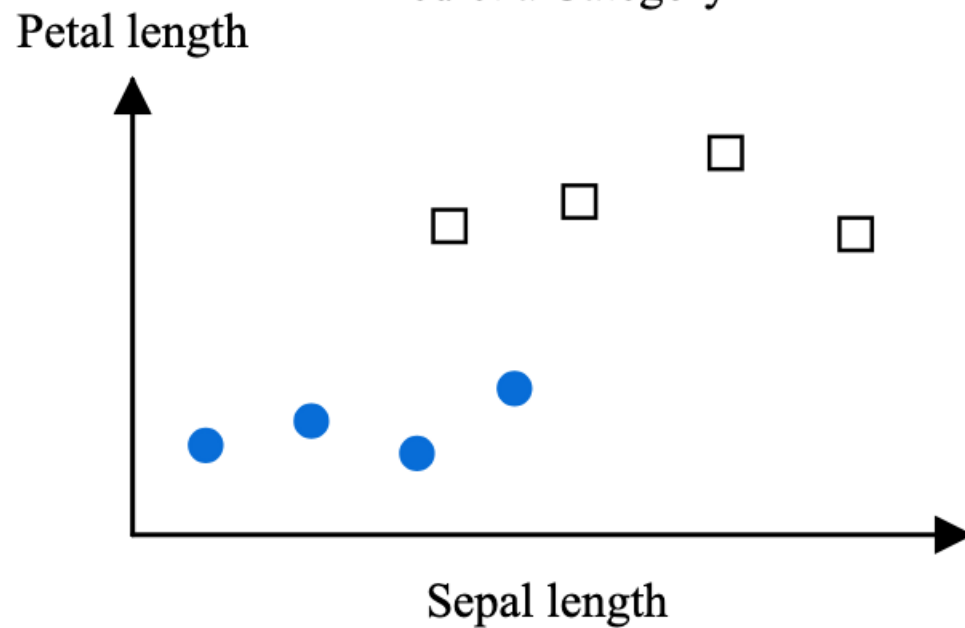
Regression

Predict a Number



Classification

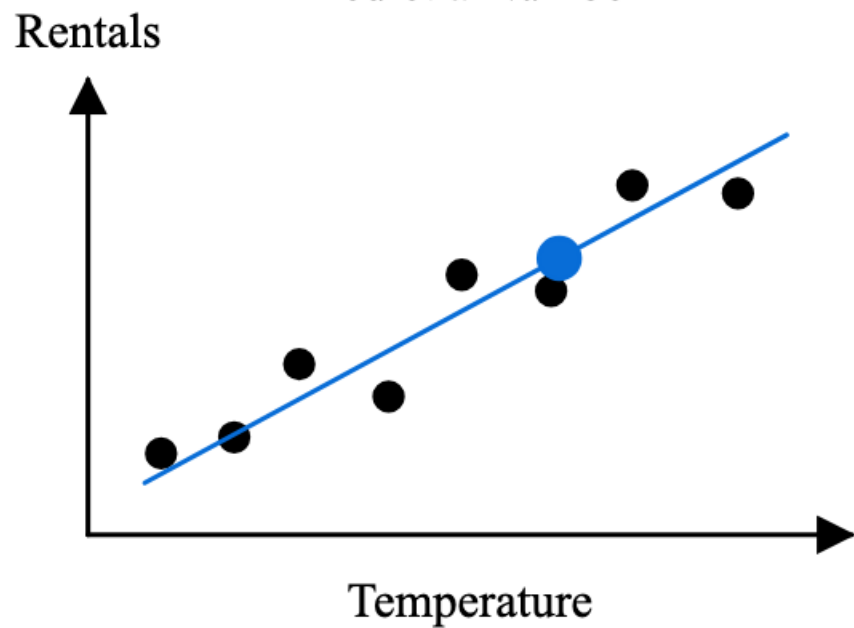
Predict a Category



What Kind of Answer?

Regression

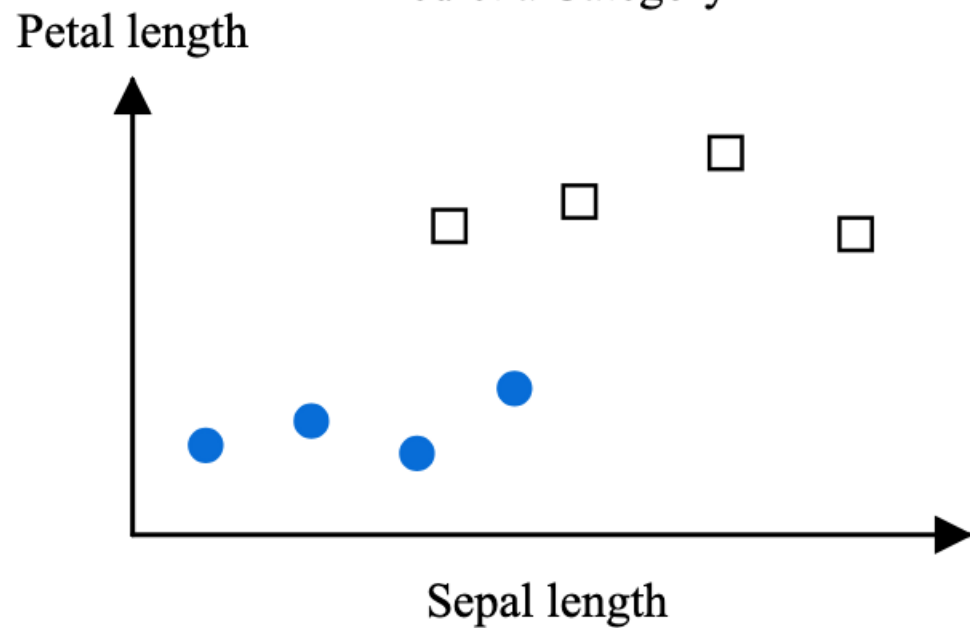
Predict a Number



1,467 rentals

Classification

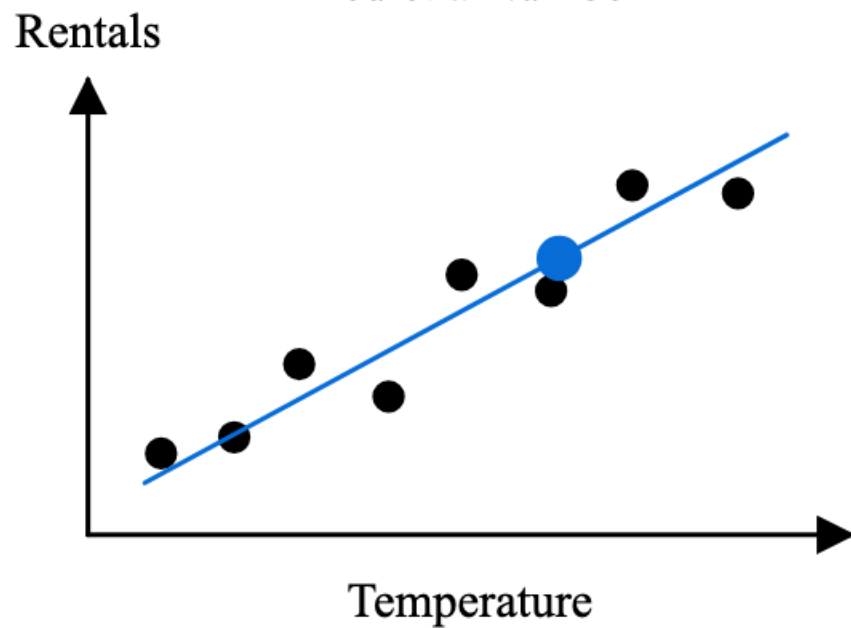
Predict a Category



What Kind of Answer?

Regression

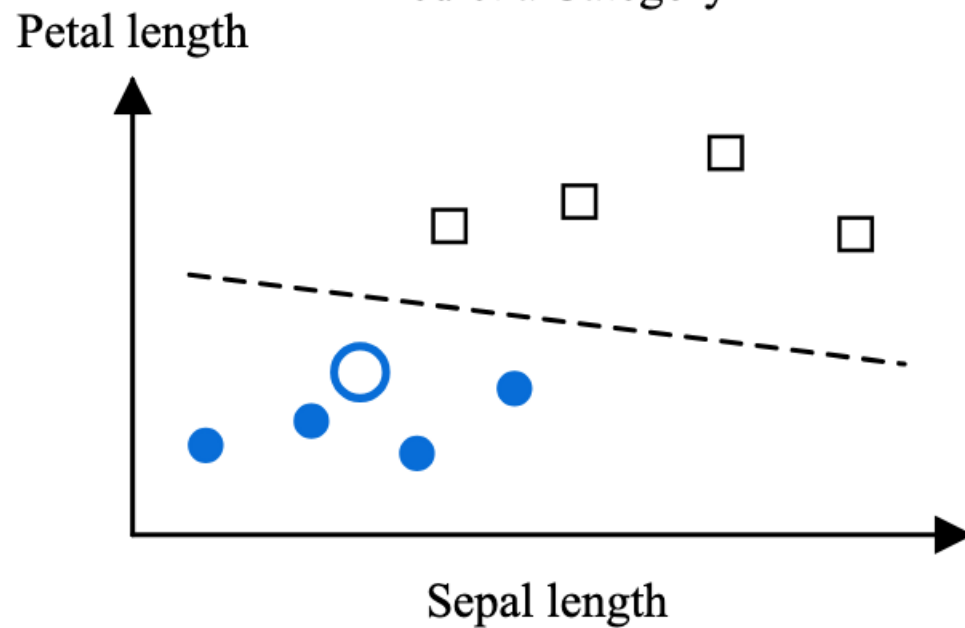
Predict a Number



1,467 rentals

Classification

Predict a Category

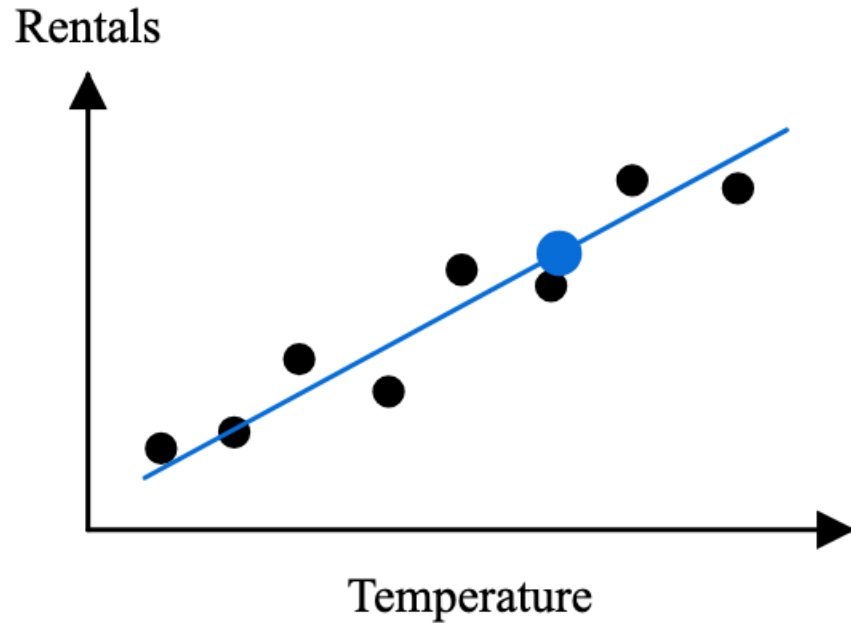


Setosa or versicolor

What Kind of Answer?

Regression

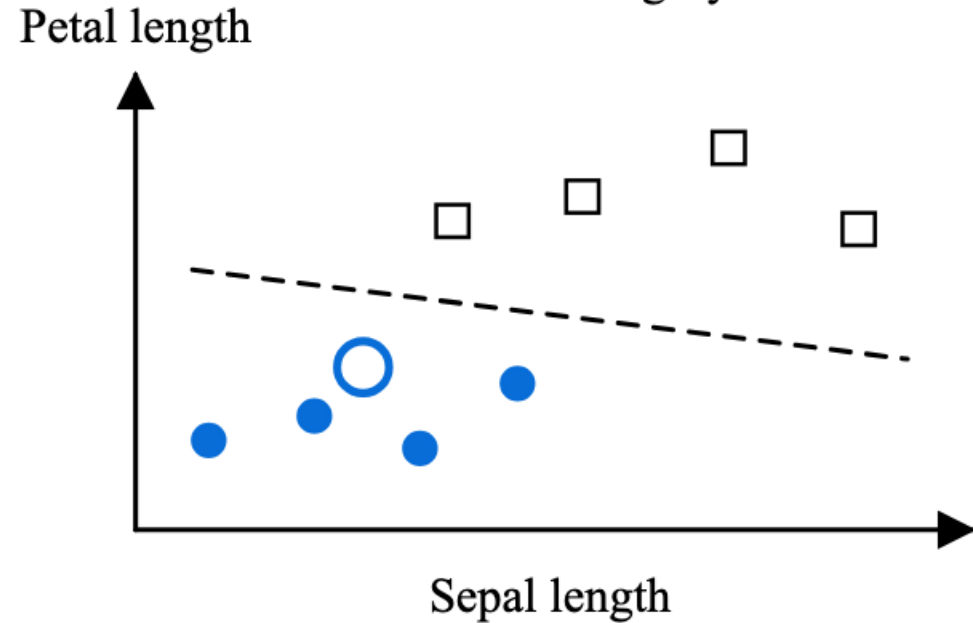
Predict a Number



1,467 rentals

Classification

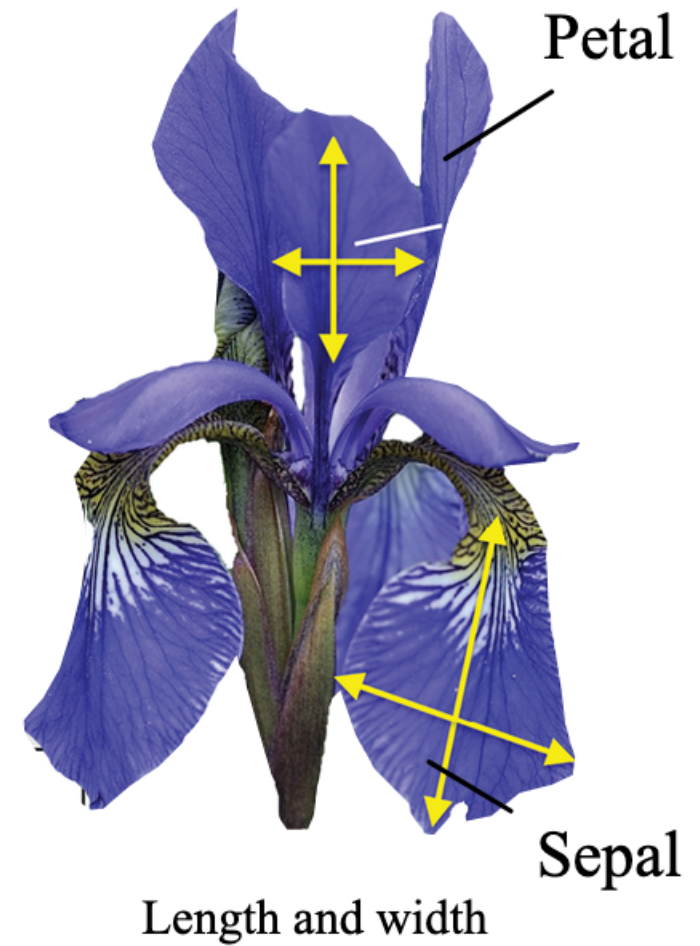
Predict a Category



Setosa or versicolor

Both learn from examples with known target answers

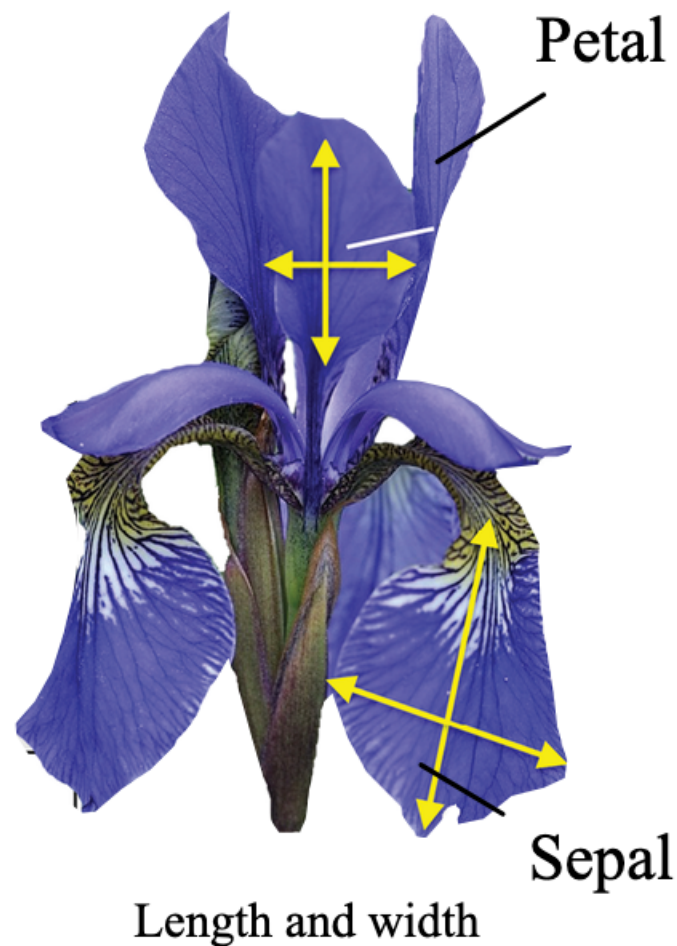
Examples, Features, and Labels



Examples, Features, and Labels

Measurements in cm

	Sepal length	Sepal width	Petal length	Petal width	Class label
1	5.1	3.5	1.4	0.2	Setosa
2	4.9	3.0	1.4	0.2	Setosa
...					
51	7.0	3.2	4.7	1.4	Versicolor
...					
150	5.9	3.0	5.1	1.8	Virginica



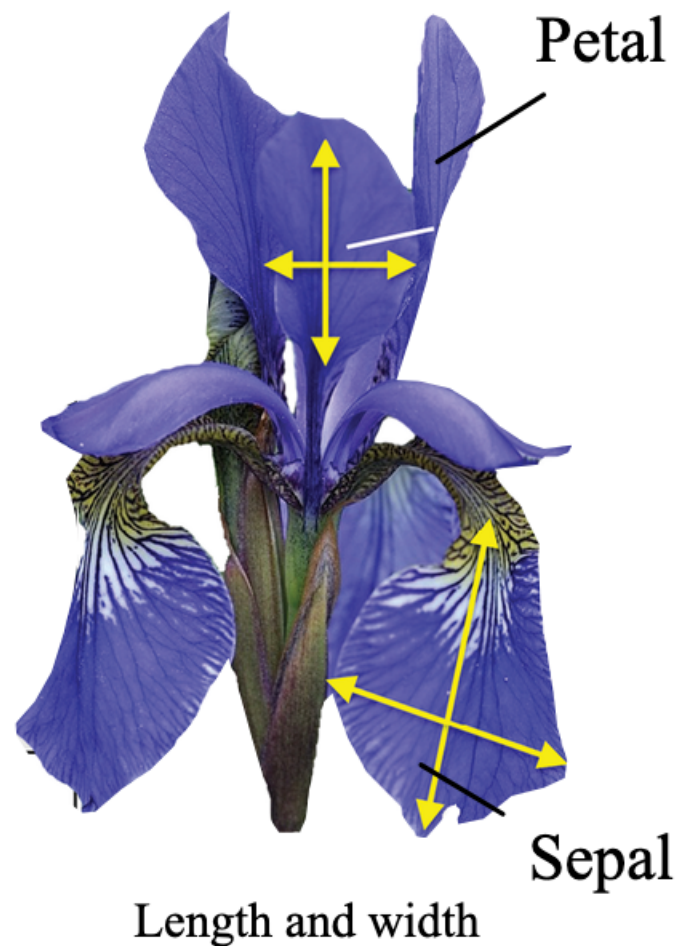
Examples, Features, and Labels

Samples

(instances, observations)

Measurements in cm

	Sepal length	Sepal width	Petal length	Petal width	Class label
1	5.1	3.5	1.4	0.2	Setosa
2	4.9	3.0	1.4	0.2	Setosa
...					
51	7.0	3.2	4.7	1.4	Versicolor
...					
150	5.9	3.0	5.1	1.8	Virginica



Examples, Features, and Labels

Samples

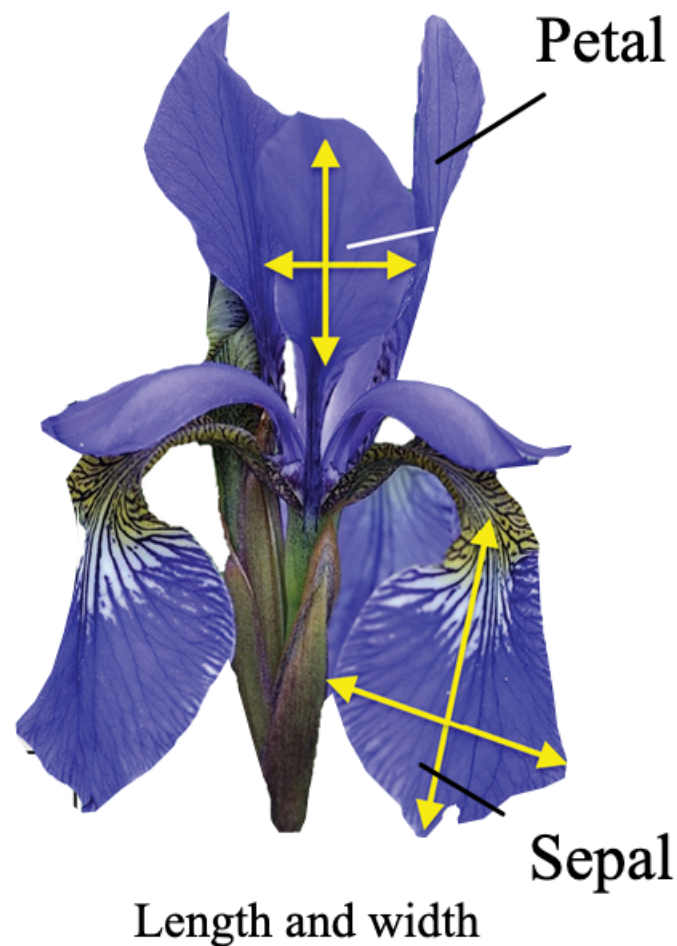
(instances, observations)

Measurements in cm

	Sepal length	Sepal width	Petal length	Petal width	Class label
1	5.1	3.5	1.4	0.2	Setosa
2	4.9	3.0	1.4	0.2	Setosa
...					
51	7.0	3.2	4.7	1.4	Versicolor
...					
150	5.9	3.0	5.1	1.8	Virginica

Features

(inputs, attributes, measurements, dimensions)



Examples, Features, and Labels

Samples

(instances, observations)

Measurements in cm

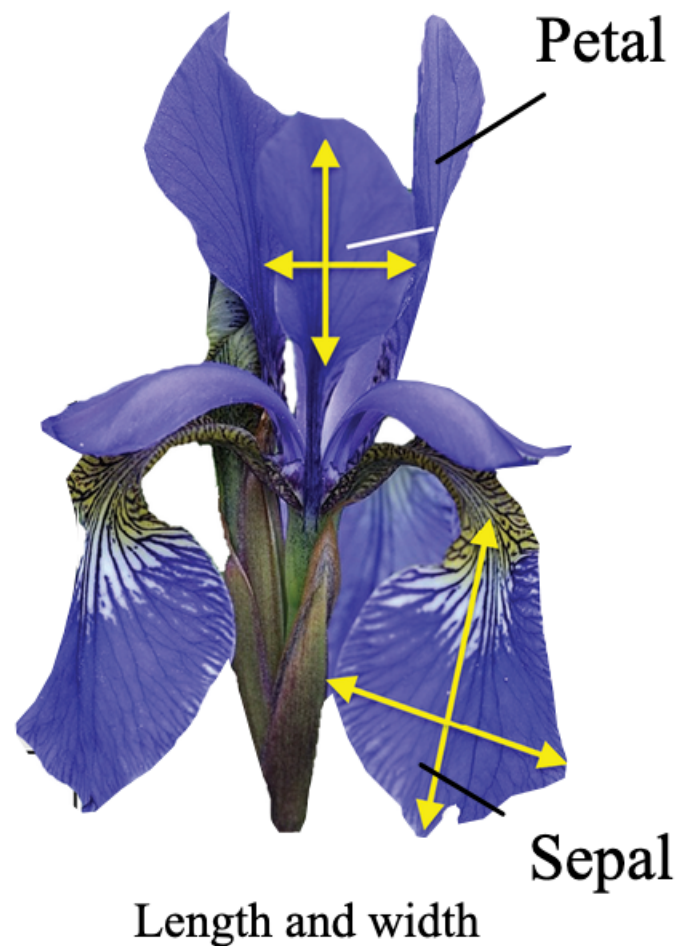
	Sepal length	Sepal width	Petal length	Petal width	Class label
1	5.1	3.5	1.4	0.2	Setosa
2	4.9	3.0	1.4	0.2	Setosa
...					
51	7.0	3.2	4.7	1.4	Versicolor
...					
150	5.9	3.0	5.1	1.8	Virginica

Features

(inputs, attributes, measurements, dimensions)

Class Labels

(targets, outcomes)



From a Table to X and y

Sepal length (cm)	Petal length (cm)	Species
5.1	1.4	setosa
4.9	1.4	setosa
7.0	4.7	versicolor
6.4	4.5	versicolor

From a Table to X and y

X

Sepal length (cm)	Petal length (cm)
5.1	1.4
4.9	1.4
7.0	4.7
6.4	4.5

4 examples $\times$ 2 features

y

Species
setosa
setosa
versicolor
versicolor

4 labels

From a Table to X and y

X

Sepal length (cm)	Petal length (cm)
5.1	1.4
4.9	1.4
7.0	4.7
6.4	4.5

4 examples $\times$ 2 features

y

Species
0
0
1
1

4 labels

setosa $\rightarrow$ 0 versicolor $\rightarrow$ 1

From a Table to X and y

X			y
Sepal length (cm)	Petal length (cm)		Species
5.1	1.4	Same flower -----	0
4.9	1.4		0
7.0	4.7		1
6.4	4.5		1
4 examples × 2 features			4 labels
setosa → 0 versicolor → 1			

Regression

PY main.py

```
1 from sklearn.linear_model import LinearRegression
2 # Toy temperatures (°C) and daily rental counts
3 X = [[10],
4      [20],
5      [30]]
6 y = [100,
7      200,
8      300]
9 model = LinearRegression()
10 model.fit(X, y)
11 print(f"Training:\n{'X':<14} y")
12 for x, label in zip(X, y):
13     print(f"{str(x):<14} {label}")
14 temperature = 25
15 prediction = round(model.predict([[temperature]])[0])
16 print("\nPrediction:")
17 print(f"{temperature}°C → {prediction} rentals")
```

Classification

PY main.py

```
1 from sklearn.tree import DecisionTreeClassifier
2 # Sepal length and petal length (cm)
3 X = [[5.1, 1.4],
4       [4.9, 1.4],
5       [7.0, 4.7],
6       [6.4, 4.5]]
7 y = ["setosa",
8       "setosa",
9       "versicolor",
10      "versicolor"]
11 model = DecisionTreeClassifier(
12     max_depth=1, random_state=0)
13 model.fit(X, y)
14 print(f"Training:\n{'X':<14} y")
15 for x, label in zip(X, y):
16     print(f"{str(x):<14} {label}")
17 flower = [5.0, 1.5]
18 prediction = model.predict([flower])
19 print("\nPrediction:")
20 print(f"{flower} → {prediction[0]}")
```

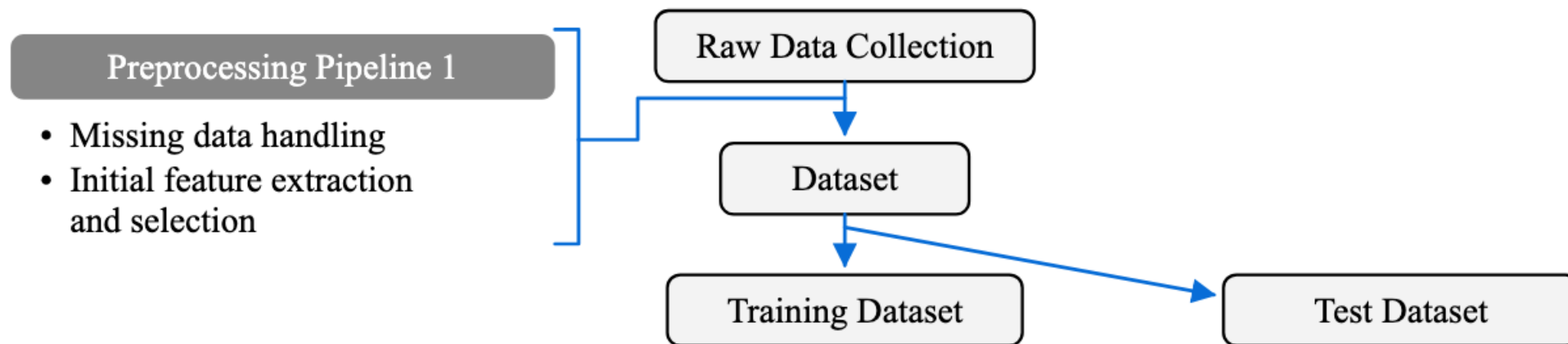
ML Roadmap

Raw Data Collection

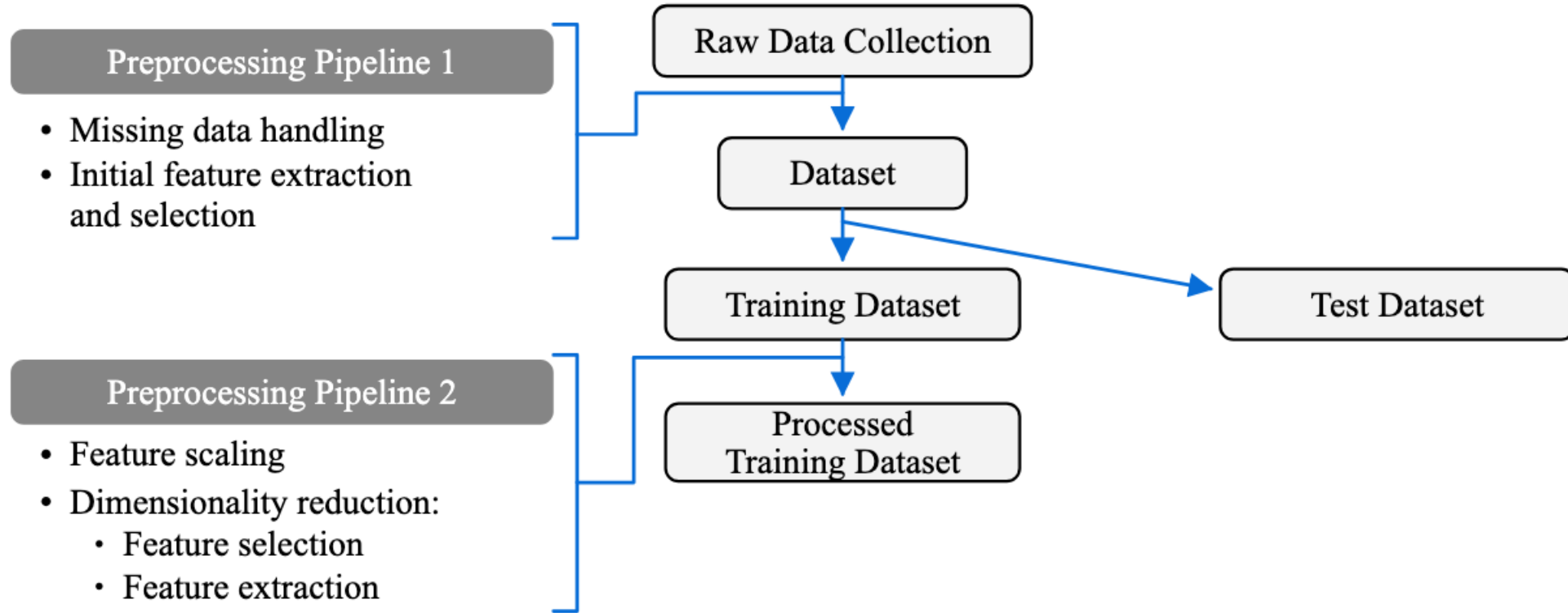
ML Roadmap



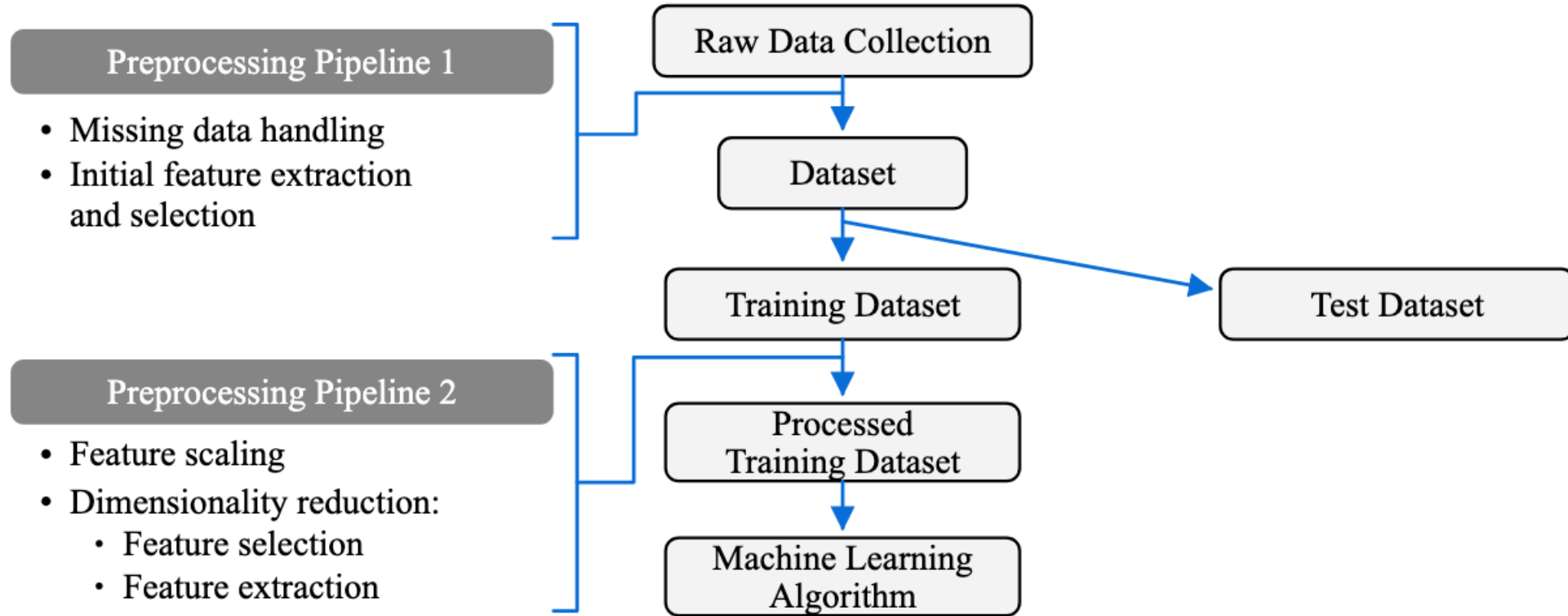
ML Roadmap



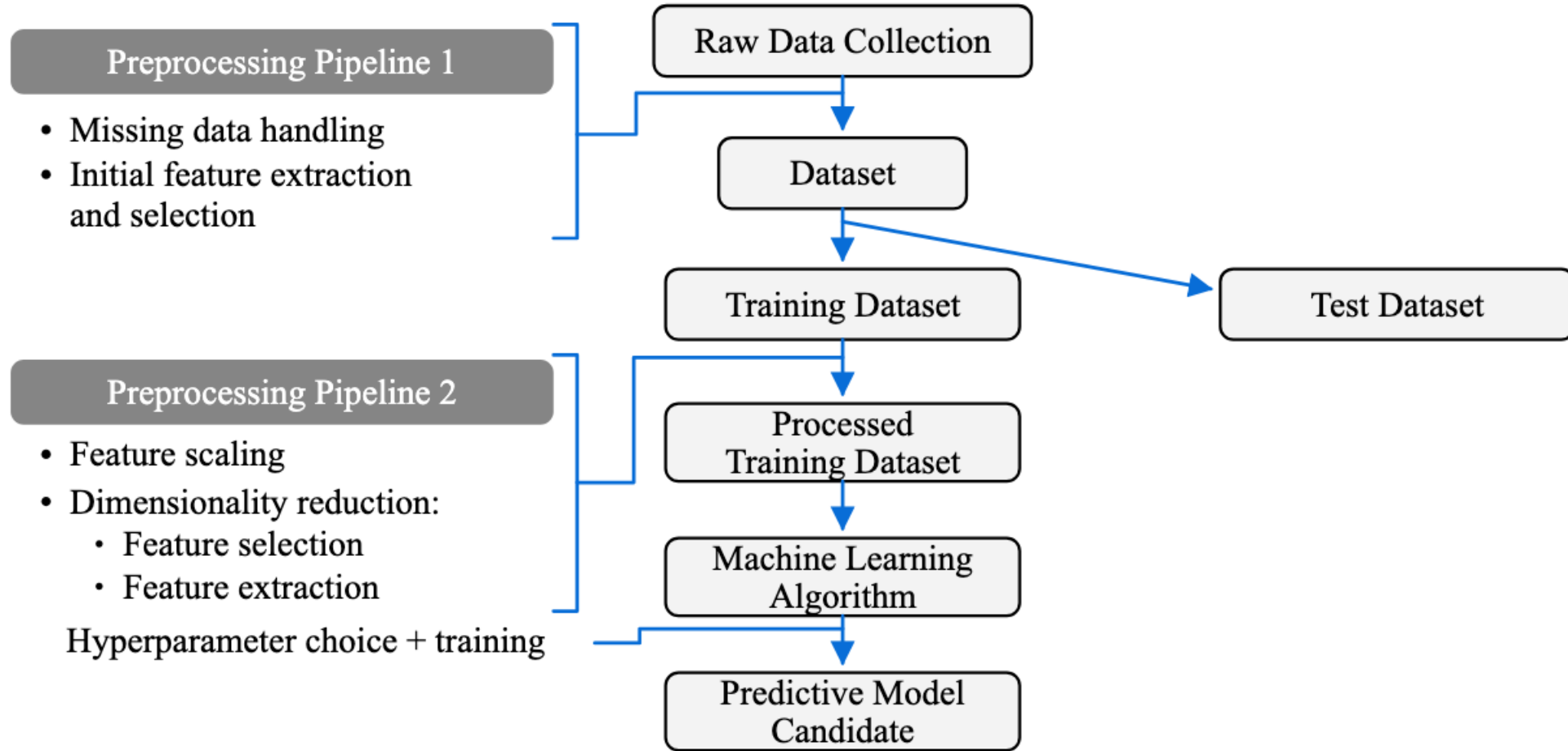
ML Roadmap



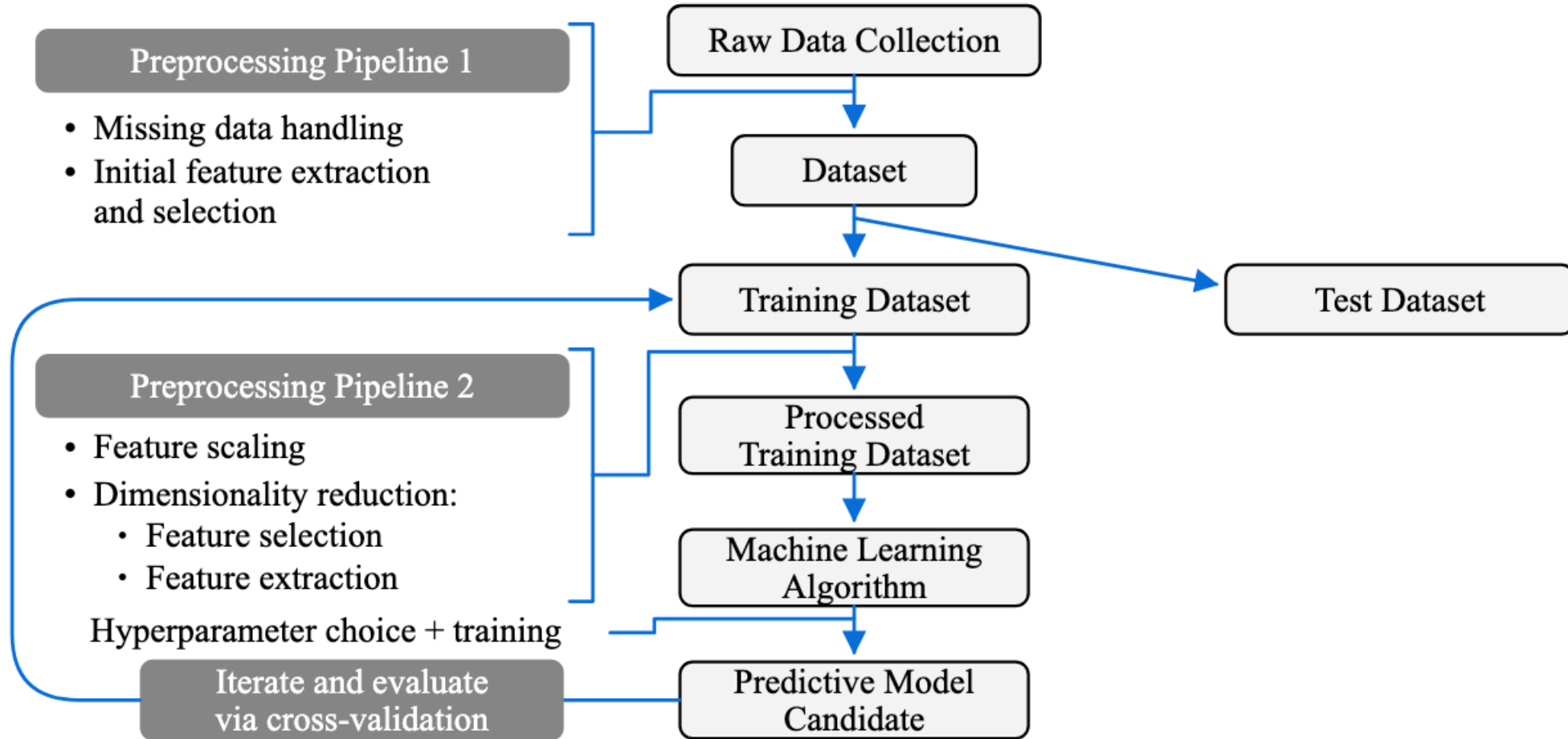
ML Roadmap



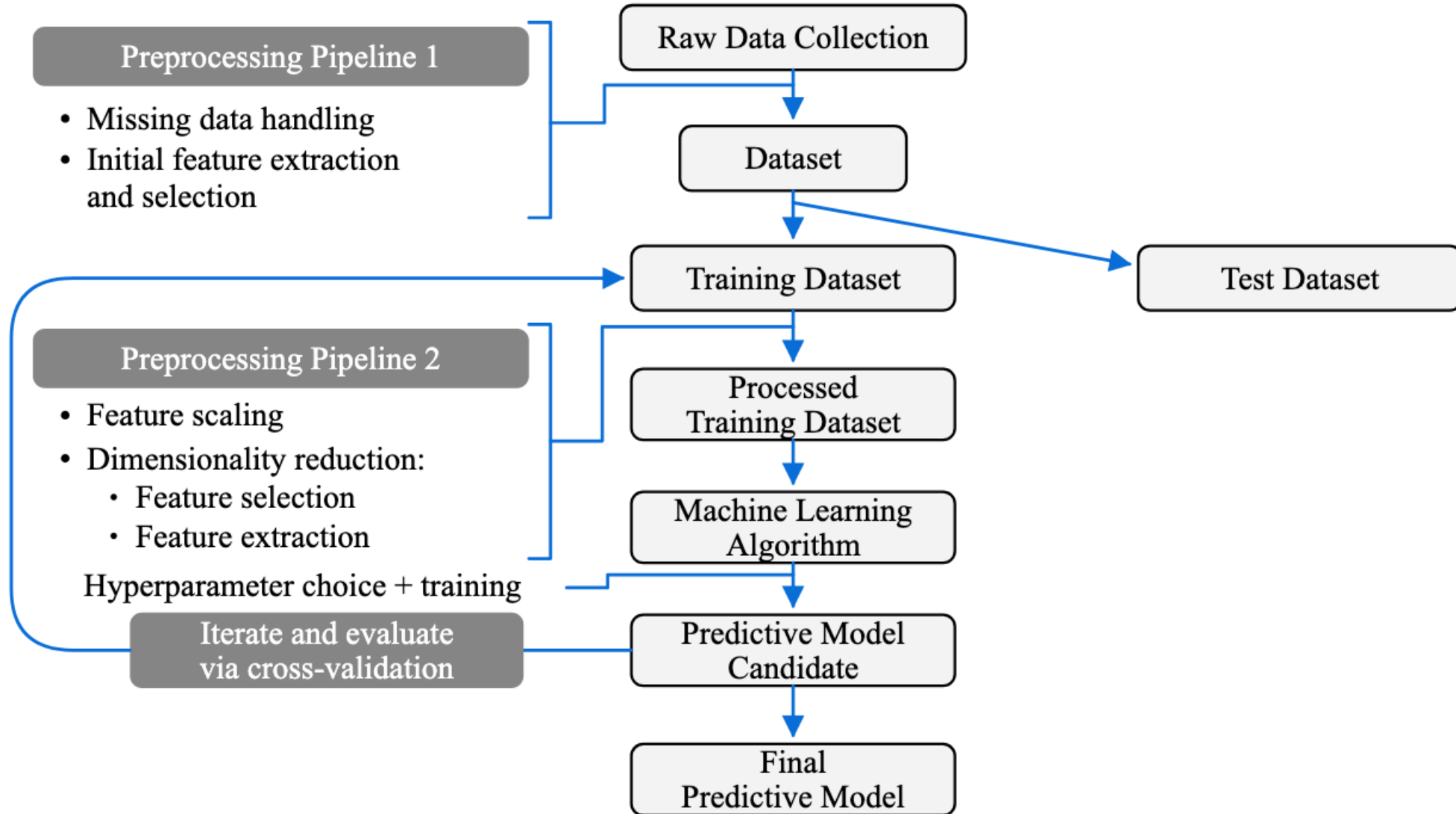
ML Roadmap



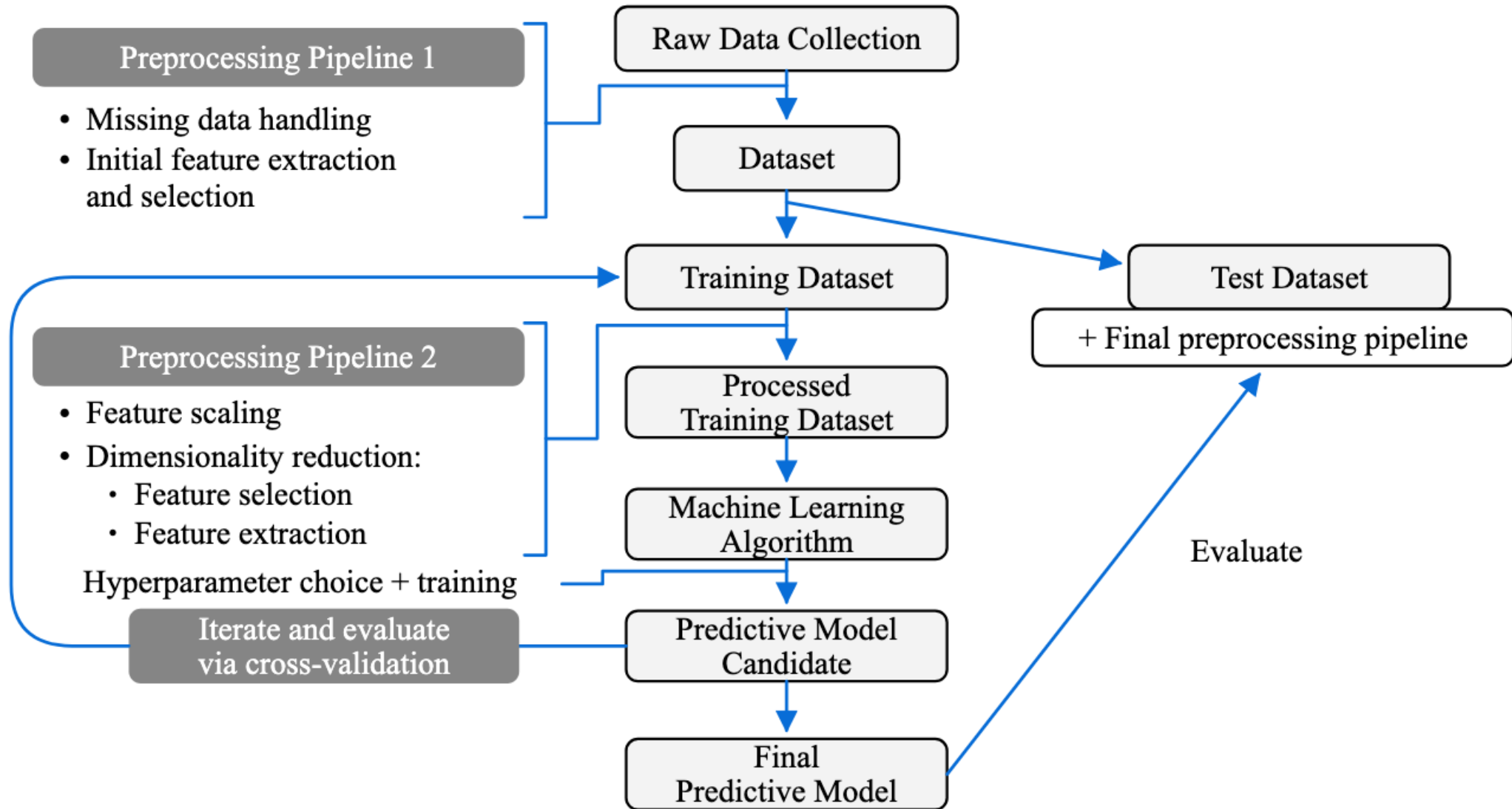
ML Roadmap



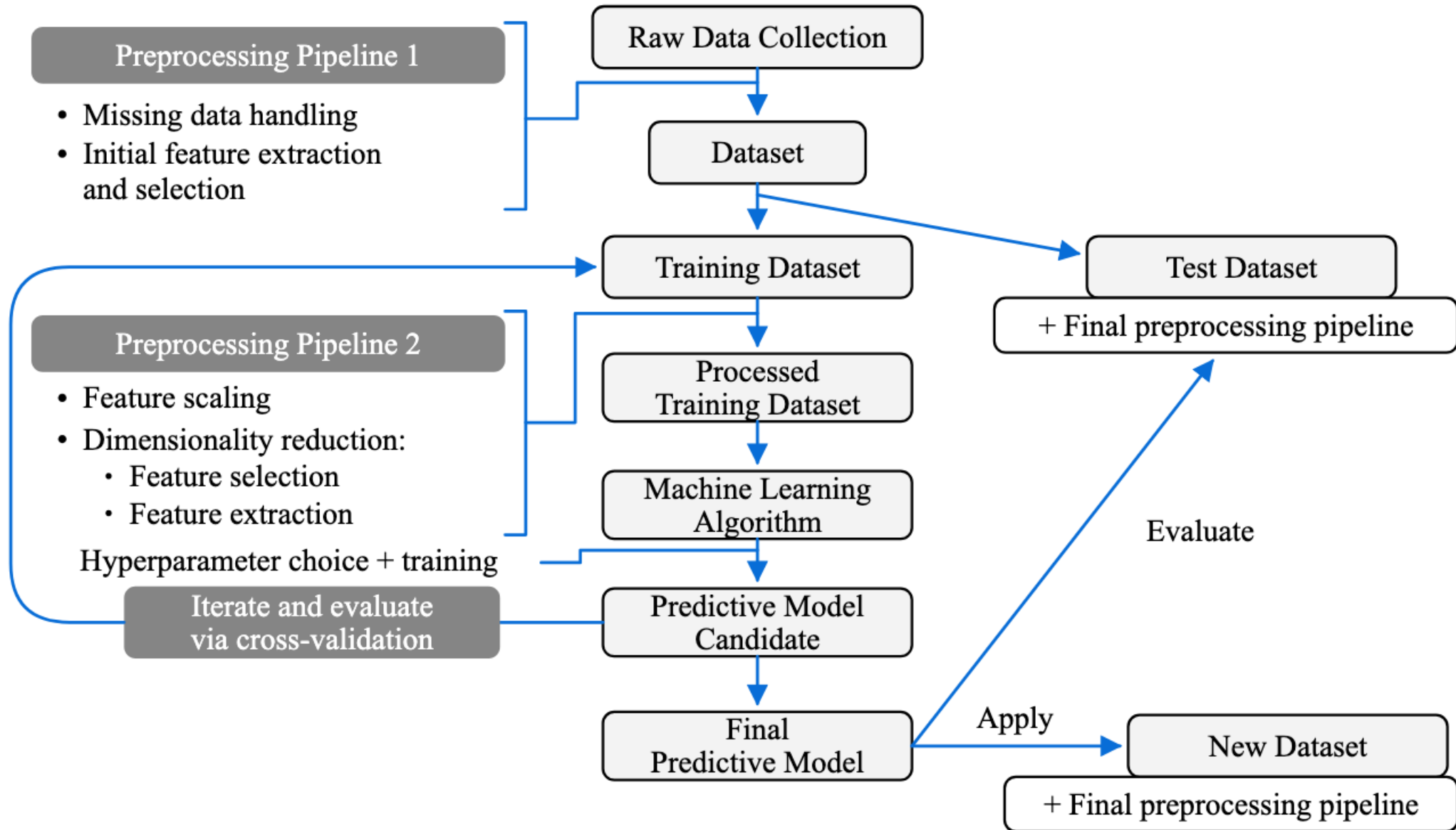
ML Roadmap



ML Roadmap



ML Roadmap



Assignment 1

Demo 3

Matrix and Vector Notation

Measurements: $\mathbf{X} \in \mathbb{R}^{150 \times 4}$

$$\begin{bmatrix} x_1^{(1)} & x_2^{(1)} & x_3^{(1)} & x_4^{(1)} \\ x_1^{(2)} & x_2^{(2)} & x_3^{(2)} & x_4^{(2)} \\ \vdots & \vdots & \vdots & \vdots \\ x_1^{(150)} & x_2^{(150)} & x_3^{(150)} & x_4^{(150)} \end{bmatrix}$$

Matrix and Vector Notation

Measurements: $\mathbf{X} \in \mathbb{R}^{150 \times 4}$

One Entry

$$x_j^{(i)}$$

i : example (row)

j : feature (column)

$$\begin{bmatrix} x_1^{(1)} & x_2^{(1)} & x_3^{(1)} & x_4^{(1)} \\ x_1^{(2)} & x_2^{(2)} & x_3^{(2)} & x_4^{(2)} \\ \vdots & \vdots & \vdots & \vdots \\ x_1^{(150)} & x_2^{(150)} & x_3^{(150)} & x_4^{(150)} \end{bmatrix}$$

Matrix and Vector Notation

Measurements: $\mathbf{X} \in \mathbb{R}^{150 \times 4}$

One Entry

$$x_j^{(i)}$$

i : example (row)

j : feature (column)

$$\begin{bmatrix} x_1^{(1)} & x_2^{(1)} & x_3^{(1)} & x_4^{(1)} \\ x_1^{(2)} & x_2^{(2)} & x_3^{(2)} & x_4^{(2)} \\ \vdots & \vdots & \vdots & \vdots \\ x_1^{(150)} & x_2^{(150)} & x_3^{(150)} & x_4^{(150)} \end{bmatrix}$$



Row Vector · 4 Features

$$\mathbf{x}^{(i)} = \begin{bmatrix} x_1^{(i)} & x_2^{(i)} & x_3^{(i)} & x_4^{(i)} \end{bmatrix}$$

Matrix and Vector Notation

Measurements: $\mathbf{X} \in \mathbb{R}^{150 \times 4}$

One Entry

$$x_j^{(i)}$$

i : example (row)

j : feature (column)

$$\begin{bmatrix} x_1^{(1)} & x_2^{(1)} & x_3^{(1)} & x_4^{(1)} \\ x_1^{(2)} & x_2^{(2)} & x_3^{(2)} & x_4^{(2)} \\ \vdots & \vdots & \vdots & \vdots \\ x_1^{(150)} & x_2^{(150)} & x_3^{(150)} & x_4^{(150)} \end{bmatrix}$$

Column Vector

$$\mathbf{x}_j = \begin{bmatrix} x_j^{(1)} \\ x_j^{(2)} \\ \vdots \\ x_j^{(150)} \end{bmatrix}$$

150 Examples

Row Vector · 4 Features

$$\mathbf{x}^{(i)} = \begin{bmatrix} x_1^{(i)} & x_2^{(i)} & x_3^{(i)} & x_4^{(i)} \end{bmatrix}$$

Matrix and Vector Notation

Measurements: $\mathbf{X} \in \mathbb{R}^{150 \times 4}$

One Entry

$$x_j^{(i)}$$

i : example (row)

j : feature (column)

$$\begin{bmatrix} x_1^{(1)} & x_2^{(1)} & x_3^{(1)} & x_4^{(1)} \\ x_1^{(2)} & x_2^{(2)} & x_3^{(2)} & x_4^{(2)} \\ \vdots & \vdots & \vdots & \vdots \\ x_1^{(150)} & x_2^{(150)} & x_3^{(150)} & x_4^{(150)} \end{bmatrix}$$

Column Vector

$$\mathbf{x}_j = \begin{bmatrix} x_j^{(1)} \\ x_j^{(2)} \\ \vdots \\ x_j^{(150)} \end{bmatrix}$$

150 Examples

Row Vector · 4 Features

$$\mathbf{x}^{(i)} = \begin{bmatrix} x_1^{(i)} & x_2^{(i)} & x_3^{(i)} & x_4^{(i)} \end{bmatrix}$$

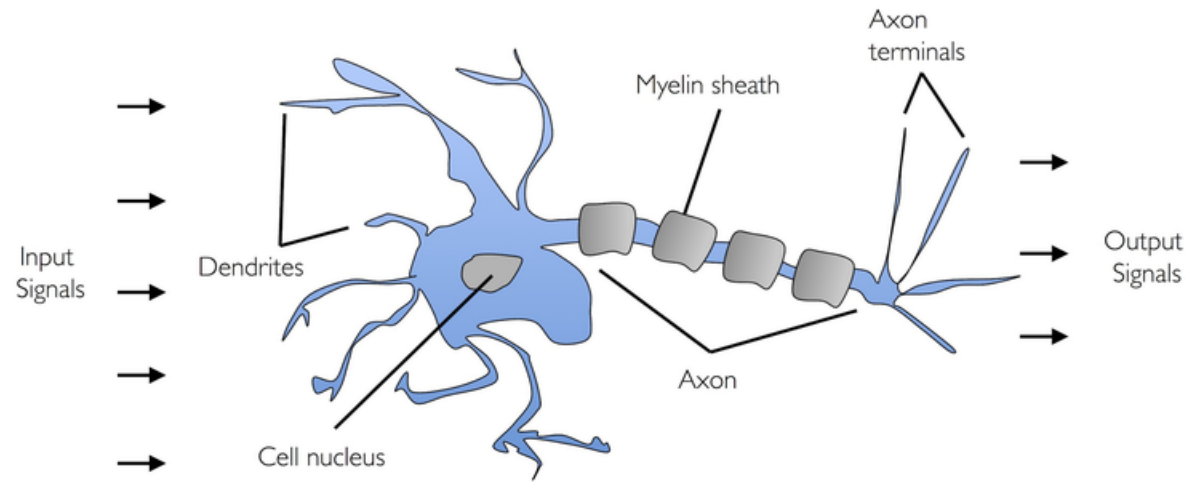
$\mathbf{y}$: Species Labels

$y^{(i)} \in \{\text{Setosa, Versicolor, Virginica}\}$

150 Labels · 3 Classes

From Biological to Artificial Neurons

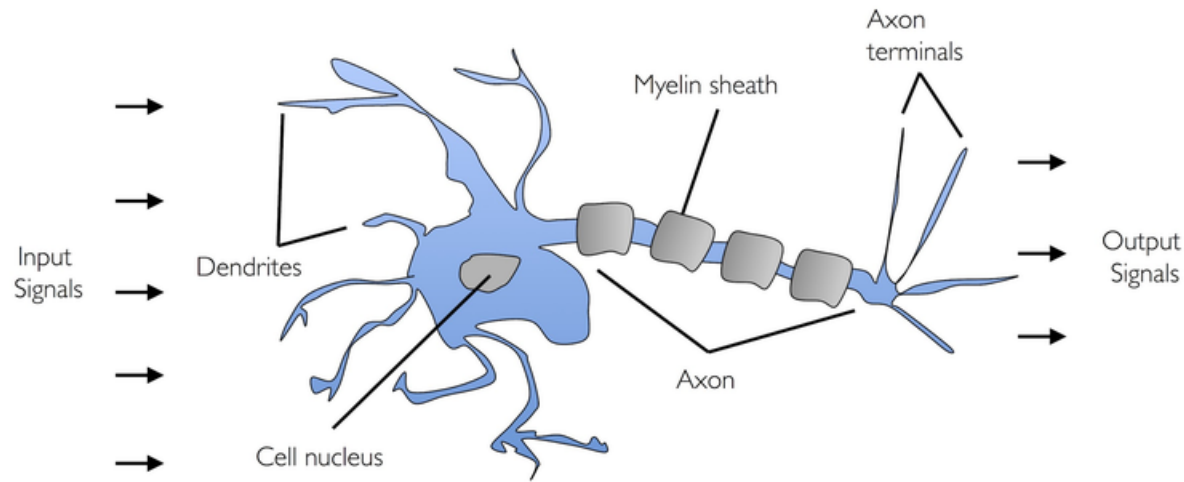
Biological Neuron



Receive → Combine → Transmit

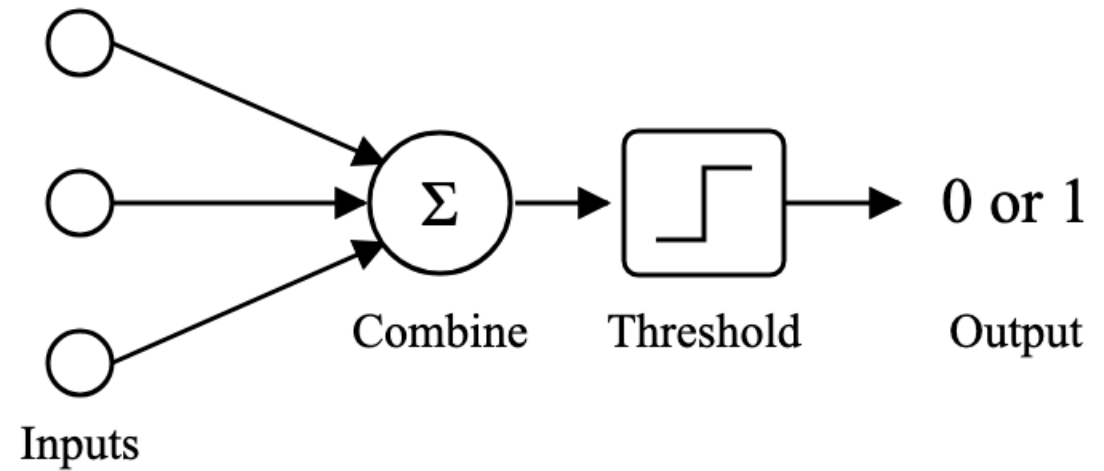
From Biological to Artificial Neurons

Biological Neuron



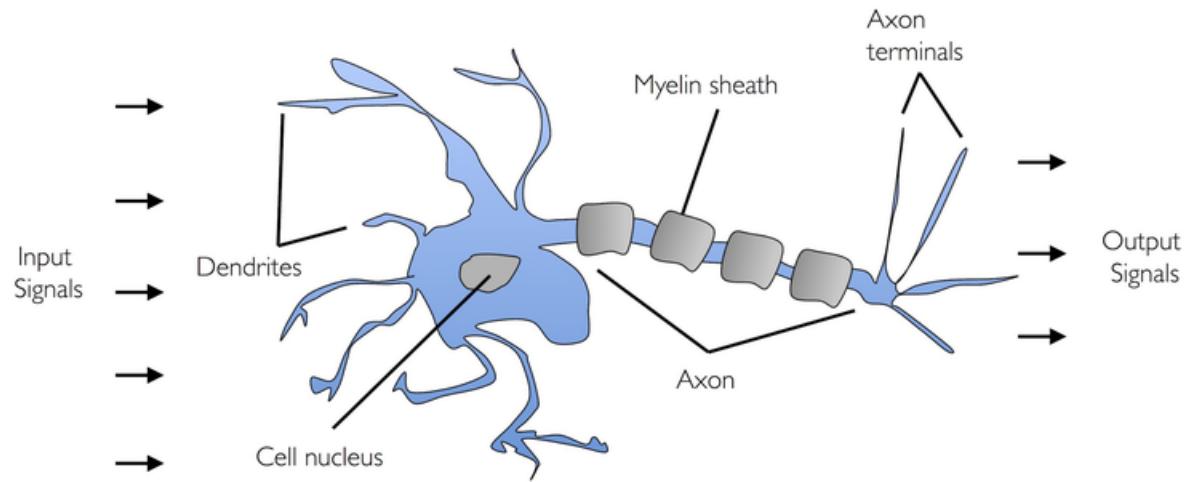
Receive → Combine → Transmit

1943 · McCulloch–Pitts



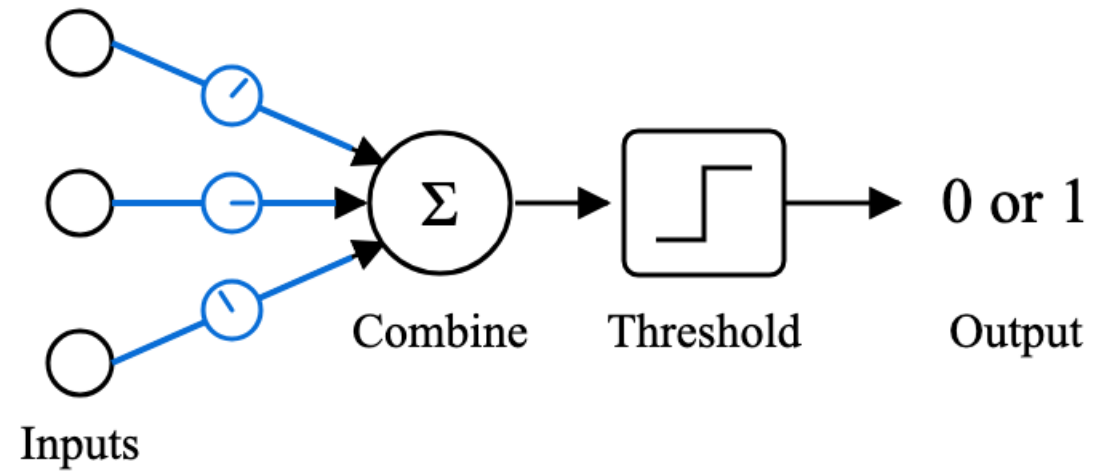
From Biological to Artificial Neurons

Biological Neuron



Receive → Combine → Transmit

1943 · McCulloch–Pitts

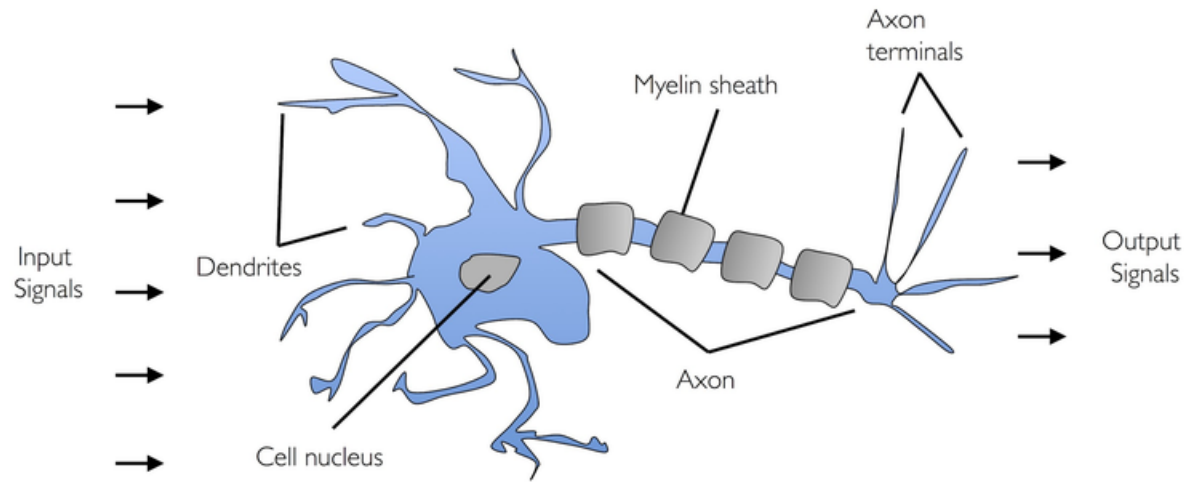


1958 · Perceptron

Learns Weights from Examples

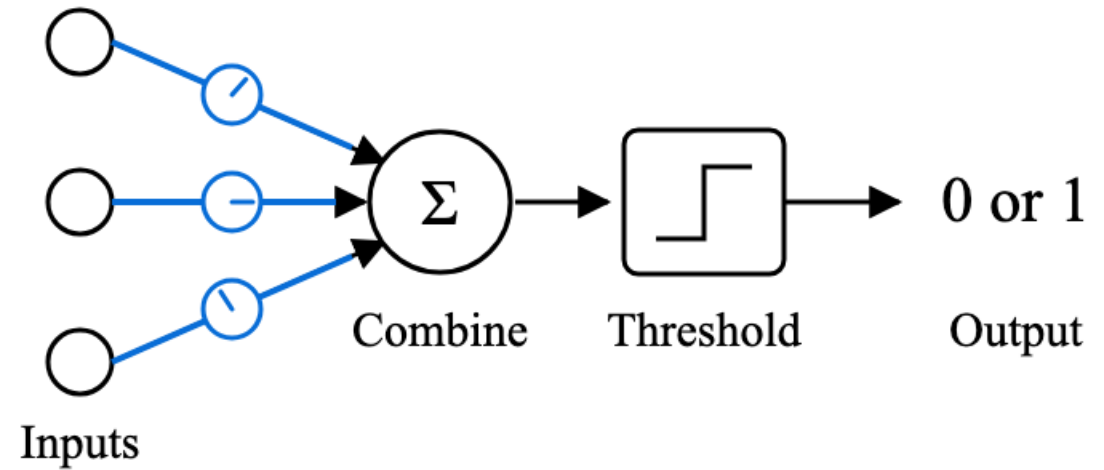
From Biological to Artificial Neurons

Biological Neuron



Receive → Combine → Transmit

1943 · McCulloch–Pitts

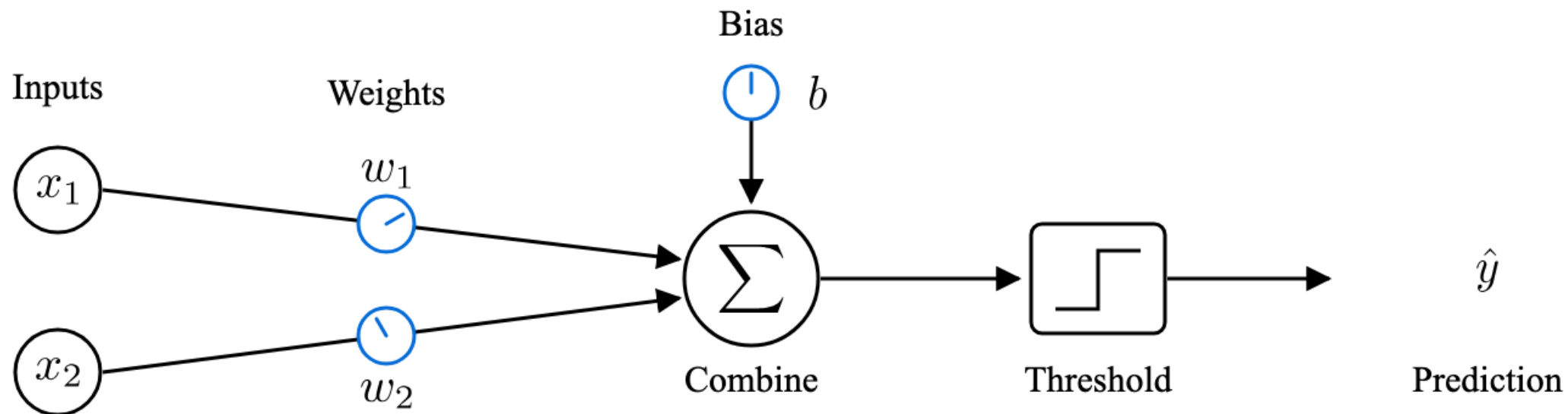


1958 · Perceptron

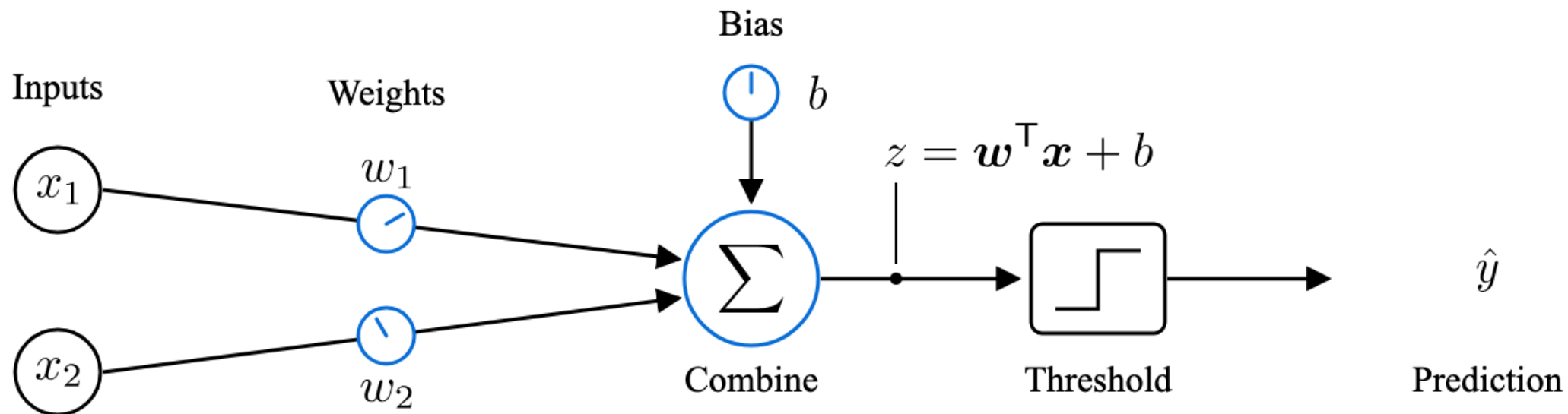
Learns Weights from Examples

Inspired by biology. Built from mathematics.

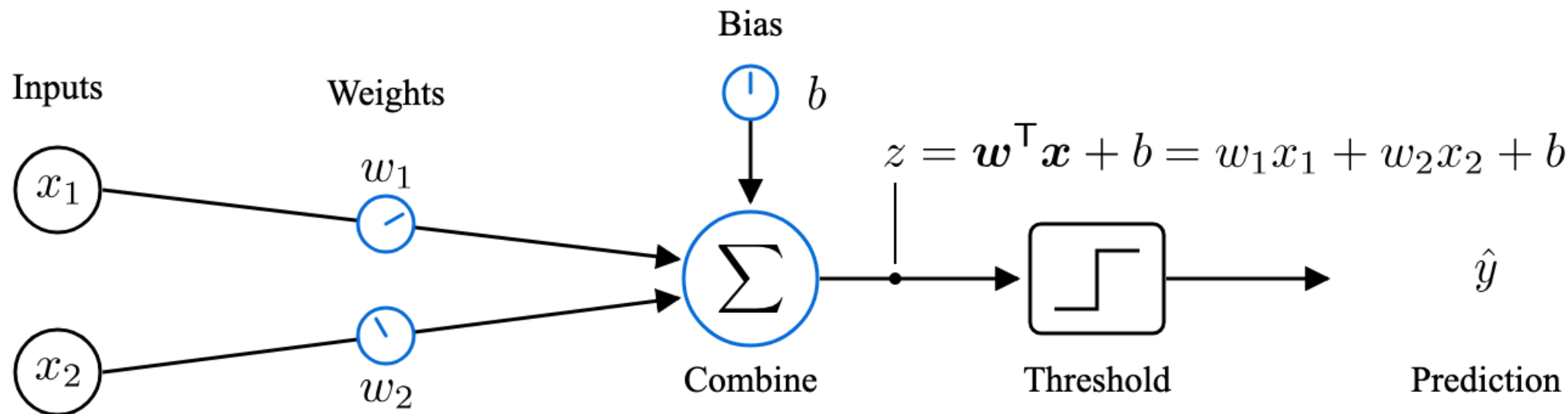
Perceptron Prediction



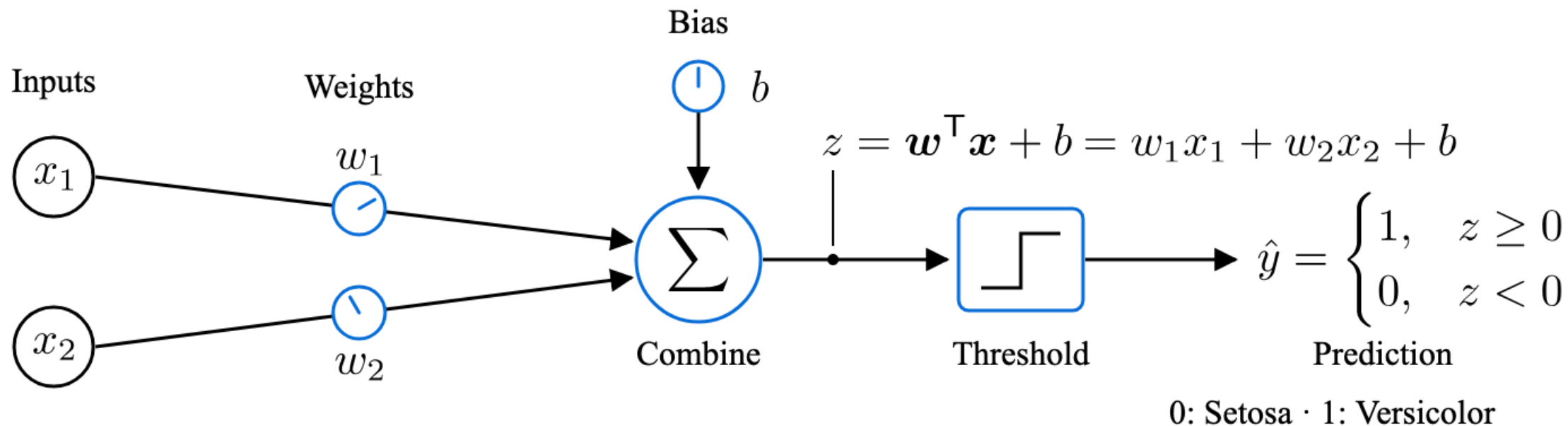
Perceptron Prediction



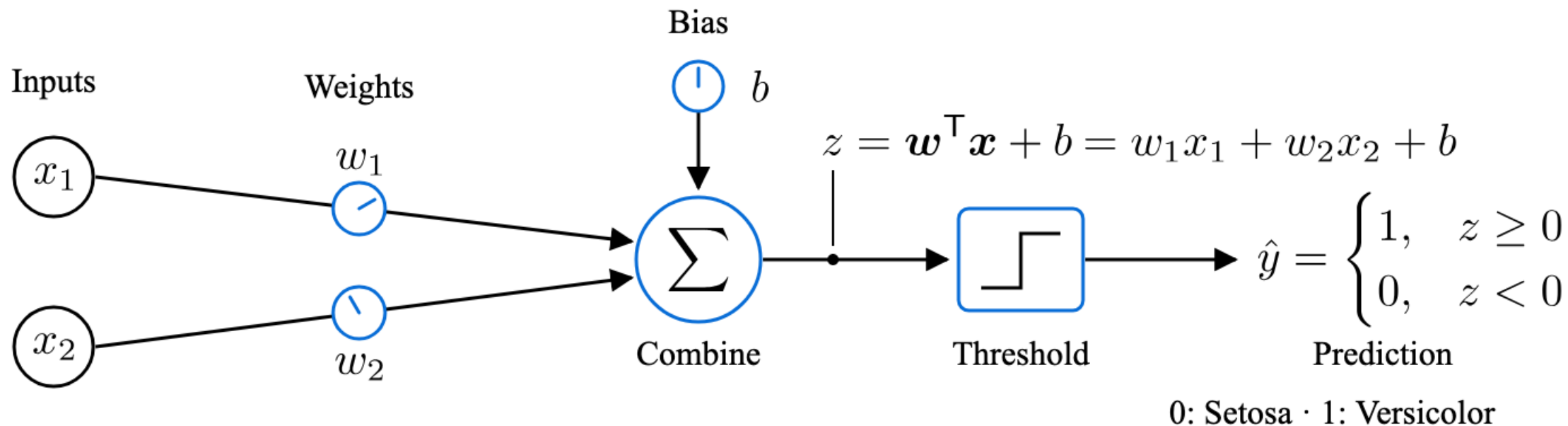
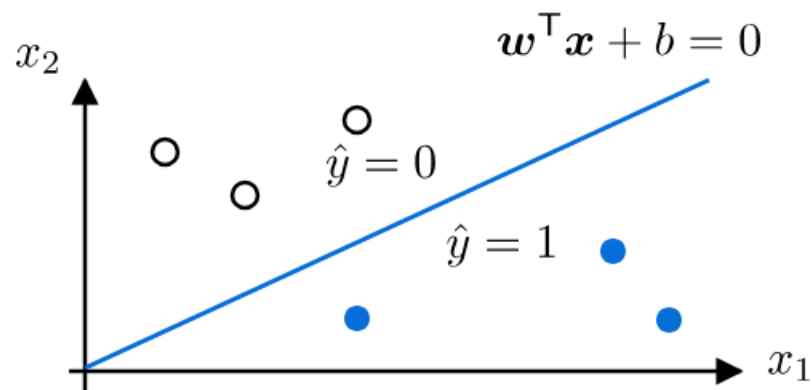
Perceptron Prediction



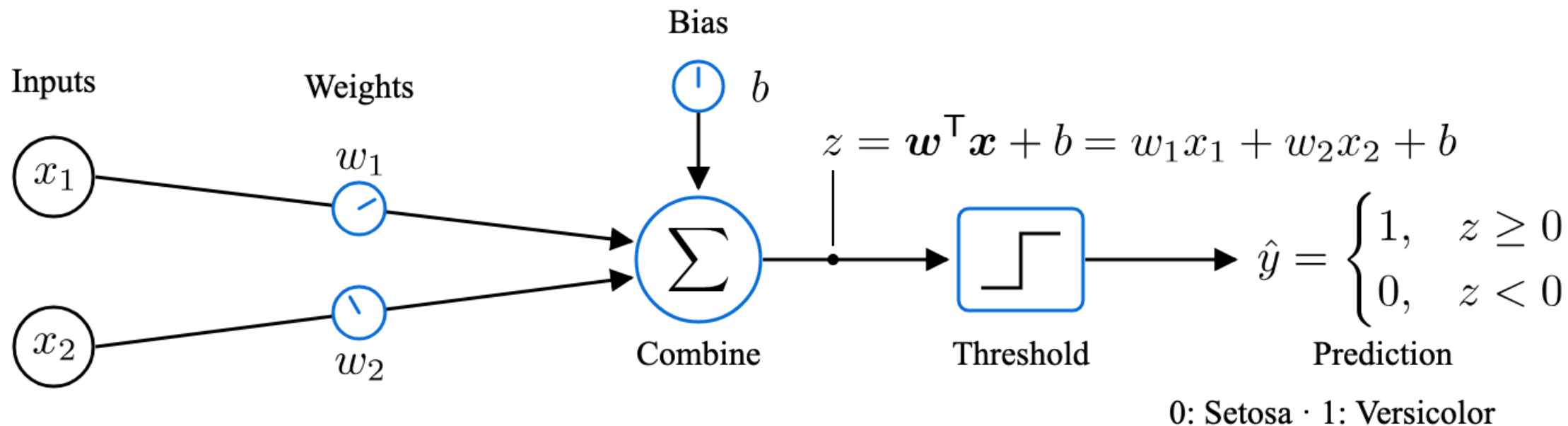
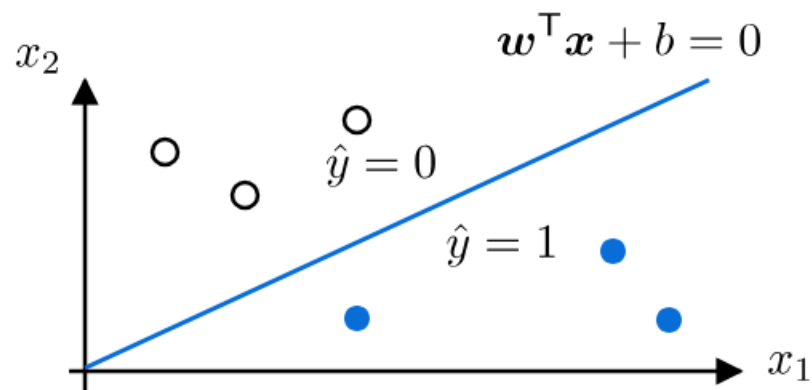
Perceptron Prediction



Perceptron Prediction

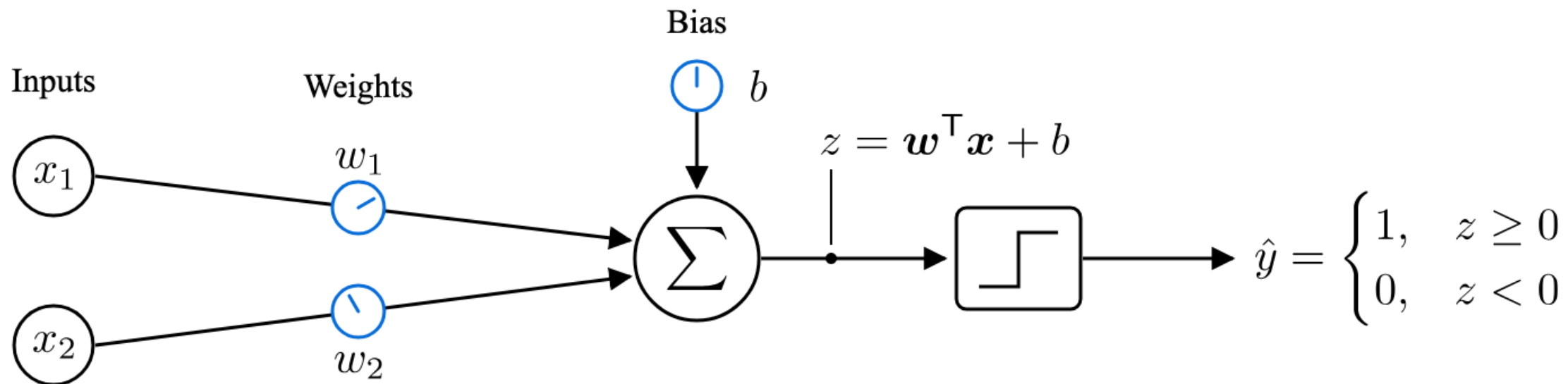


Perceptron Prediction

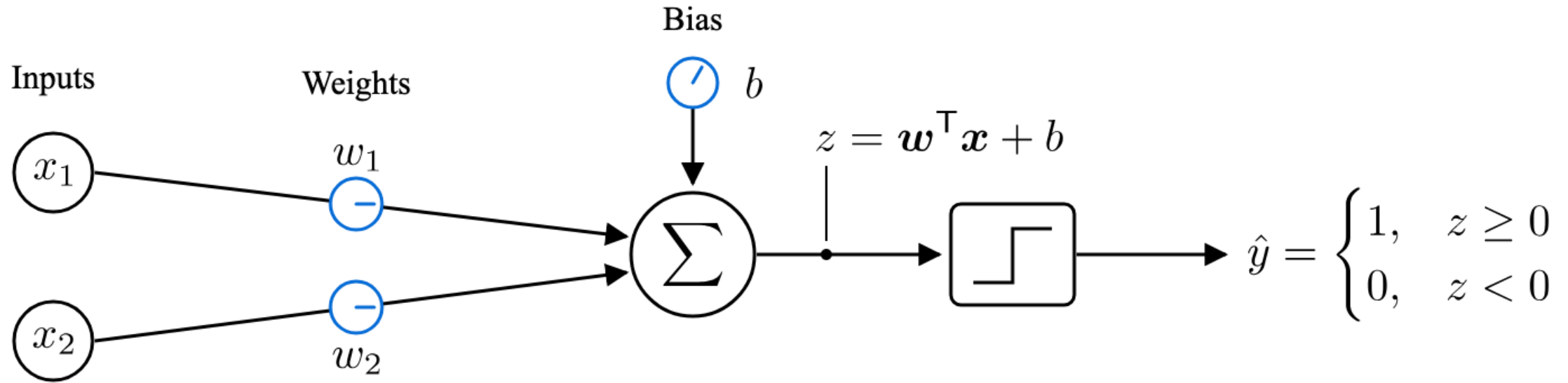


Prediction uses fixed weights and bias

Perceptron Learning



Perceptron Learning

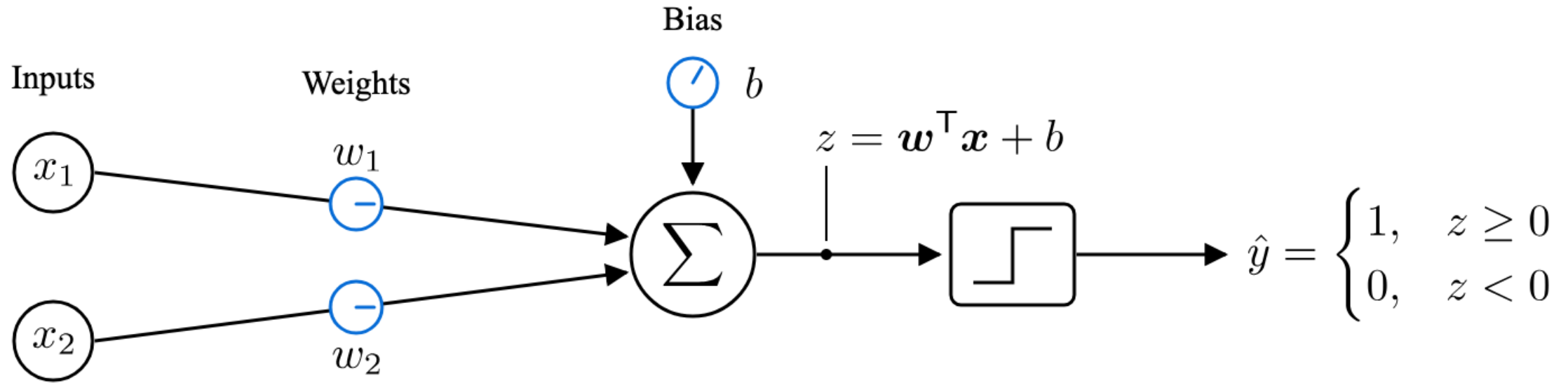


Update Rules

$$w_j \leftarrow w_j + \eta(y - \hat{y})x_j$$

$$b \leftarrow b + \eta(y - \hat{y})$$

Perceptron Learning



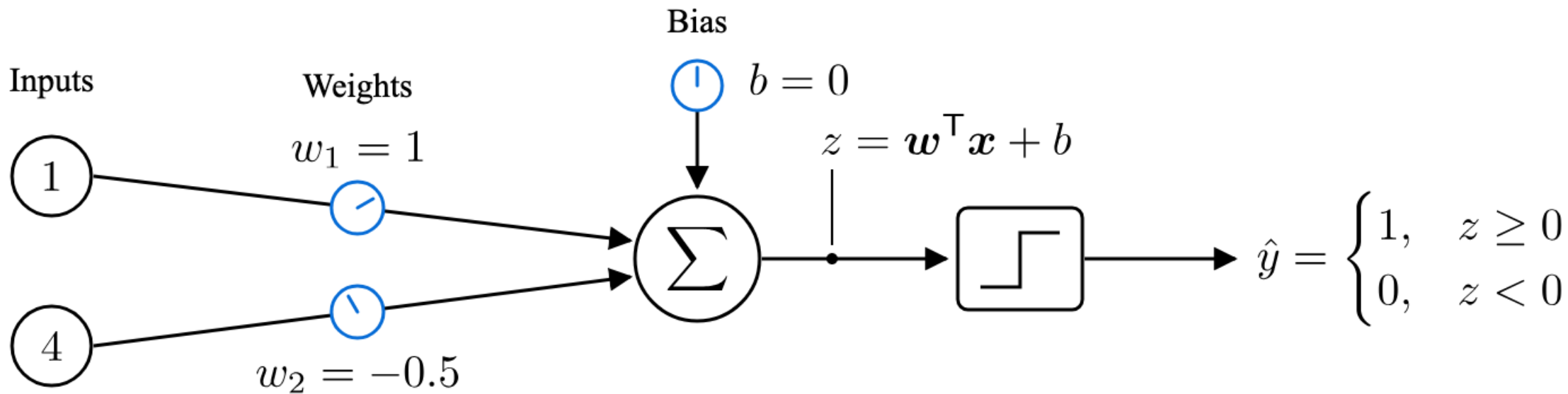
Update Rules

$$w_j \leftarrow w_j + \eta(y - \hat{y})x_j$$

$$b \leftarrow b + \eta(y - \hat{y})$$

η Learning Rate

Perceptron Learning Example



$$y = 0$$

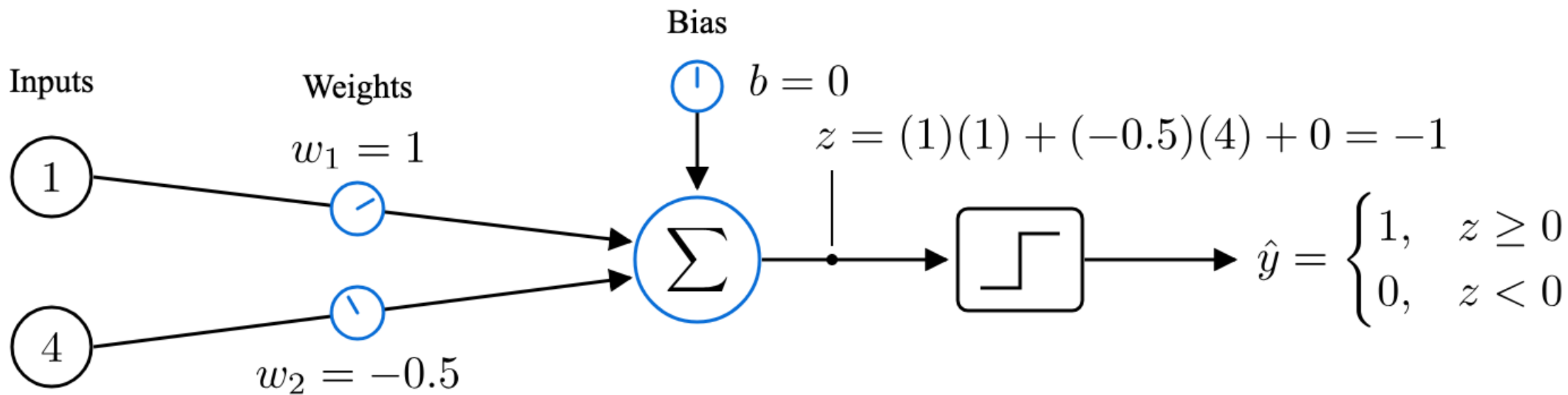
Update Rules

$$w_j \leftarrow w_j + \eta(y - \hat{y})x_j$$

$$b \leftarrow b + \eta(y - \hat{y})$$

$$\eta = 0.5 \quad \text{Learning Rate}$$

Perceptron Learning Example



$$y = 0$$

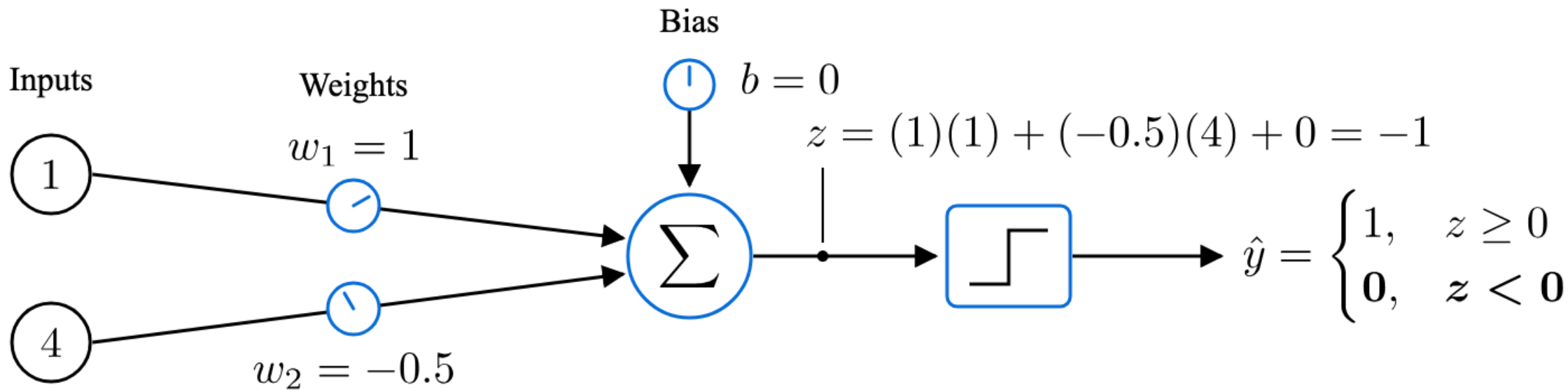
Update Rules

$$w_j \leftarrow w_j + \eta(y - \hat{y})x_j$$

$$b \leftarrow b + \eta(y - \hat{y})$$

$$\eta = 0.5 \quad \text{Learning Rate}$$

Perceptron Learning Example



$$y = 0$$

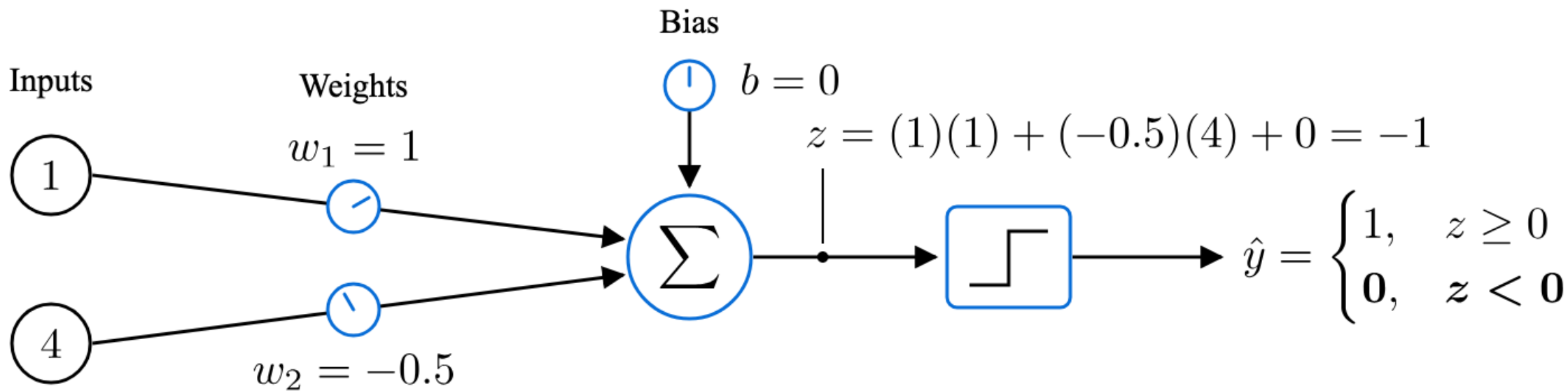
Update Rules

$$w_j \leftarrow w_j + \eta(y - \hat{y})x_j$$

$$b \leftarrow b + \eta(y - \hat{y})$$

$$\eta = 0.5 \quad \text{Learning Rate}$$

Perceptron Learning Example



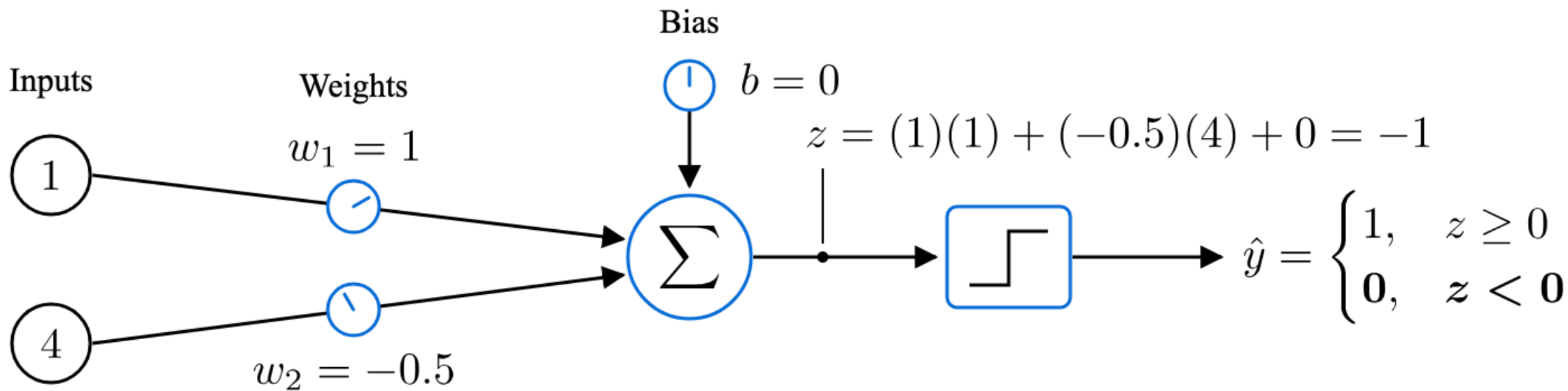
Update Rules

$$w_j \leftarrow w_j + 0.5(0)x_j = w_j$$

$$b \leftarrow b + 0.5(0) = b$$

$$\eta = 0.5 \quad \text{Learning Rate}$$

Perceptron Learning Example



$$y = 0$$

Correct Prediction $\rightarrow$ No Update

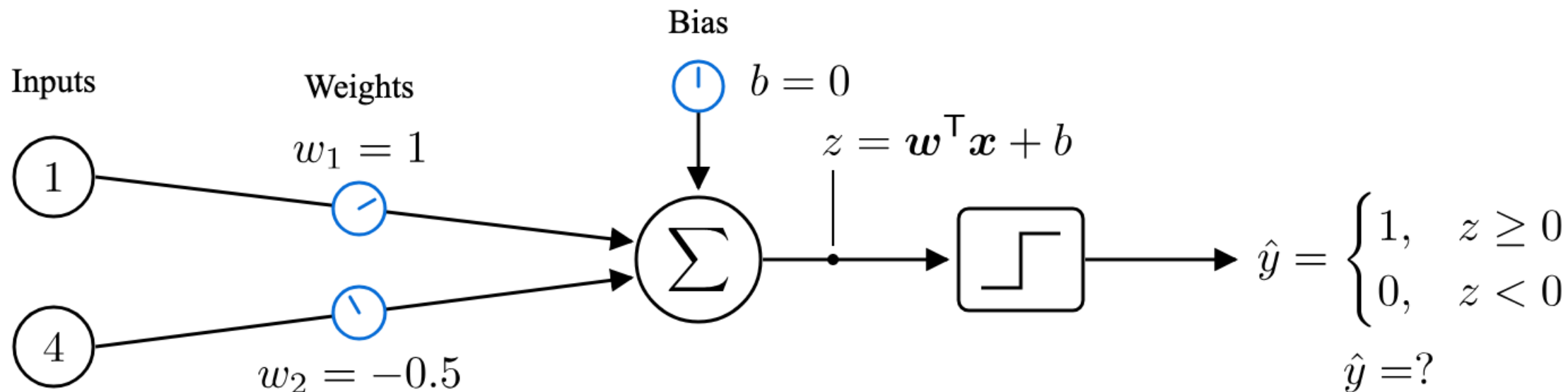
Update Rules

$$w_j \leftarrow w_j + 0.5(0)x_j = w_j$$

$$b \leftarrow b + 0.5(0) = b$$

$$\eta = 0.5 \quad \text{Learning Rate}$$

Perceptron Learning Exercise



Your Turn

$$y = 1$$

1. Predict the class
 2. Update both weights and the bias
 3. Predict again
- One Submission per Group on Ed

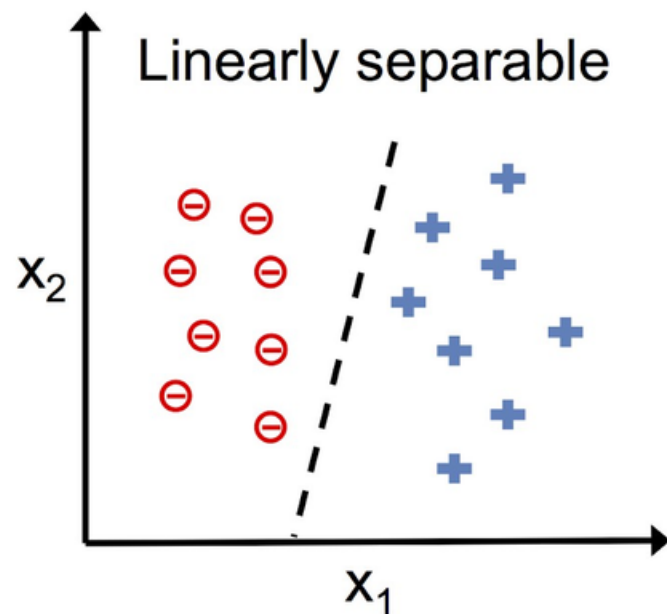
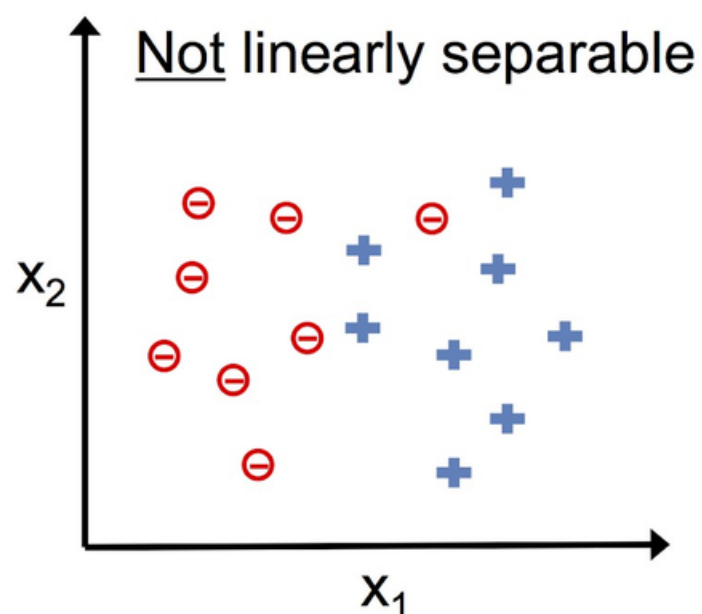
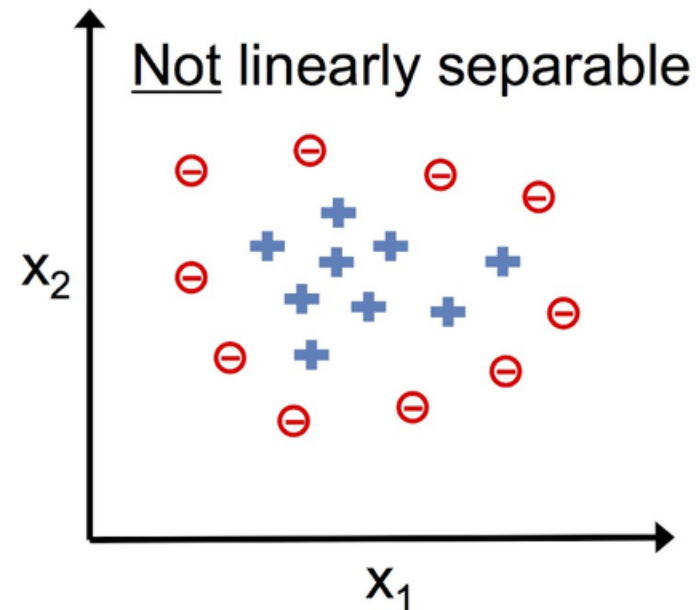
Update Rules

$$w_j \leftarrow w_j + \eta(y - \hat{y})x_j$$

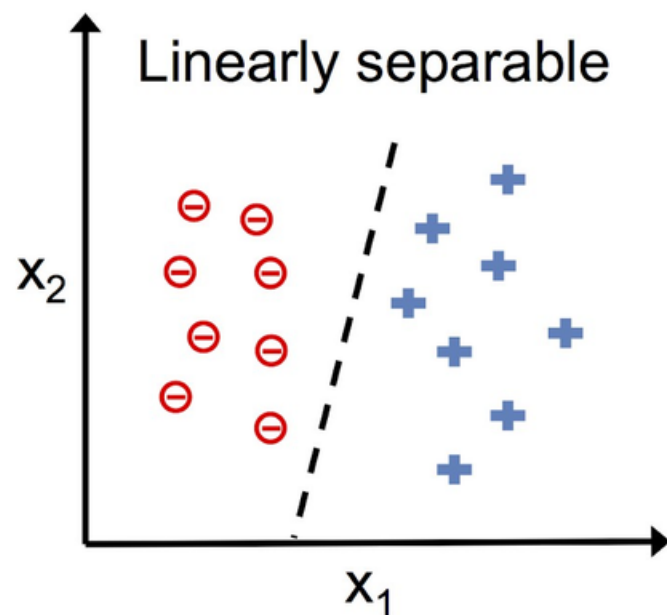
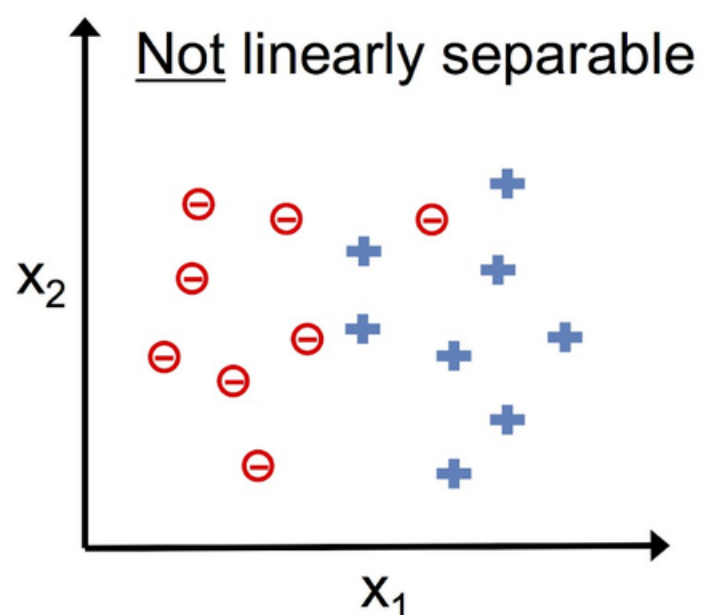
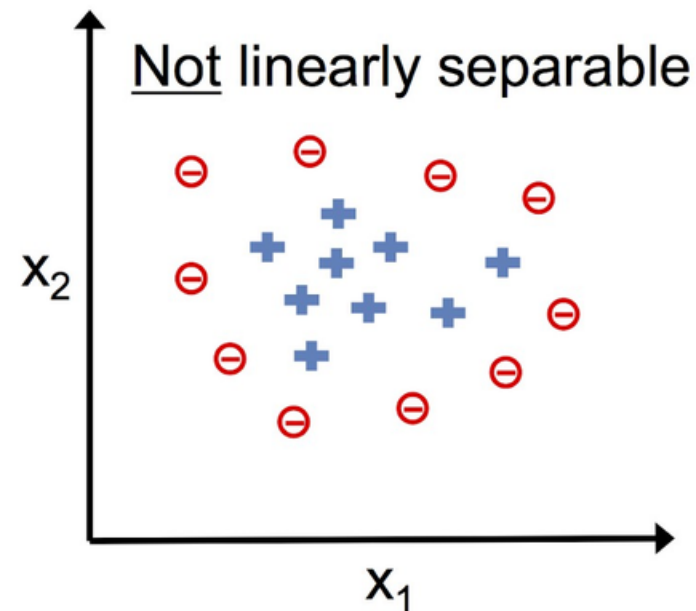
$$b \leftarrow b + \eta(y - \hat{y})$$

$$\eta = 0.5 \quad \text{Learning Rate}$$

Linearly Separable vs. Not Linearly Separable



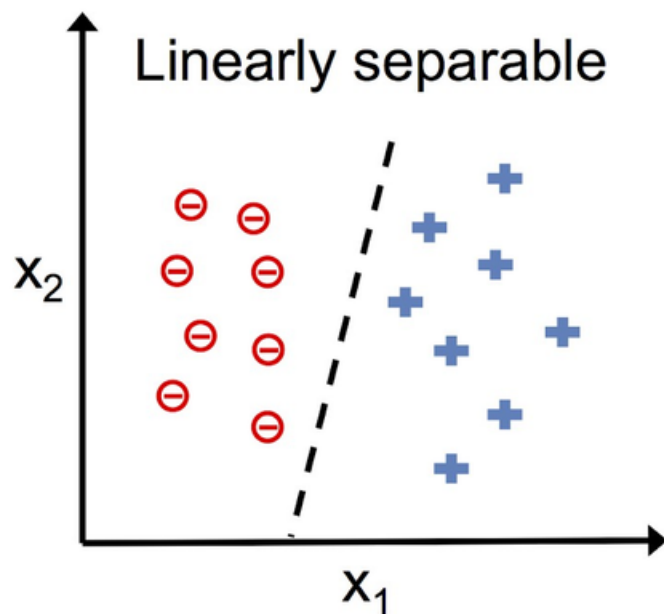
Linearly Separable vs. Not Linearly Separable



Perceptron Converges

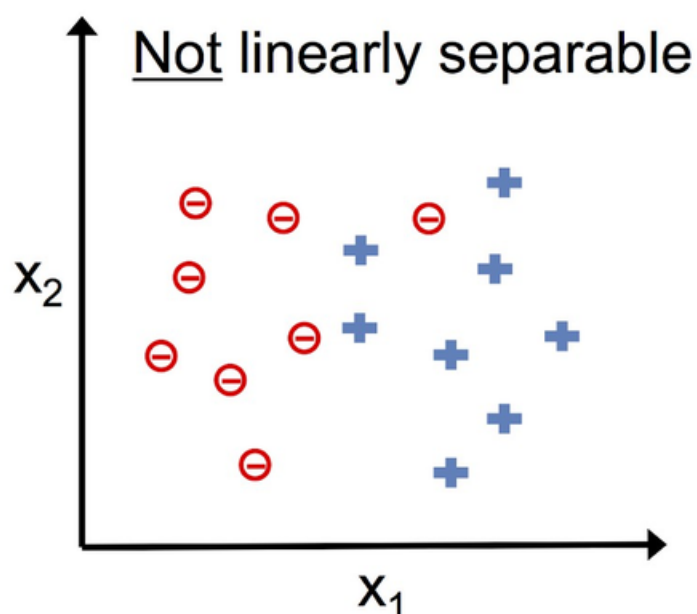
Fixed Positive Learning Rate

Linearly Separable vs. Not Linearly Separable



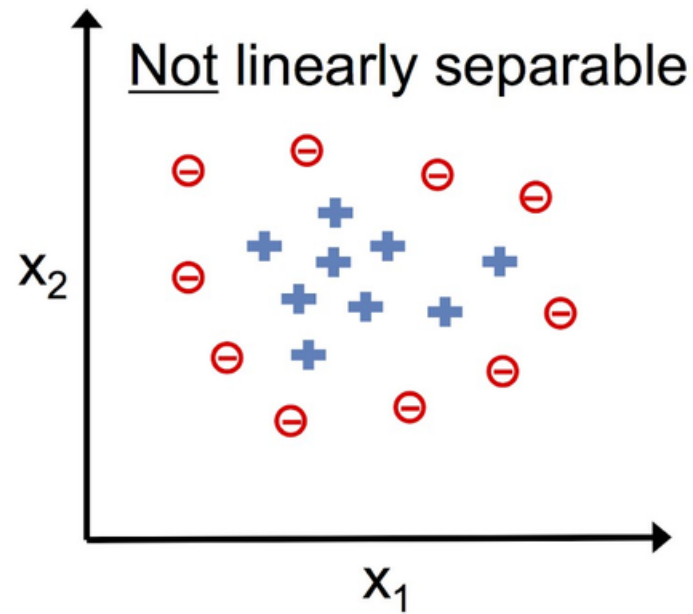
Perceptron Converges

Fixed Positive Learning Rate



Updates Can Continue Indefinitely

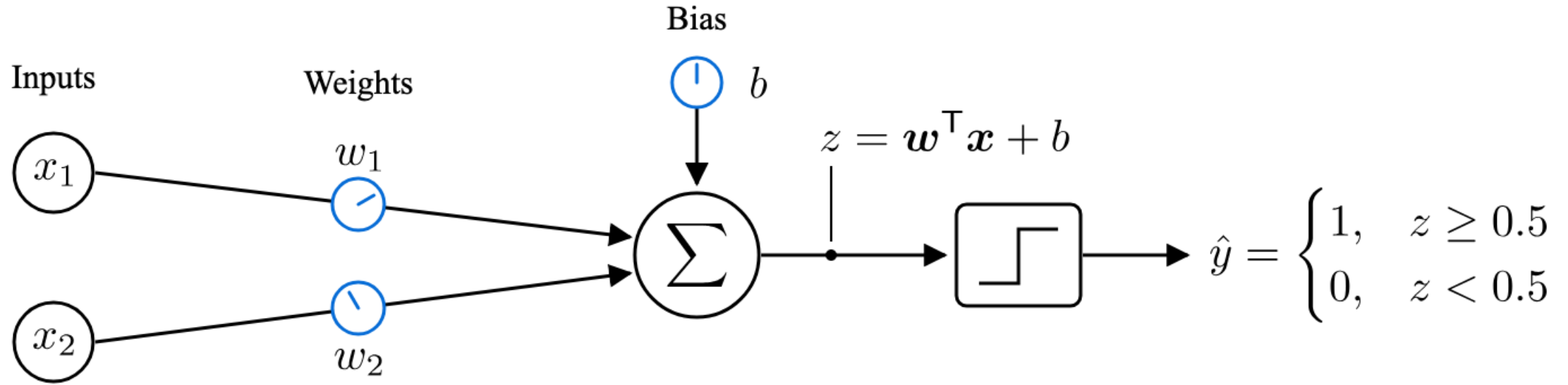
Limit the Number of Epochs



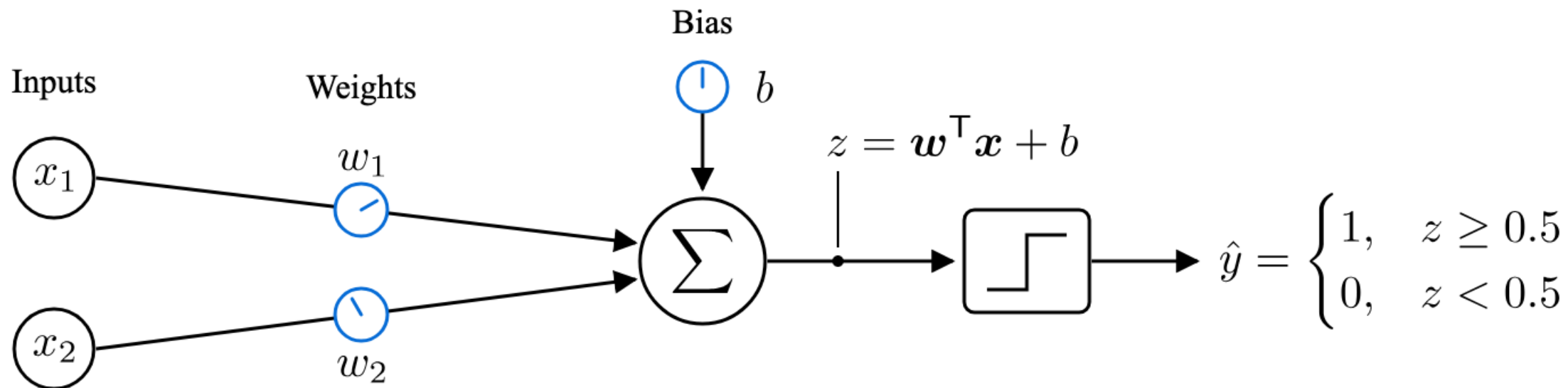
maib7002.dirk.hk/L2.zip

Demo 4: perceptron.ipynb

Adaptive Linear Neuron



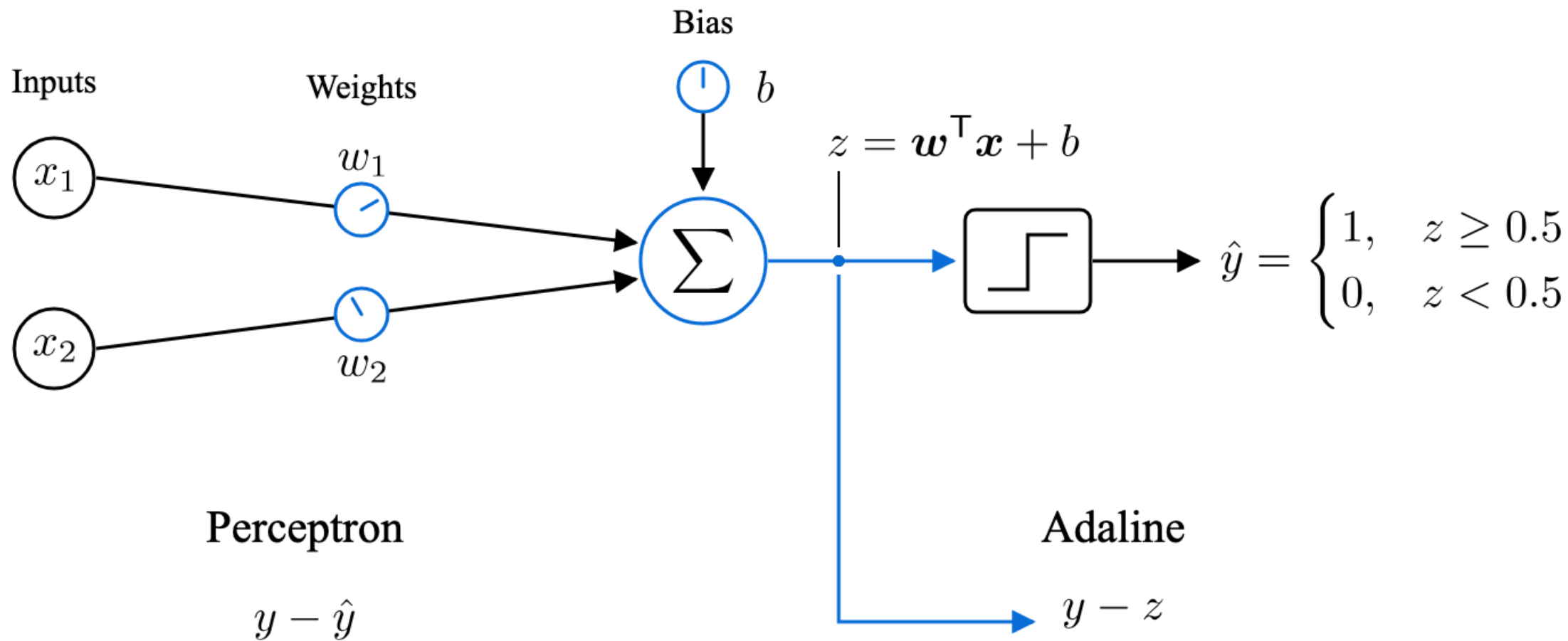
Adaptive Linear Neuron



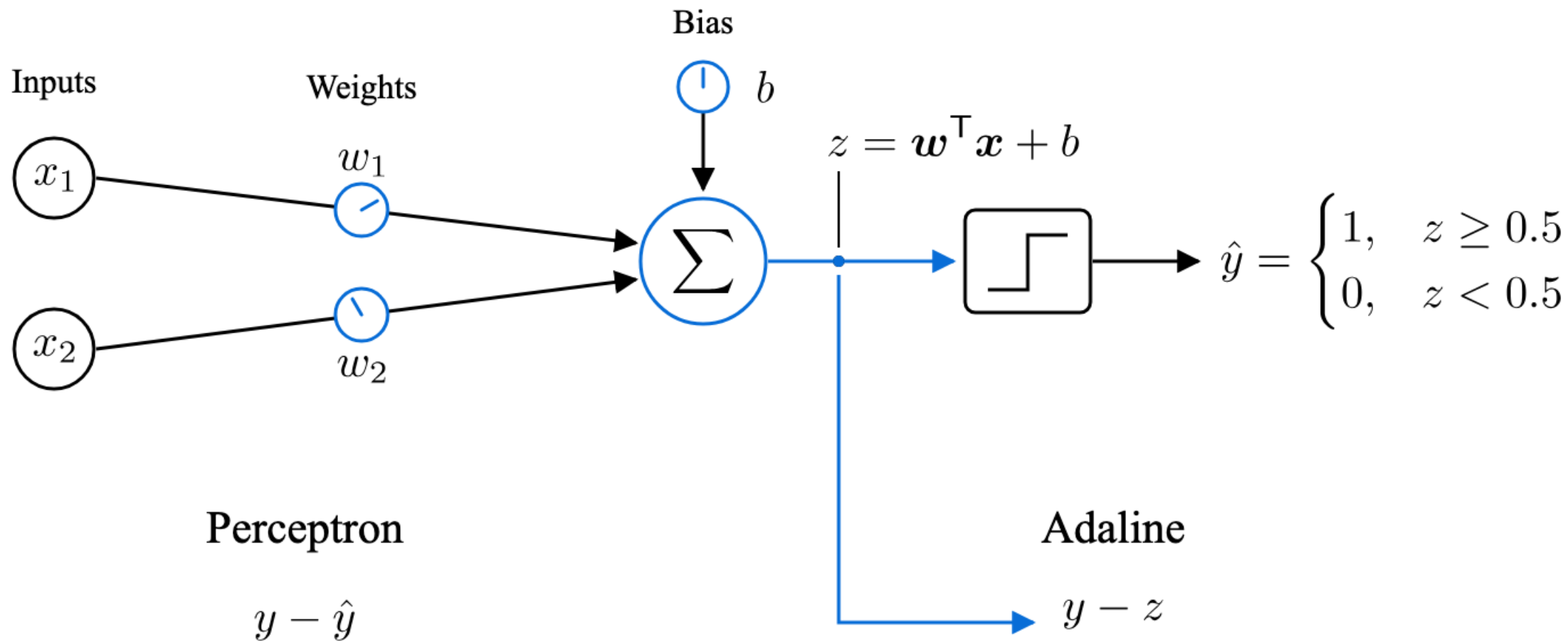
Perceptron

$$y - \hat{y}$$

Adaptive Linear Neuron



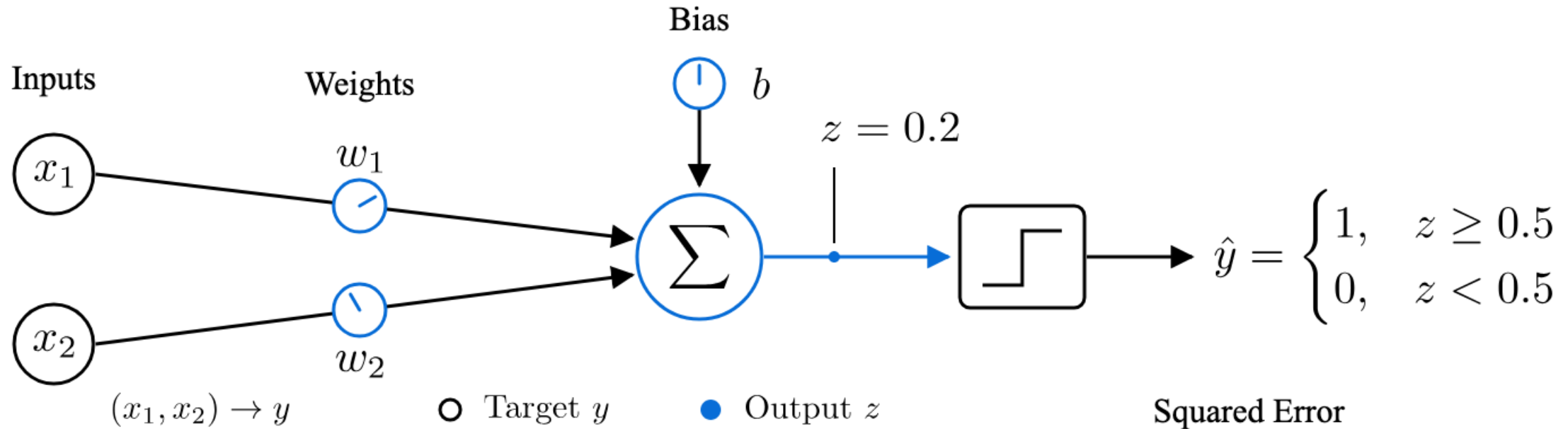
Adaptive Linear Neuron



Learning Uses the Continuous Output

Bernard Widrow and Ted Hoff · 1960

Adaline Learning: Measure the Error

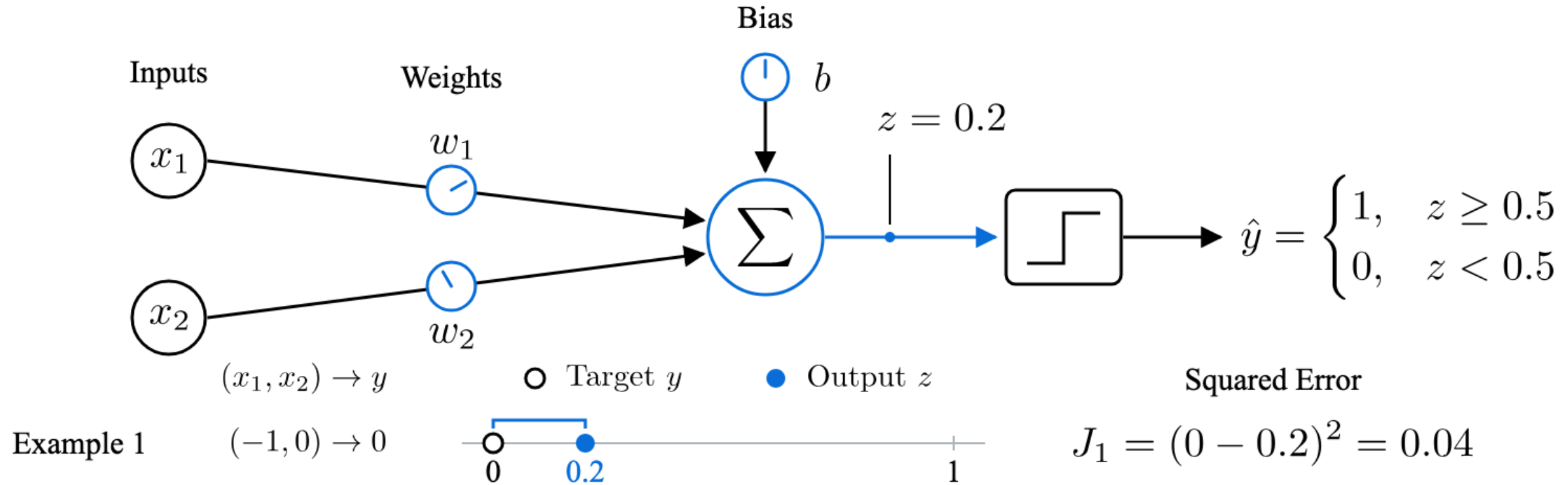


Example 1

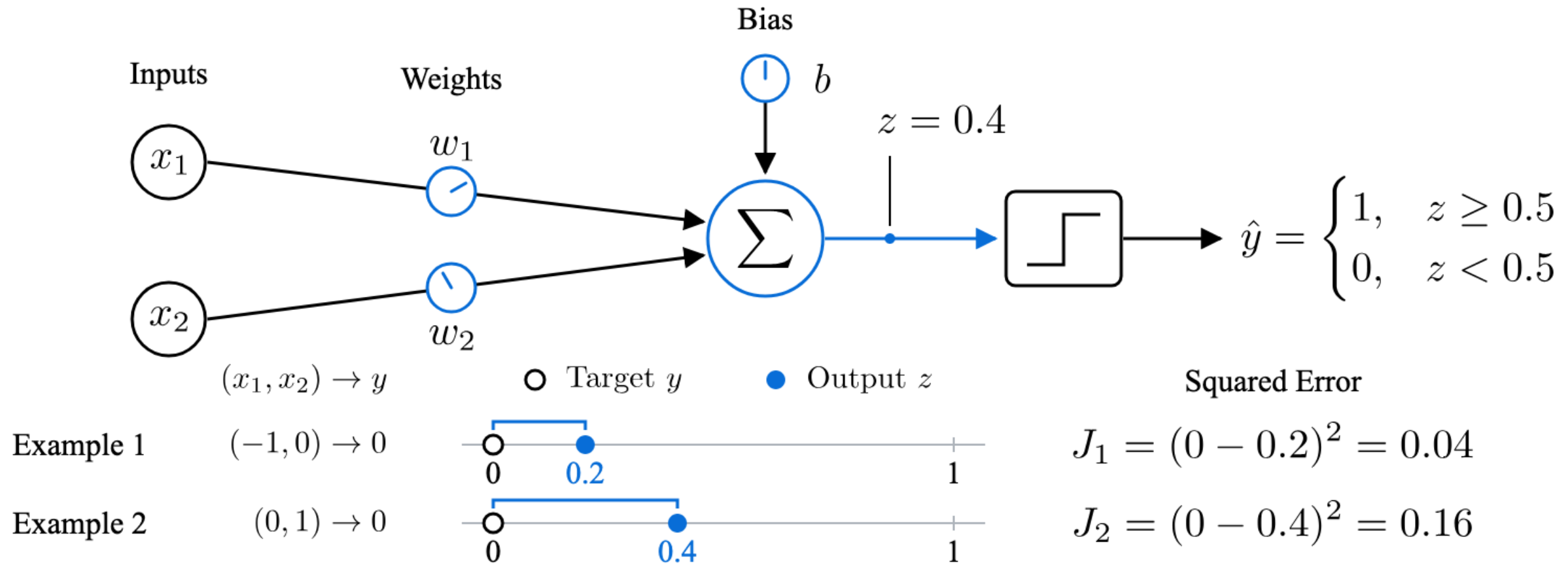
$(-1, 0) \rightarrow 0$



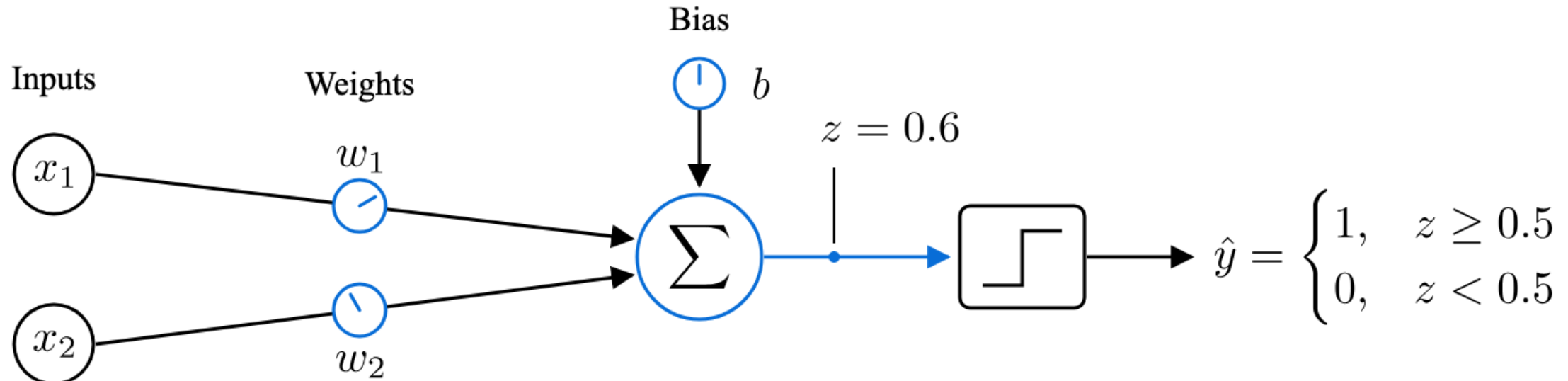
Adaline Learning: Measure the Error



Adaline Learning: Measure the Error

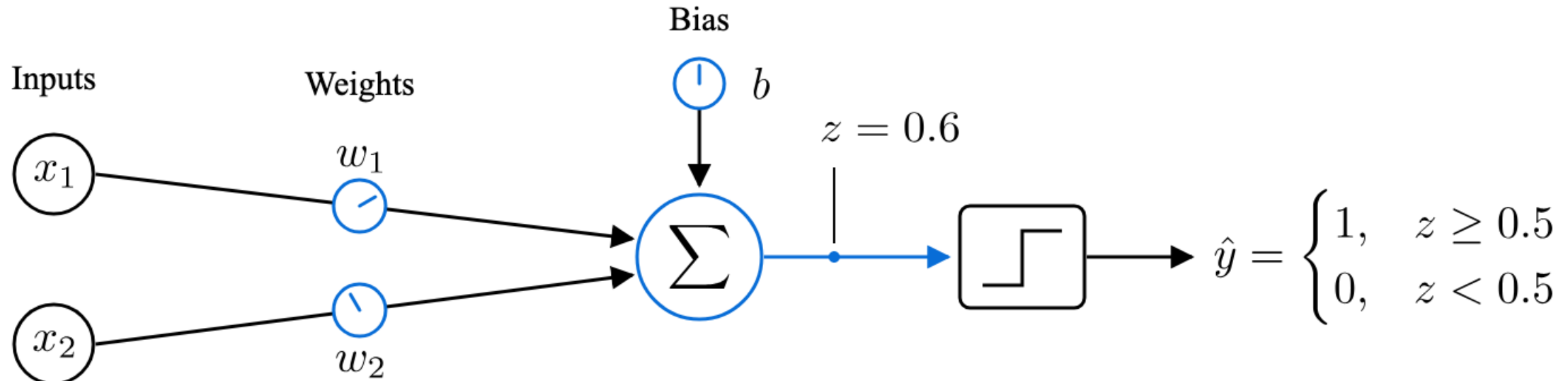


Adaline Learning: Measure the Error



	$(x_1, x_2) \rightarrow y$	Target y Output z	Squared Error
Example 1	$(-1, 0) \rightarrow 0$		$J_1 = (0 - 0.2)^2 = 0.04$
Example 2	$(0, 1) \rightarrow 0$		$J_2 = (0 - 0.4)^2 = 0.16$
Example 3	$(1, 0) \rightarrow 1$		$J_3 = (1 - 0.6)^2 = 0.16$

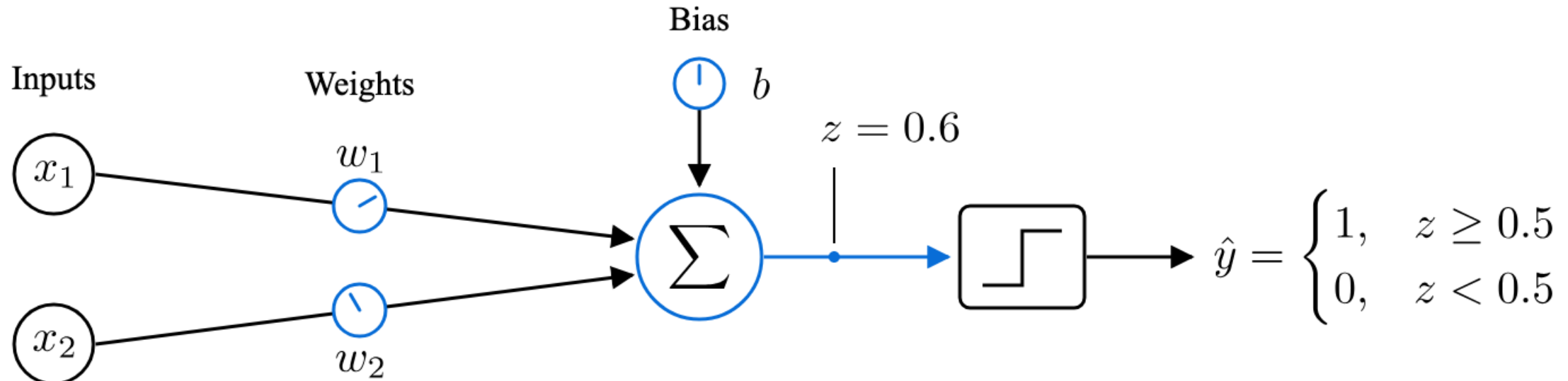
Adaline Learning: Measure the Error



	$(x_1, x_2) \rightarrow y$	○ Target y ● Output z	Squared Error
Example 1	$(-1, 0) \rightarrow 0$		$J_1 = (0 - 0.2)^2 = 0.04$
Example 2	$(0, 1) \rightarrow 0$		$J_2 = (0 - 0.4)^2 = 0.16$
Example 3	$(1, 0) \rightarrow 1$		$J_3 = (1 - 0.6)^2 = 0.16$

$$J = \frac{0.04 + 0.16 + 0.16}{3} = 0.12$$

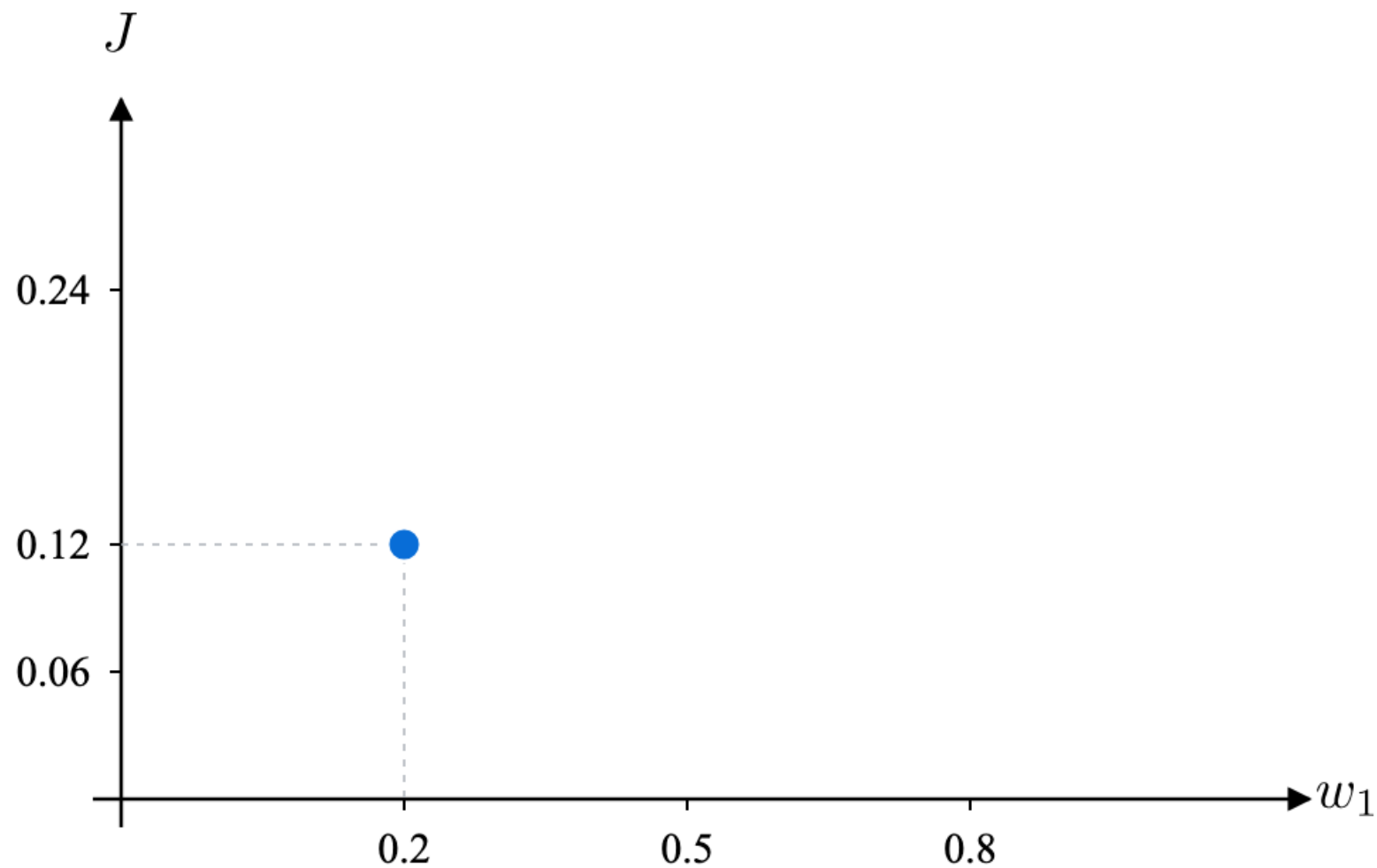
Adaline Learning: Measure the Error



	$(x_1, x_2) \rightarrow y$	○ Target y ● Output z	Squared Error
Example 1	$(-1, 0) \rightarrow 0$		$J_1 = (0 - 0.2)^2 = 0.04$
Example 2	$(0, 1) \rightarrow 0$		$J_2 = (0 - 0.4)^2 = 0.16$
Example 3	$(1, 0) \rightarrow 1$		$J_3 = (1 - 0.6)^2 = 0.16$

$$J = \frac{1}{n} \sum_{i=1}^n J_i = \frac{1}{n} \sum_{i=1}^n \left(y^{(i)} - z^{(i)} \right)^2$$

Gradient Descent: Move Downhill



Other Parameters Fixed

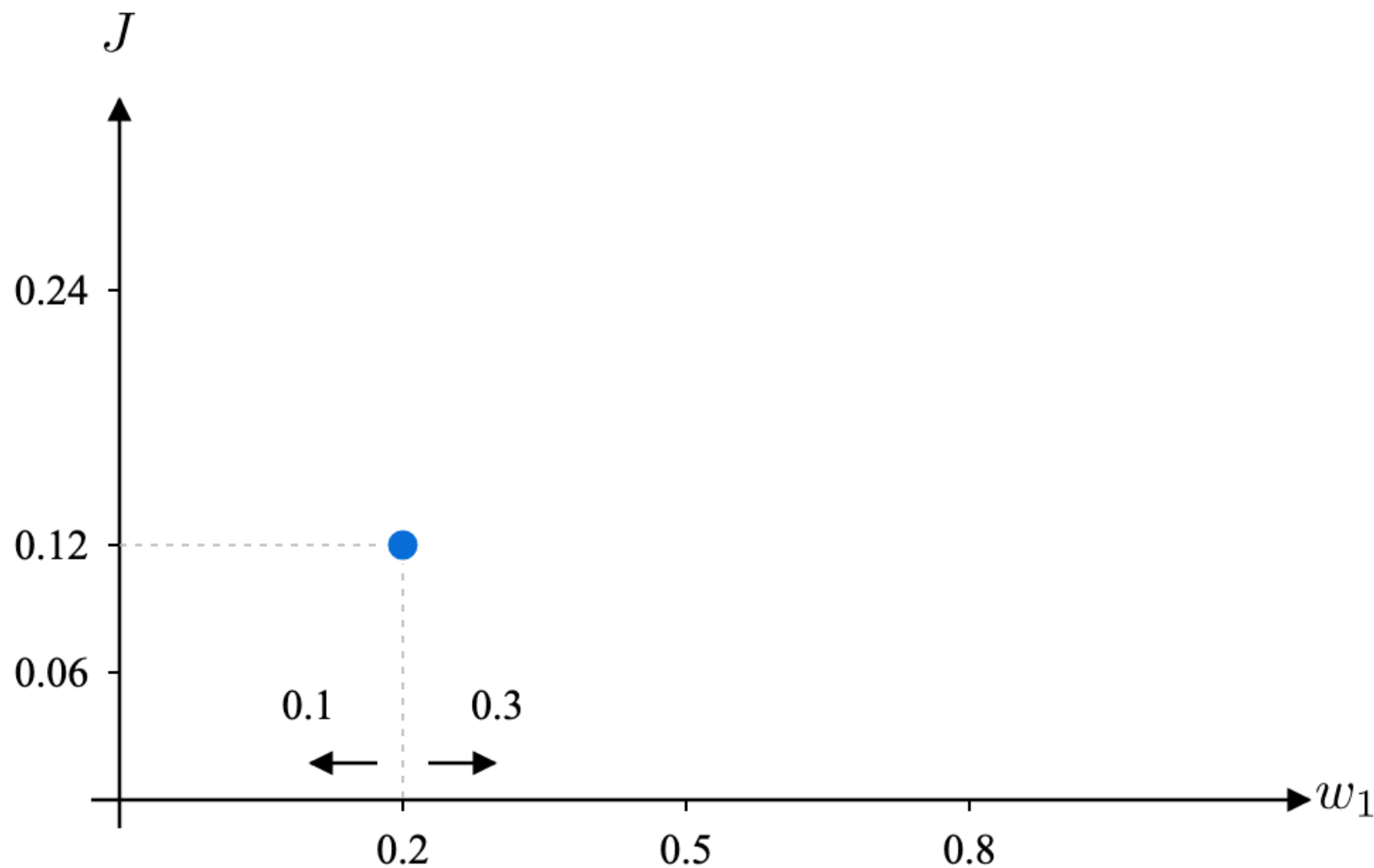
$$w_2 = 0, \quad b = 0.4$$

$$w_1 = 0.200$$



$$J = 0.120$$

Gradient Descent: Move Downhill



Other Parameters Fixed

$$w_2 = 0, \quad b = 0.4$$

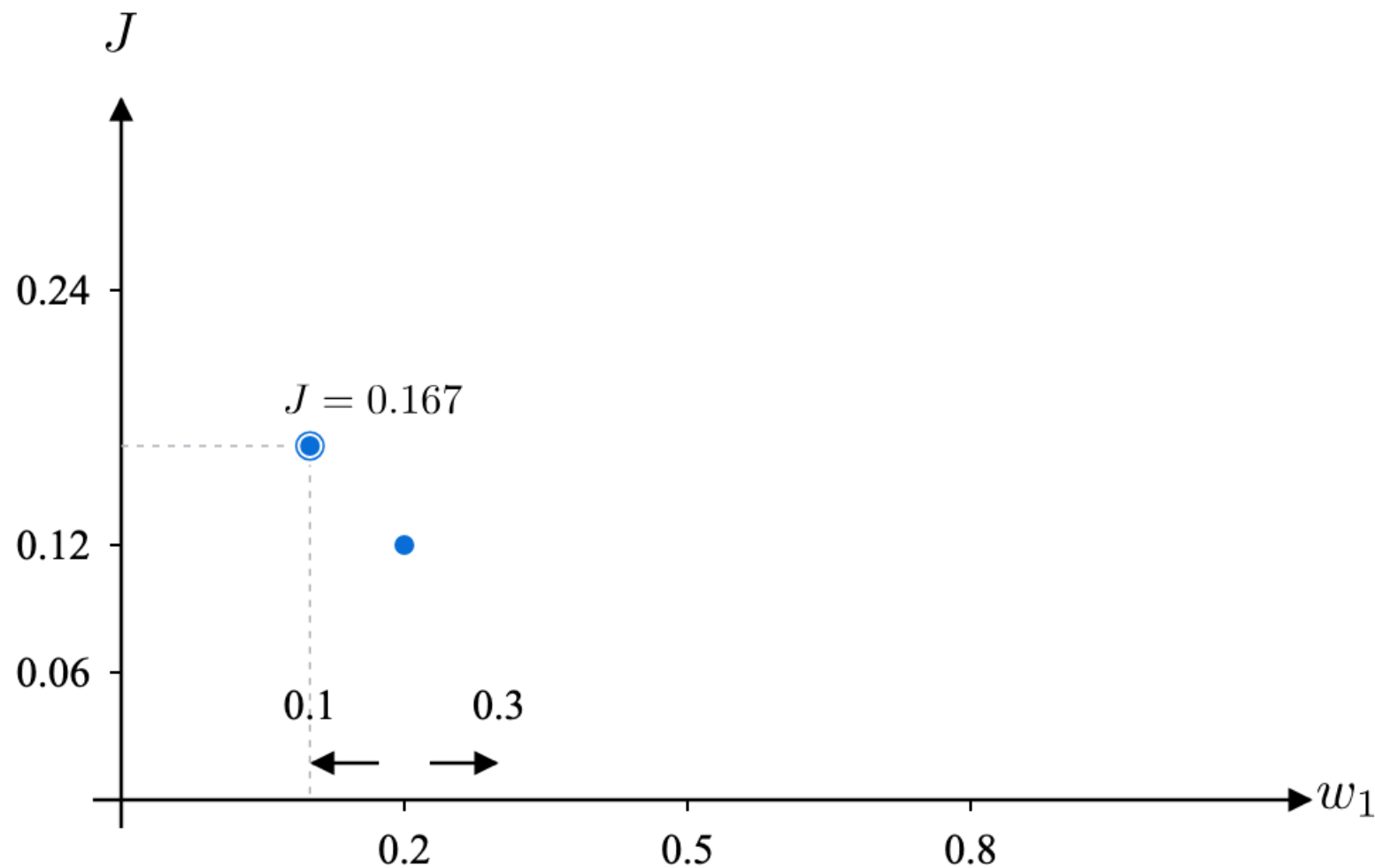
$$w_1 = 0.200$$



$$J = 0.120$$

Which Value Should We Try Next?

Gradient Descent: Move Downhill



Other Parameters Fixed

$$w_2 = 0, \quad b = 0.4$$

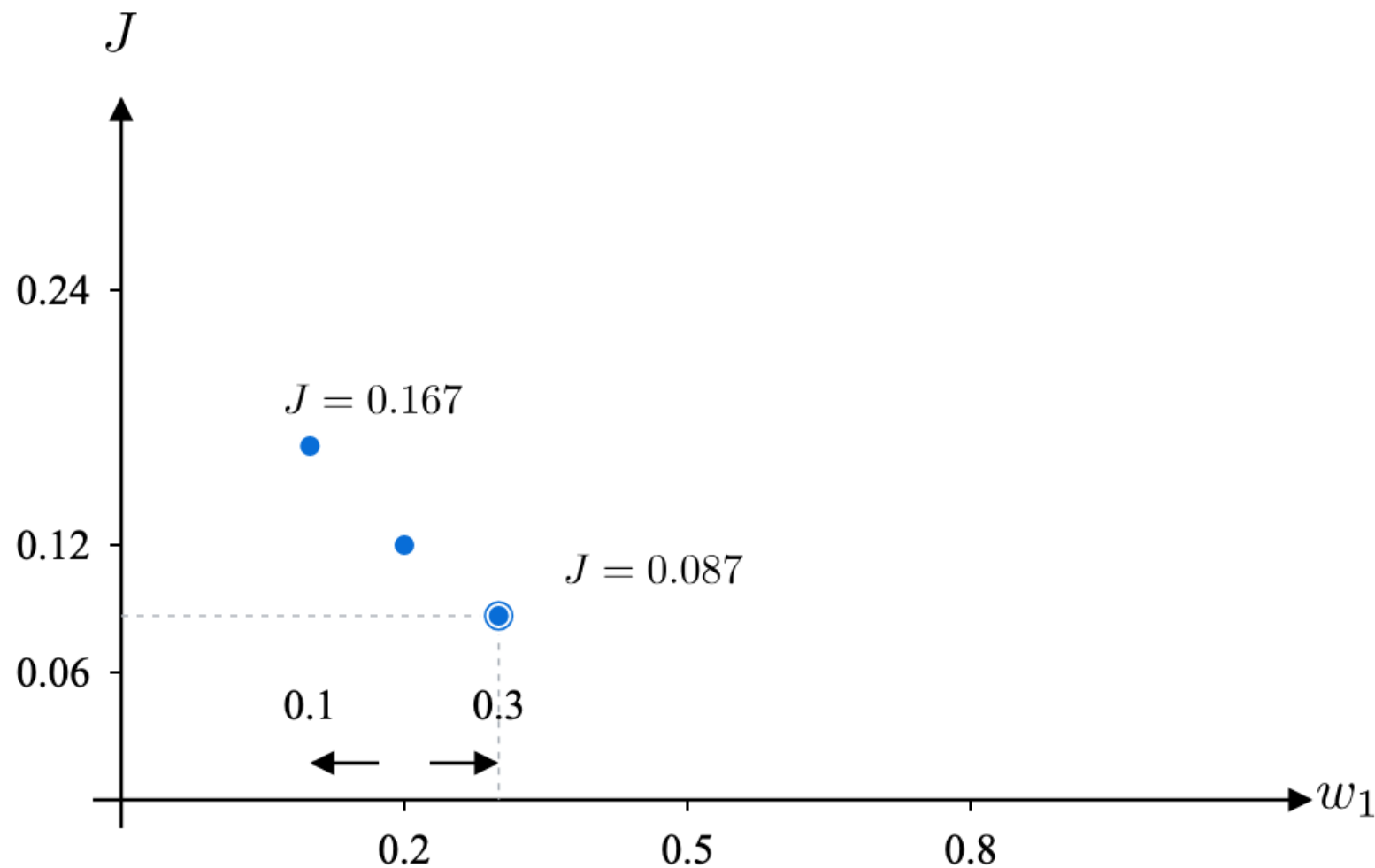
$$w_1 = 0.100$$



$$J = 0.167$$

Which Value Should We Try Next?

Gradient Descent: Move Downhill



Other Parameters Fixed

$$w_2 = 0, \quad b = 0.4$$

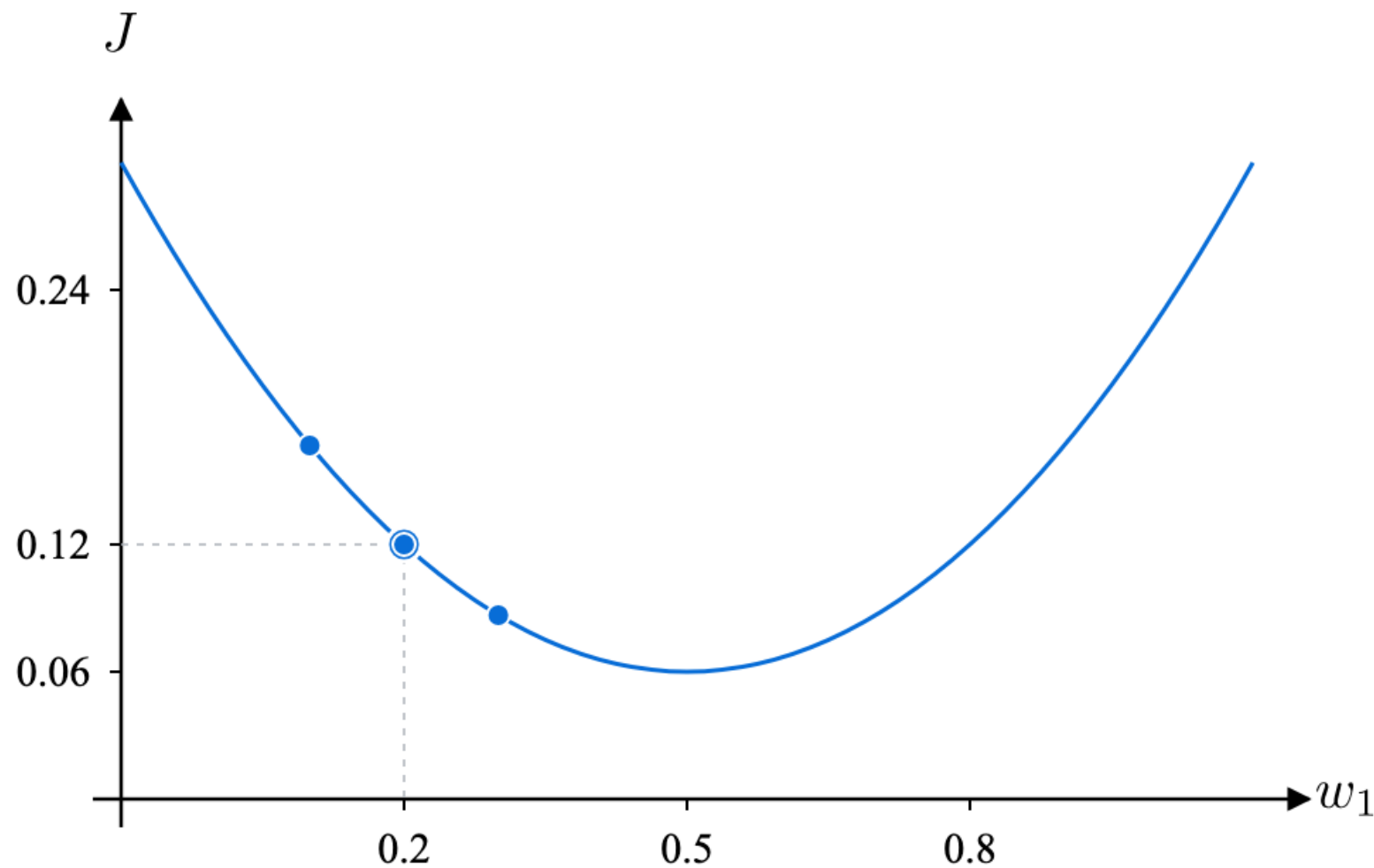
$$w_1 = 0.300$$



$$J = 0.087$$

Which Value Should We Try Next?

Gradient Descent: Move Downhill



Other Parameters Fixed

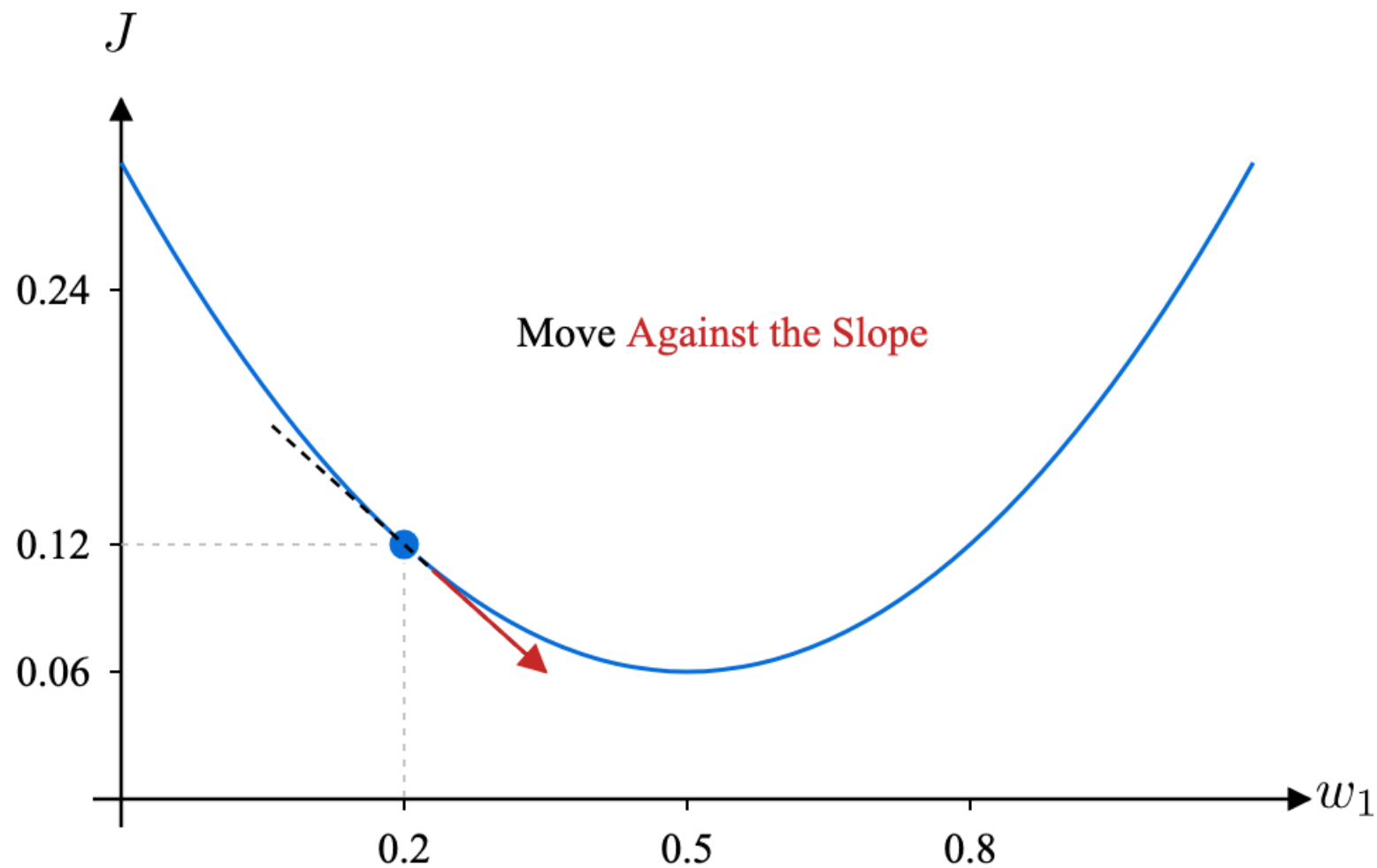
$$w_2 = 0, \quad b = 0.4$$

$$w_1 = 0.200$$



$$J = 0.120$$

Gradient Descent: Move Downhill



Other Parameters Fixed

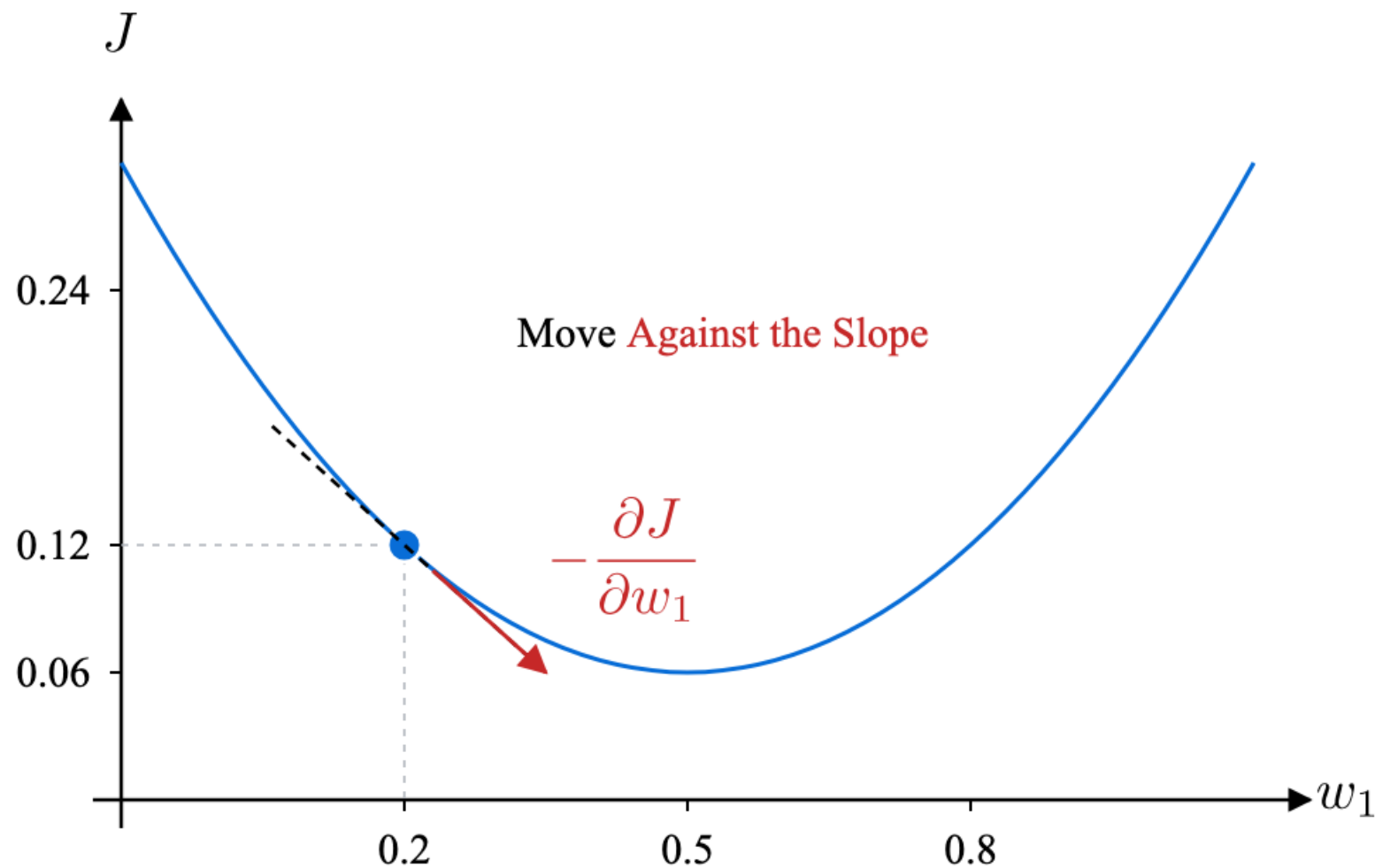
$$w_2 = 0, \quad b = 0.4$$

$$w_1 = 0.200$$



$$J = 0.120$$

Gradient Descent: Move Downhill



Other Parameters Fixed

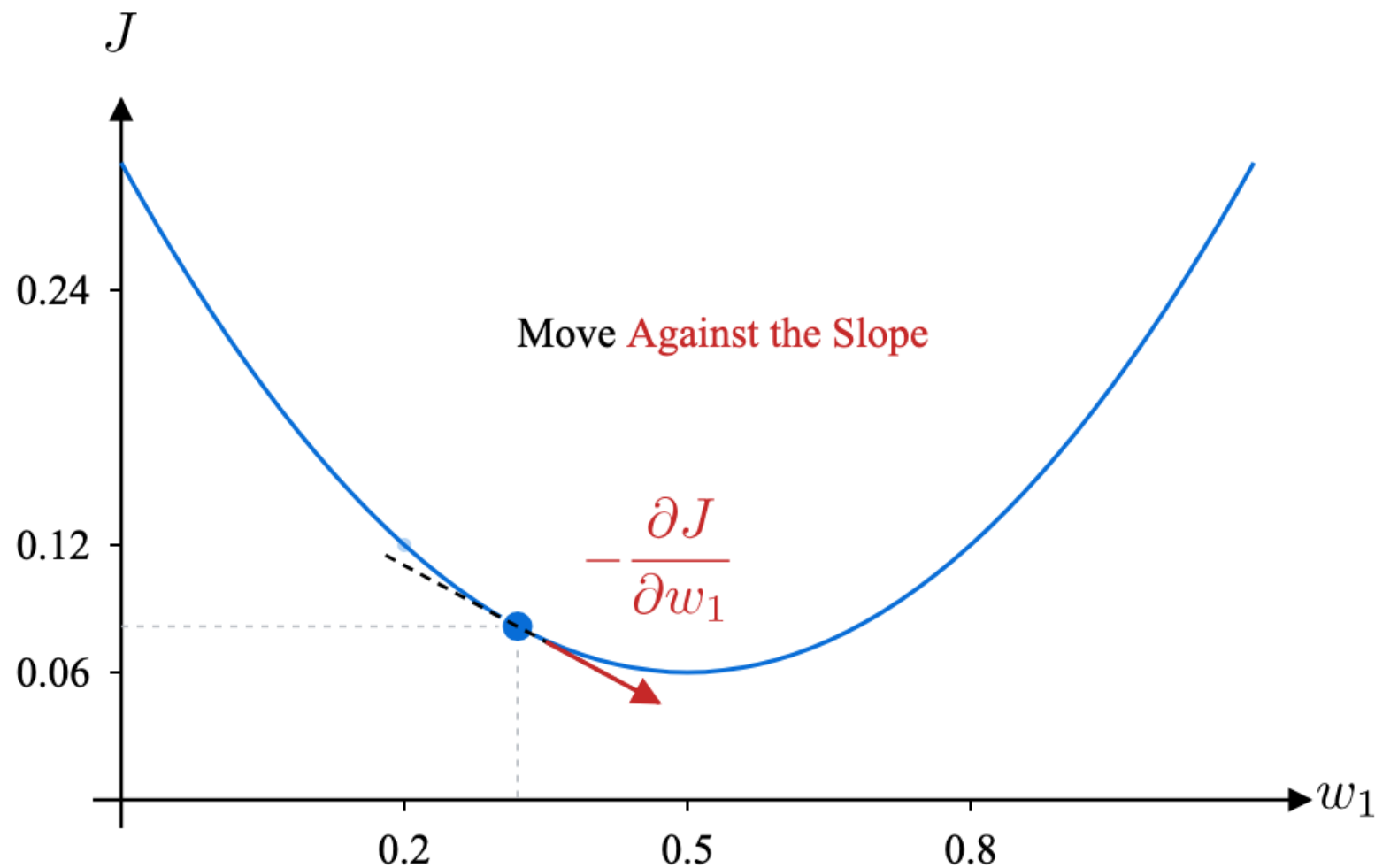
$$w_2 = 0, \quad b = 0.4$$

$$w_1 = 0.200$$



$$J = 0.120$$

Gradient Descent: Move Downhill



Other Parameters Fixed

$$w_2 = 0, \quad b = 0.4$$

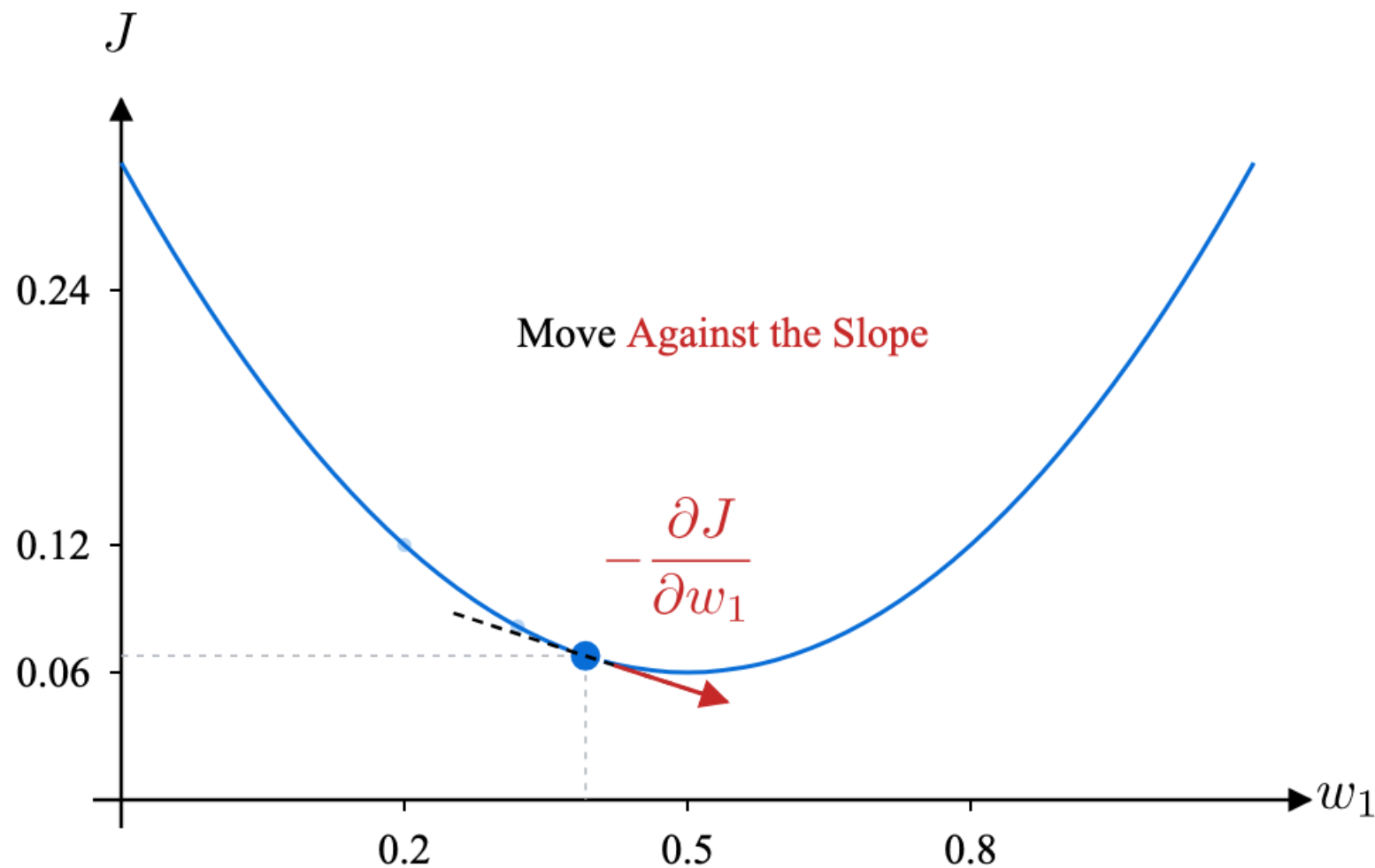
$$w_1 = 0.320$$



$$J = 0.082$$

$$w_1 \leftarrow w_1 - \eta \frac{\partial J}{\partial w_1}$$

Gradient Descent: Move Downhill



Other Parameters Fixed

$$w_2 = 0, \quad b = 0.4$$

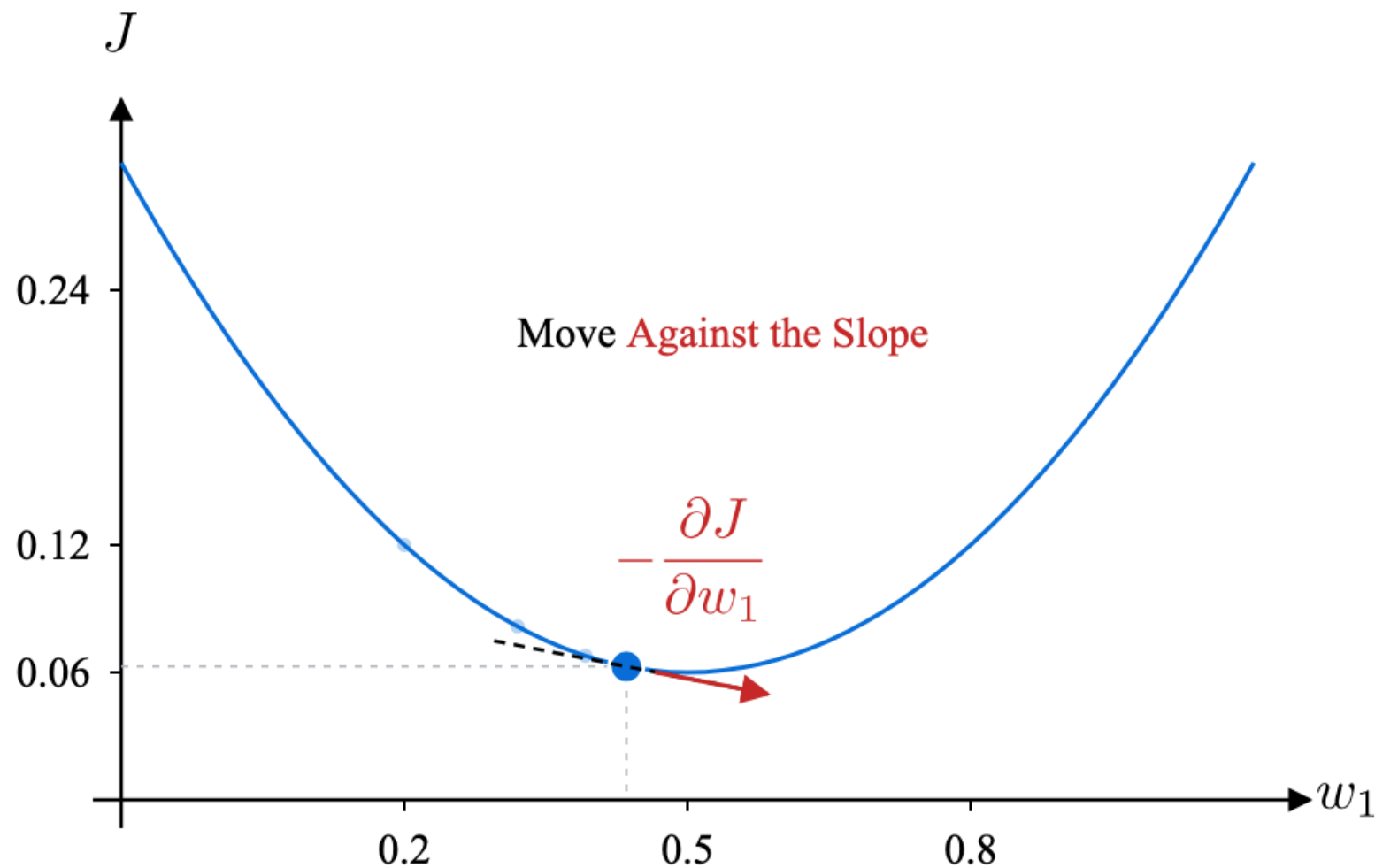
$$w_1 = 0.392$$



$$J = 0.068$$

$$w_1 \leftarrow w_1 - \eta \frac{\partial J}{\partial w_1}$$

Gradient Descent: Move Downhill



Other Parameters Fixed

$$w_2 = 0, \quad b = 0.4$$

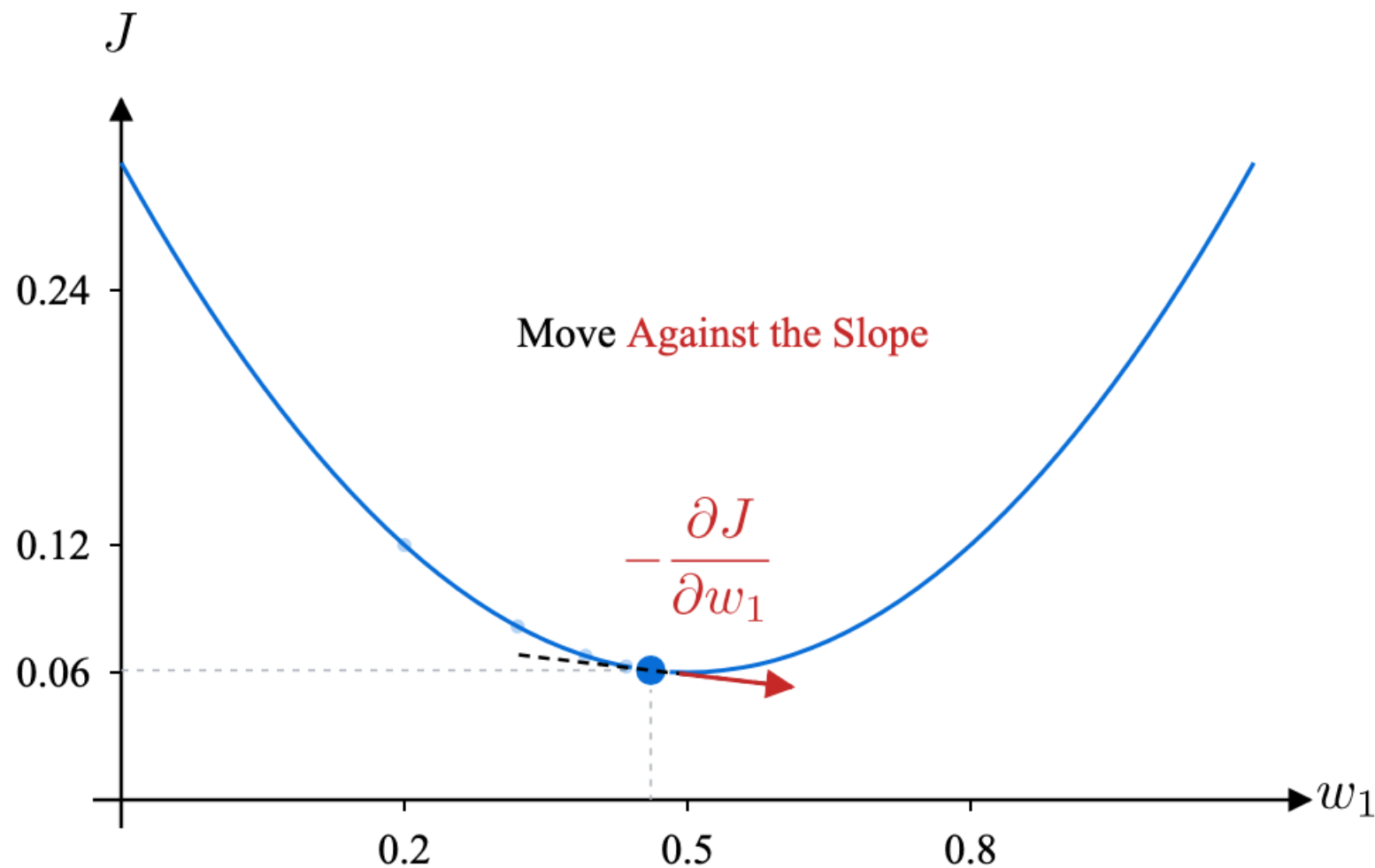
$$w_1 = 0.435$$



$$J = 0.063$$

$$w_1 \leftarrow w_1 - \eta \frac{\partial J}{\partial w_1}$$

Gradient Descent: Move Downhill



Other Parameters Fixed

$$w_2 = 0, \quad b = 0.4$$

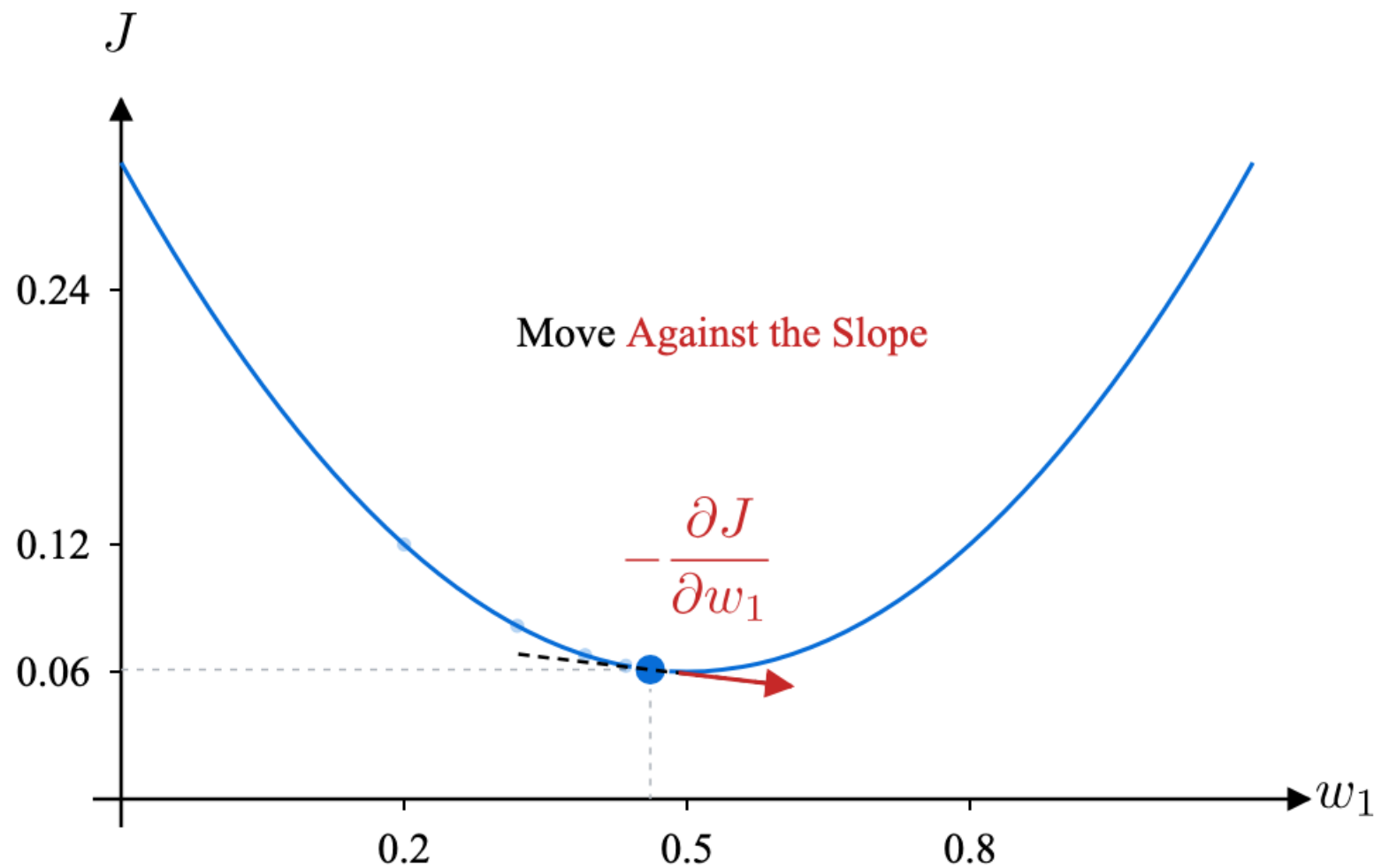
$$w_1 = 0.461$$



$$J = 0.061$$

$$w_1 \leftarrow w_1 - \eta \frac{\partial J}{\partial w_1}$$

Gradient Descent: Move Downhill



When Should We Stop?

Other Parameters Fixed

$$w_2 = 0, \quad b = 0.4$$

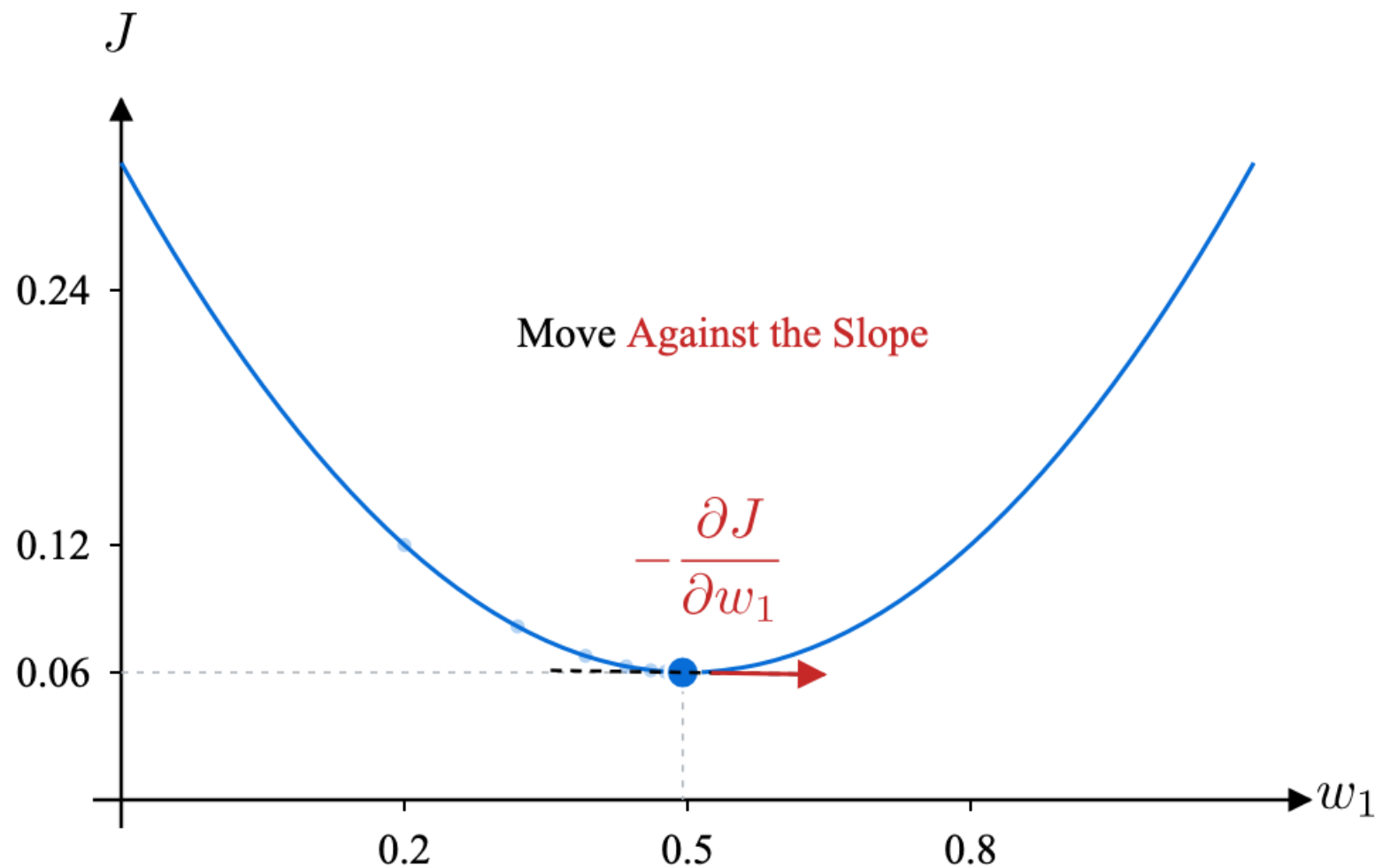
$$w_1 = 0.461$$



$$J = 0.061$$

$$w_1 \leftarrow w_1 - \eta \frac{\partial J}{\partial w_1}$$

Gradient Descent: Move Downhill



Other Parameters Fixed

$$w_2 = 0, \quad b = 0.4$$

$$w_1 = 0.495$$



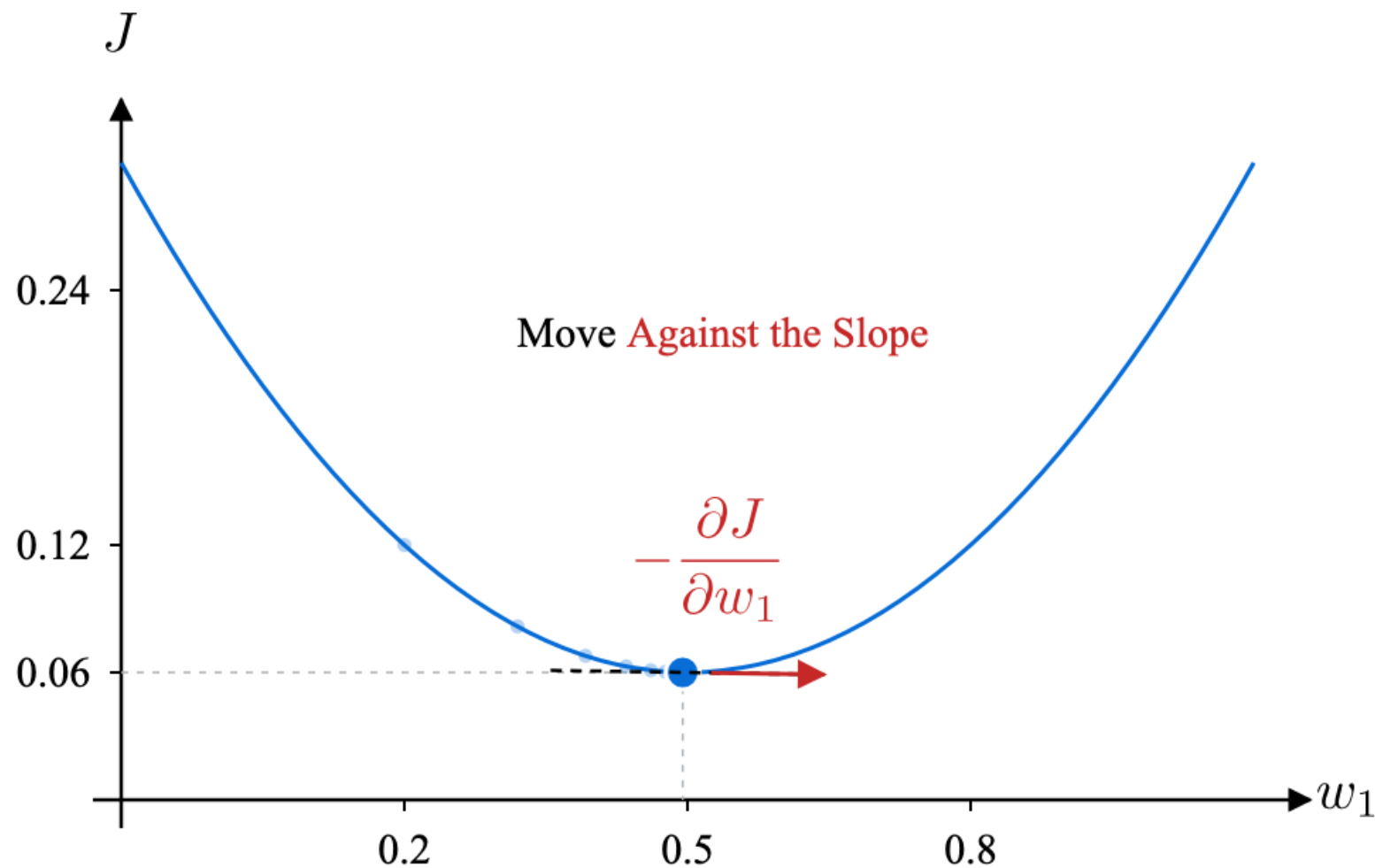
$$J = 0.060$$

$$w_1 \leftarrow w_1 - \eta \frac{\partial J}{\partial w_1}$$

When Should We Stop?

When the Derivative Is Sufficiently Close to Zero

Gradient Descent: Move Downhill



Other Parameters Fixed

$$w_2 = 0, \quad b = 0.4$$

$$w_1 = 0.495$$



$$J = 0.060$$

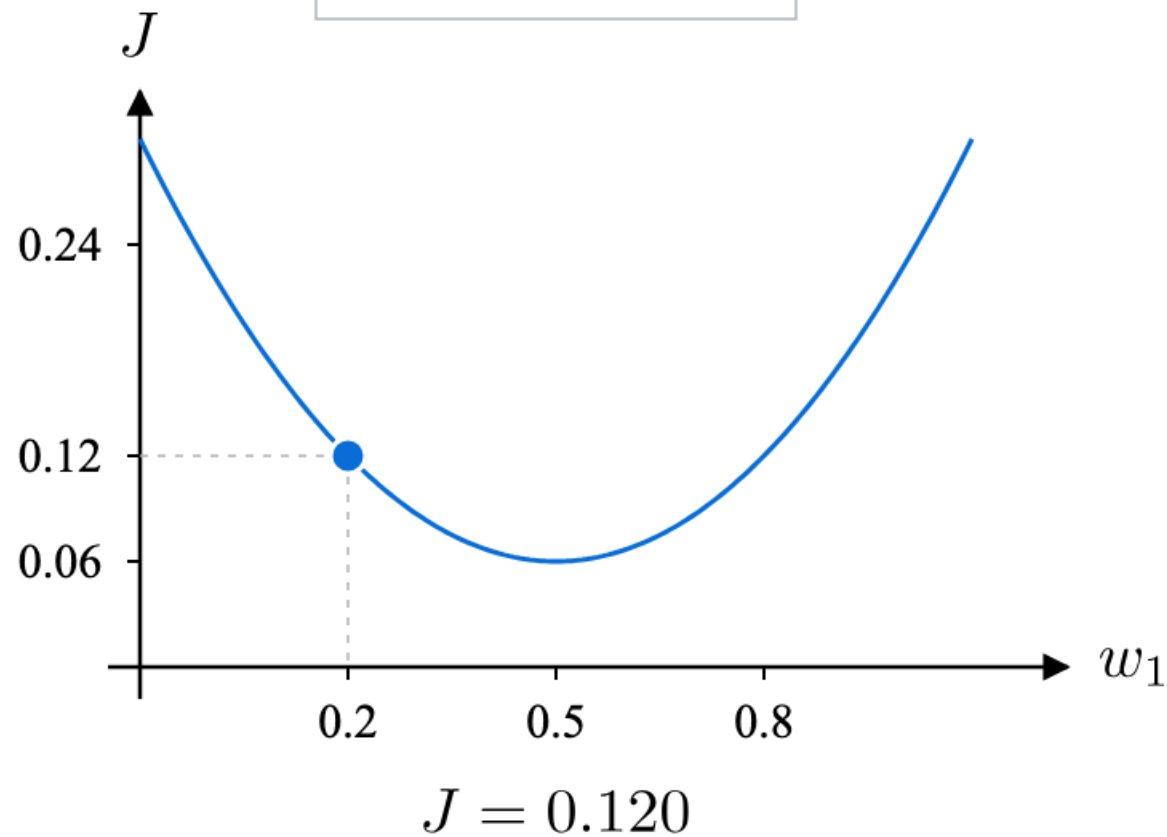
$$w_1 \leftarrow w_1 - \eta \frac{\partial J}{\partial w_1}$$

$$J = \frac{1}{n} \sum_{i=1}^n \left(y^{(i)} - z^{(i)} \right)^2$$

Learning Rate Matters

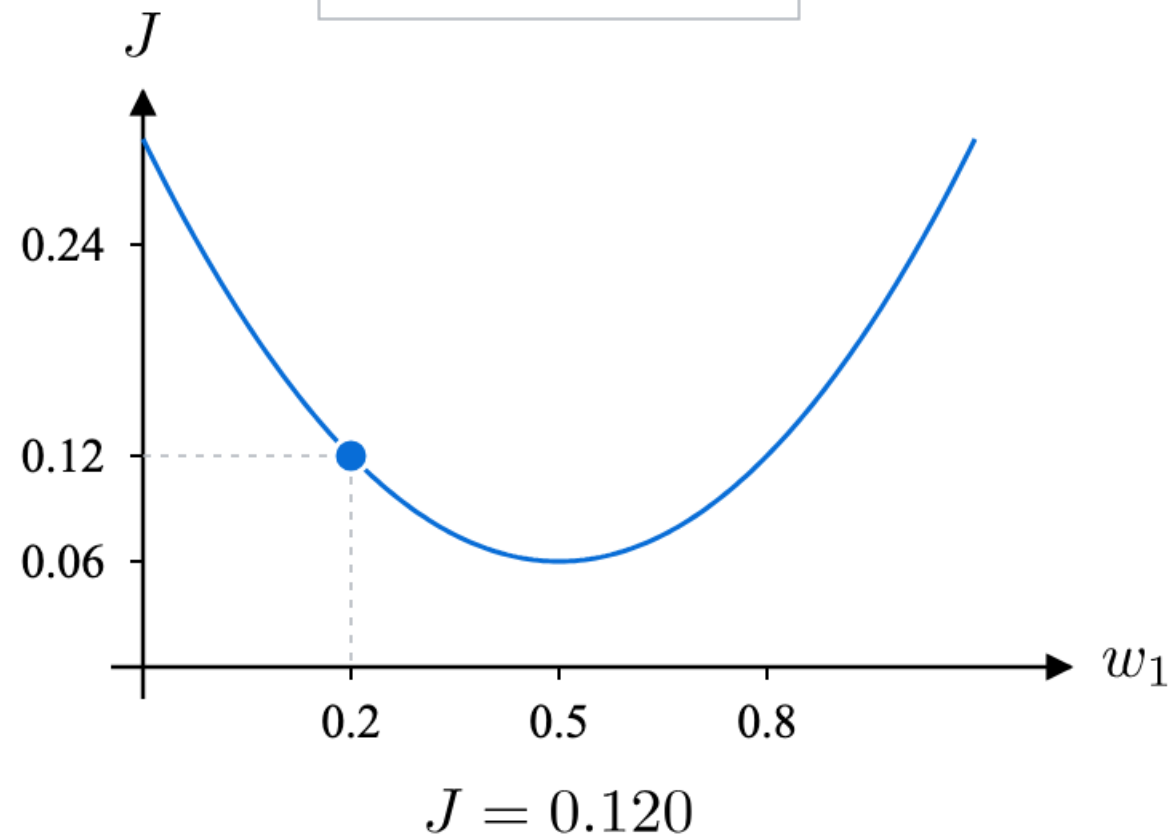
Too Small

$$\eta = 0.03$$



Too Large

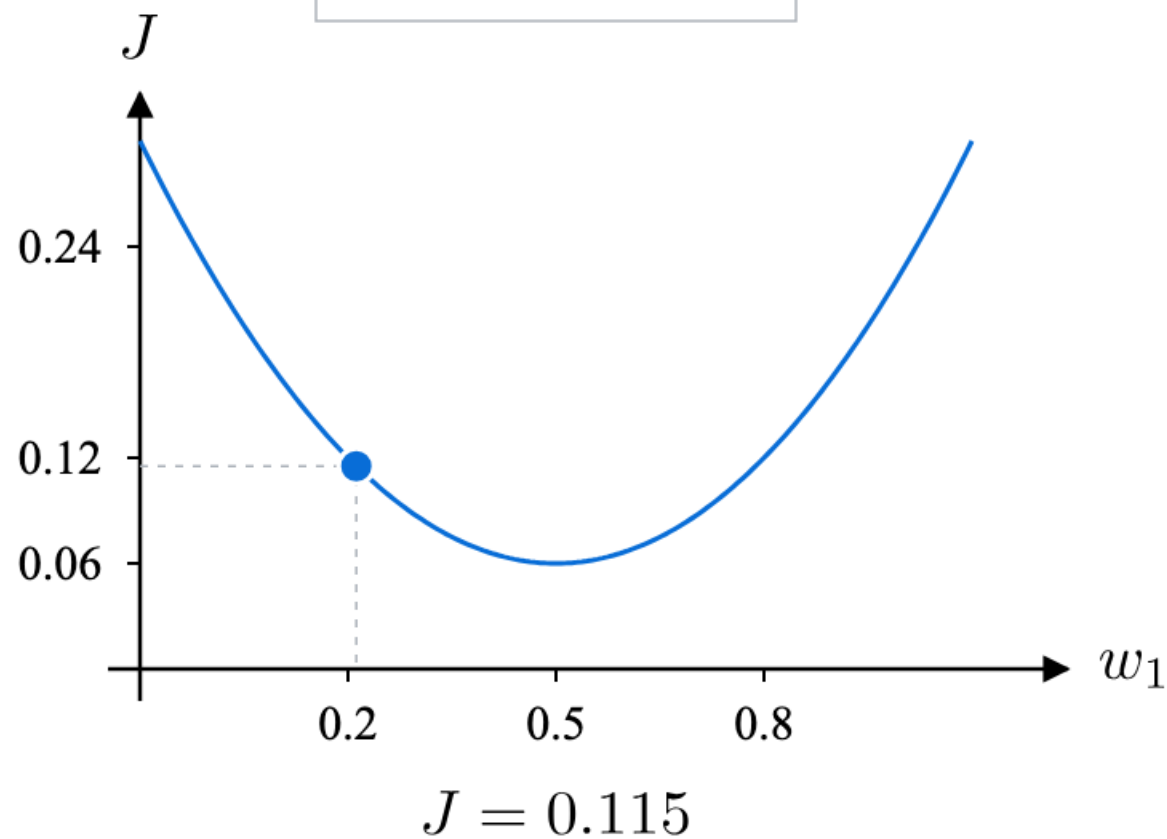
$$\eta = 1.55$$



Learning Rate Matters

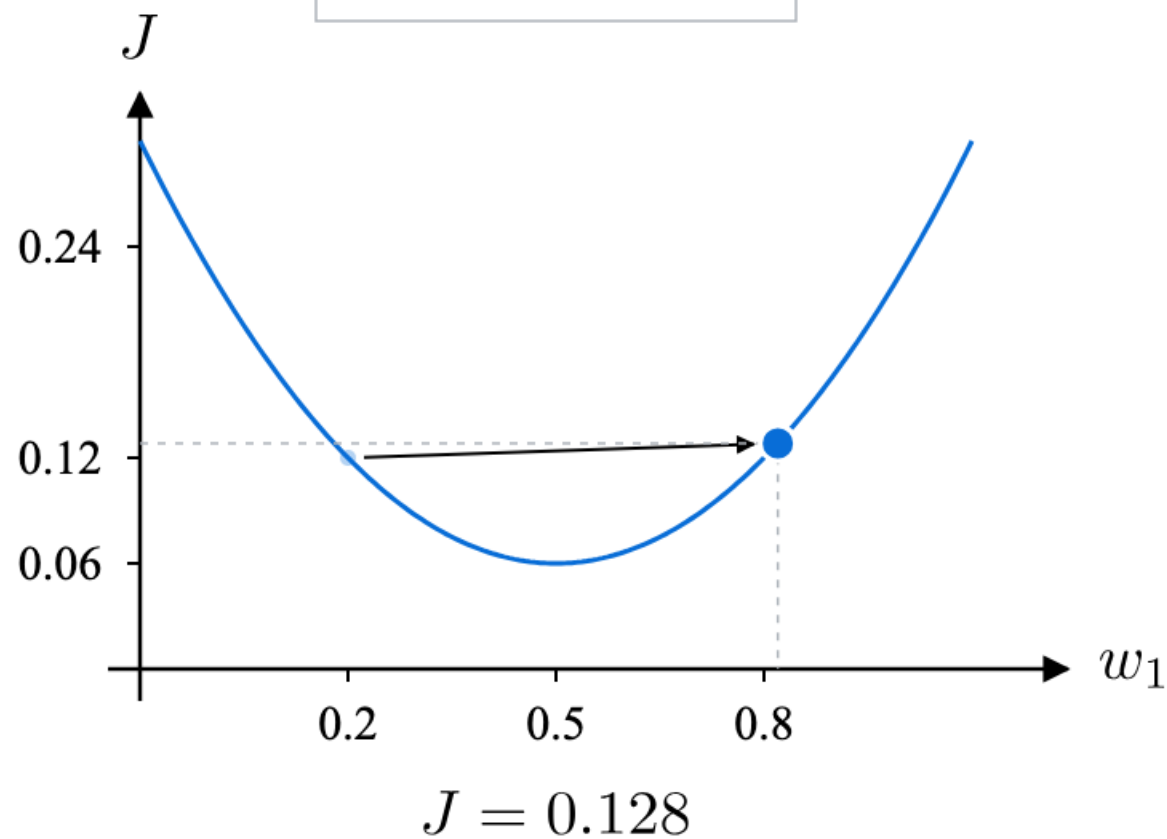
Too Small

$$\eta = 0.03$$



Too Large

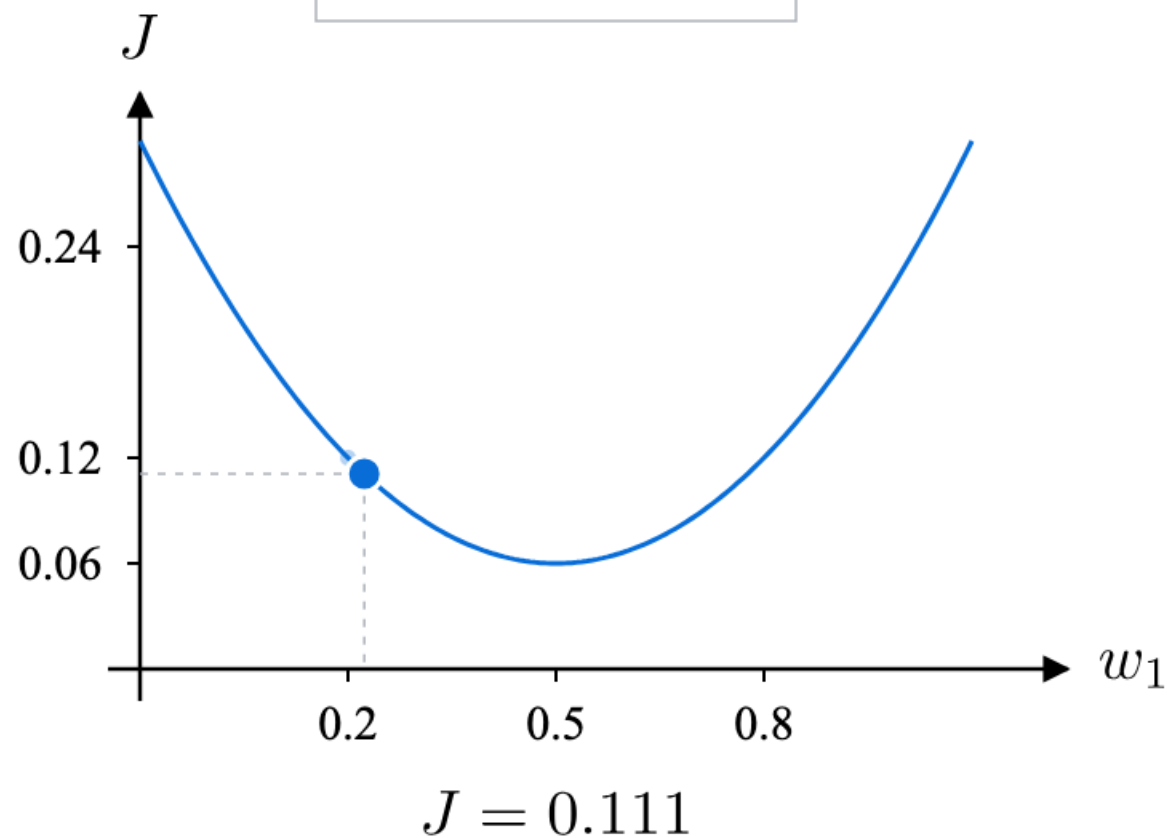
$$\eta = 1.55$$



Learning Rate Matters

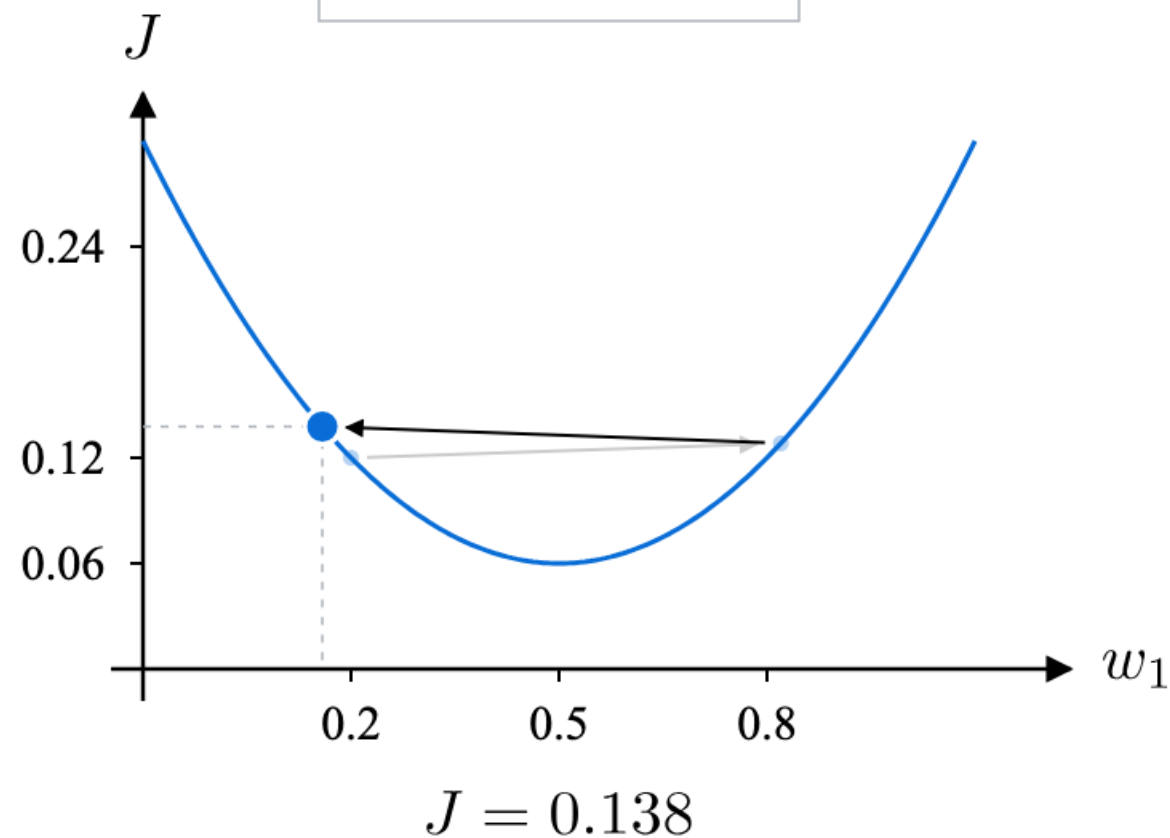
Too Small

$$\eta = 0.03$$



Too Large

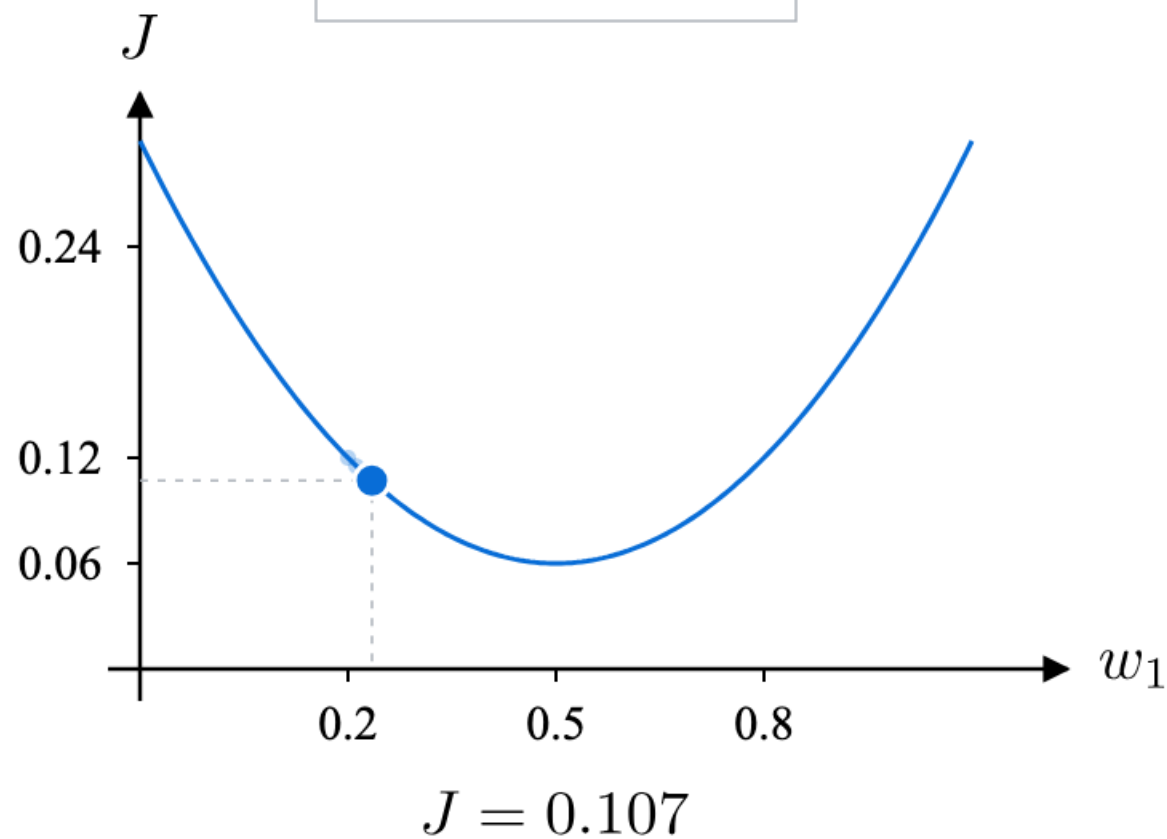
$$\eta = 1.55$$



Learning Rate Matters

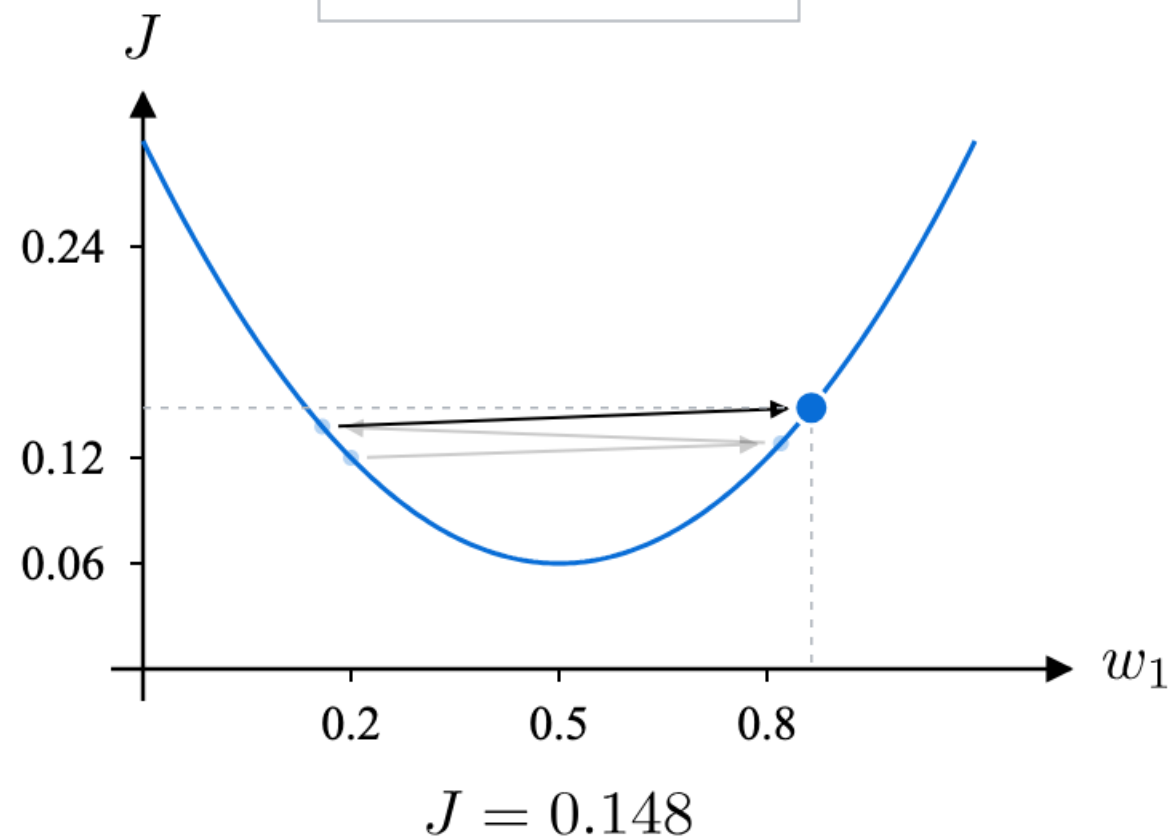
Too Small

$$\eta = 0.03$$



Too Large

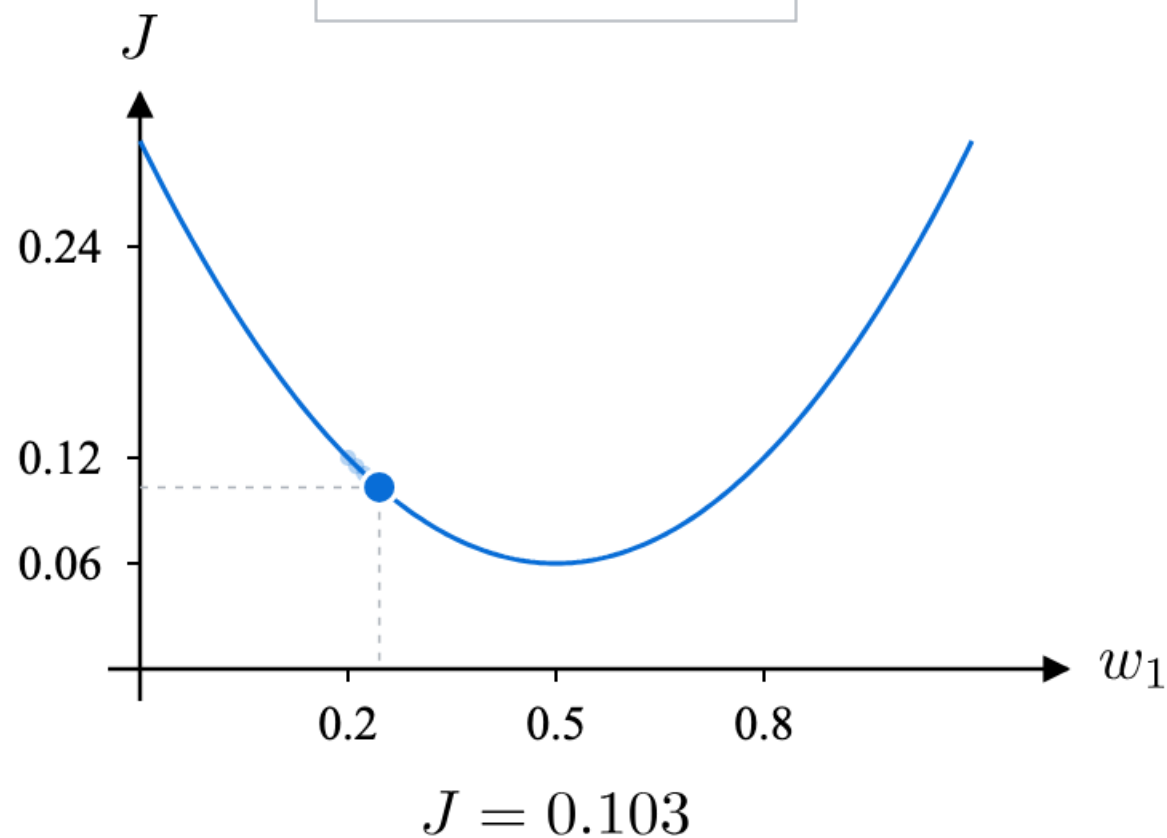
$$\eta = 1.55$$



Learning Rate Matters

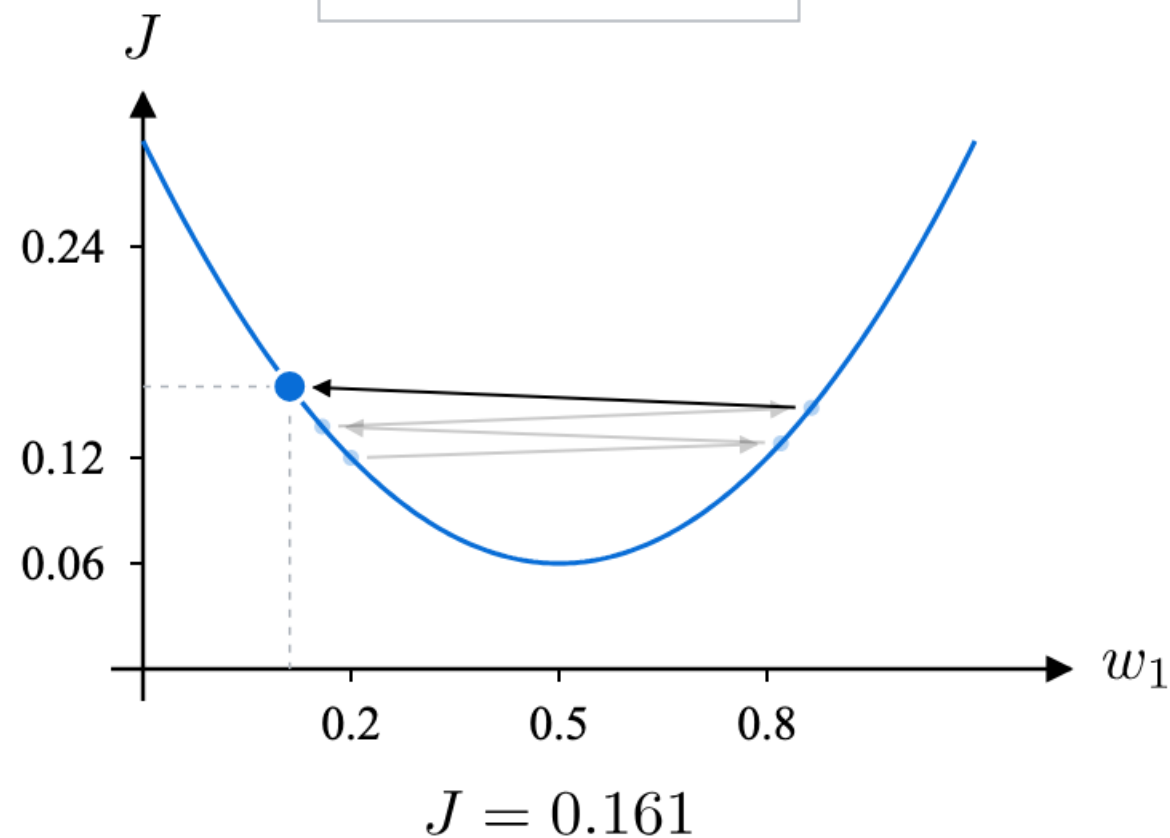
Too Small

$$\eta = 0.03$$



Too Large

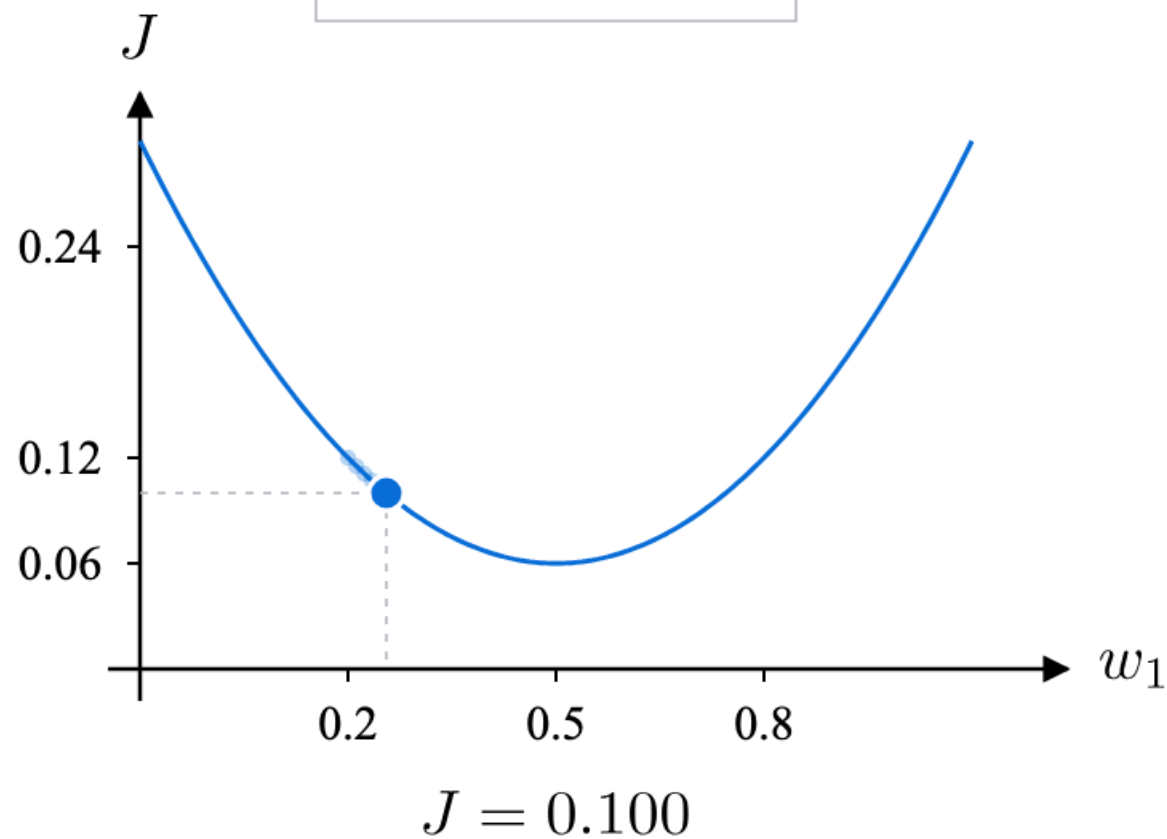
$$\eta = 1.55$$



Learning Rate Matters

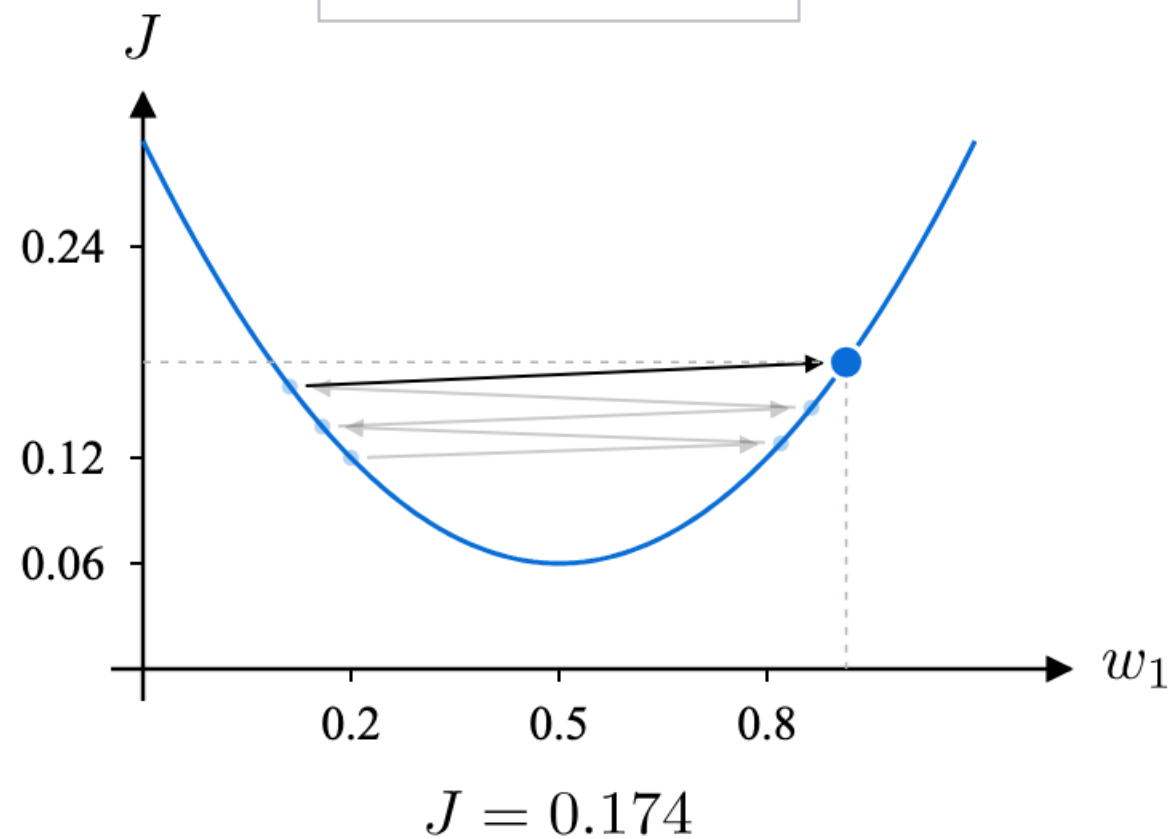
Too Small

$$\eta = 0.03$$



Too Large

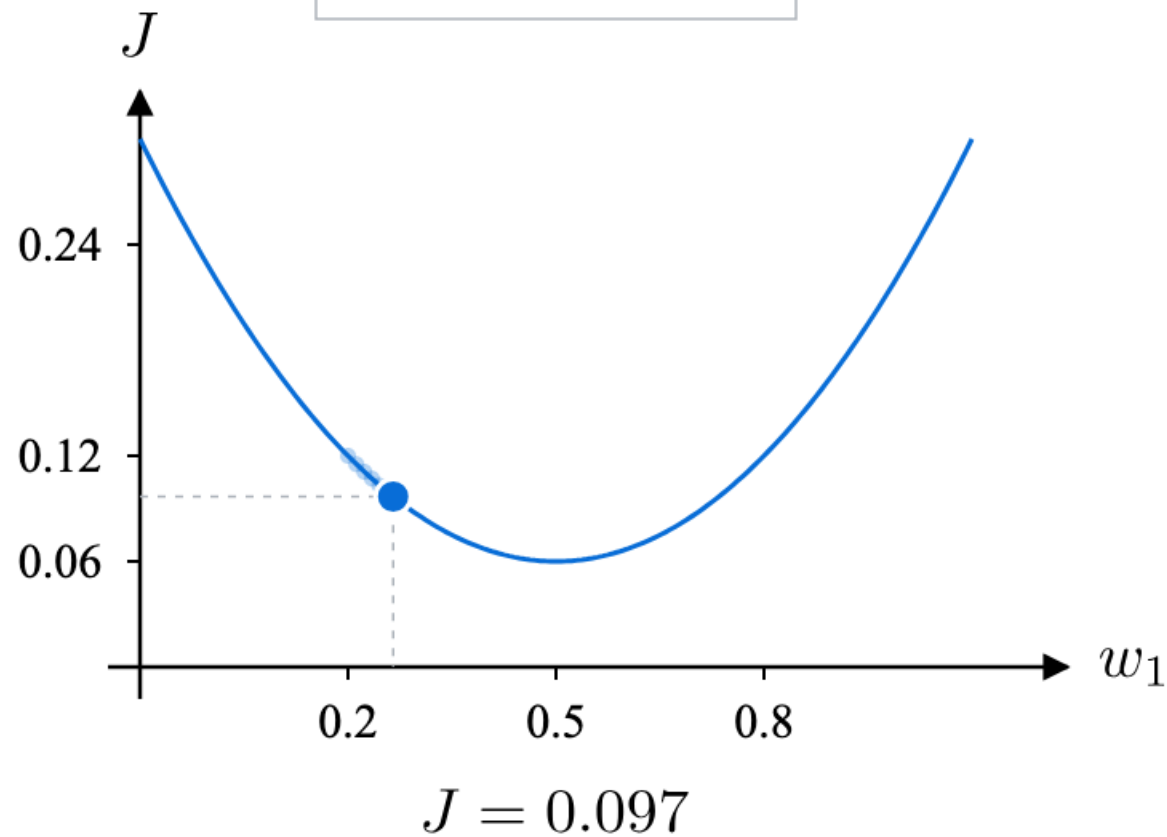
$$\eta = 1.55$$



Learning Rate Matters

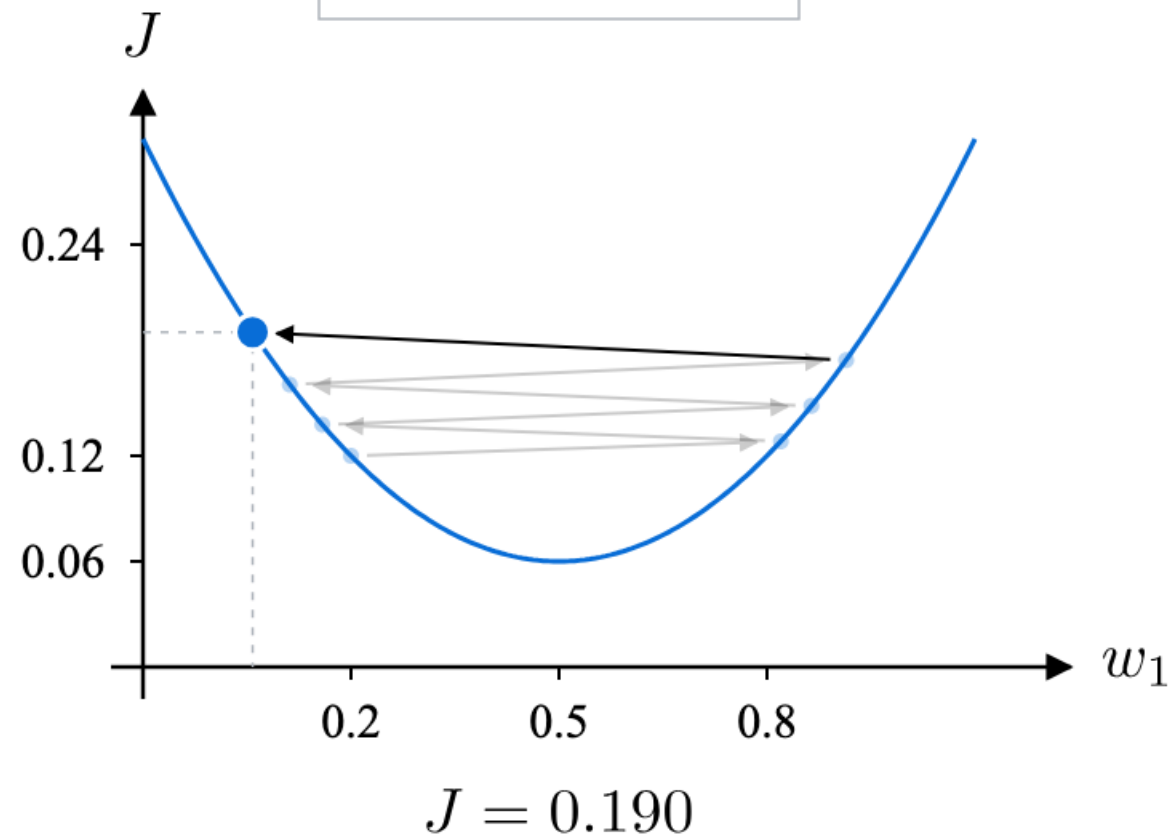
Too Small

$$\eta = 0.03$$



Too Large

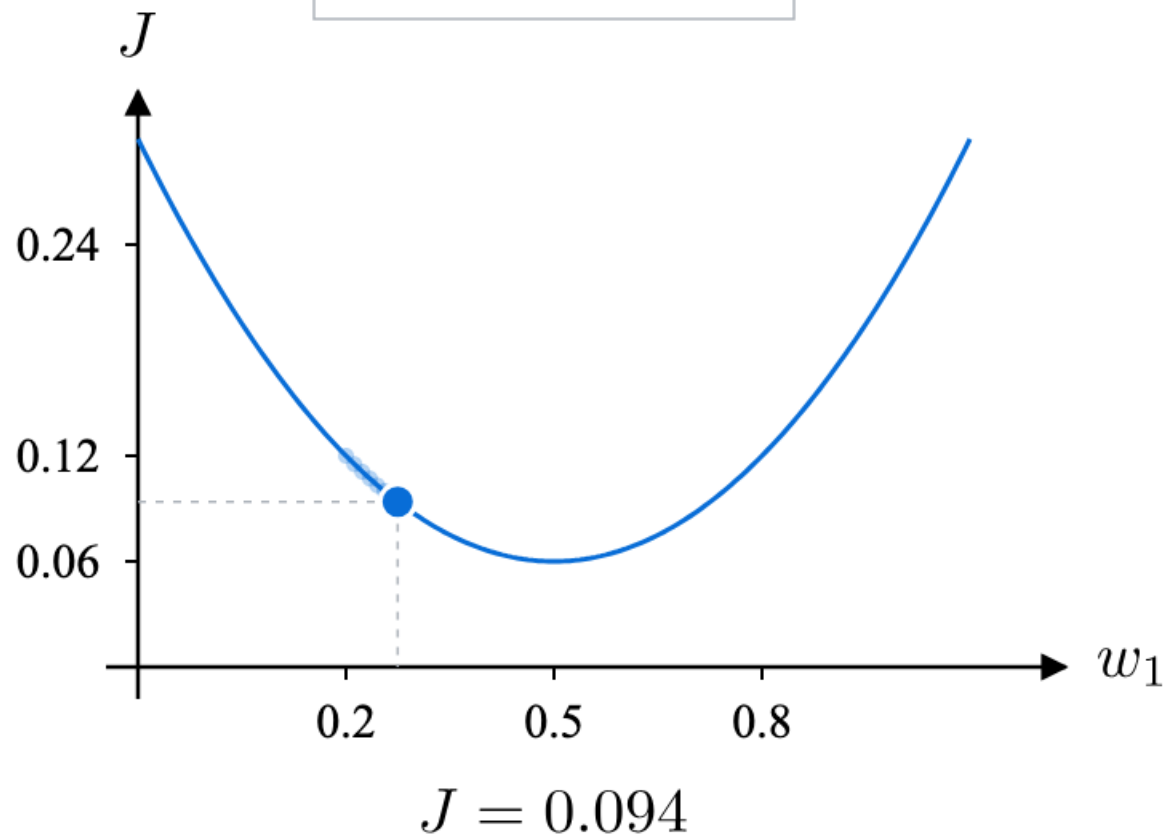
$$\eta = 1.55$$



Learning Rate Matters

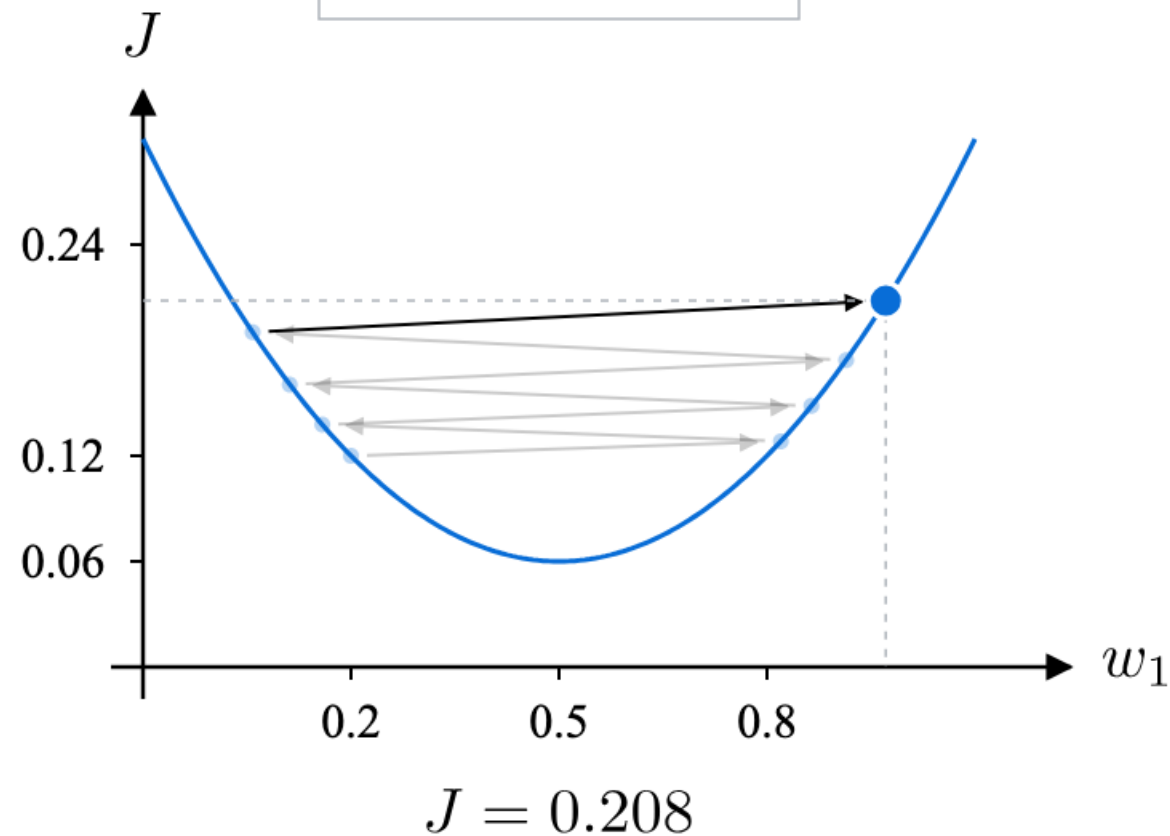
Too Small

$$\eta = 0.03$$



Too Large

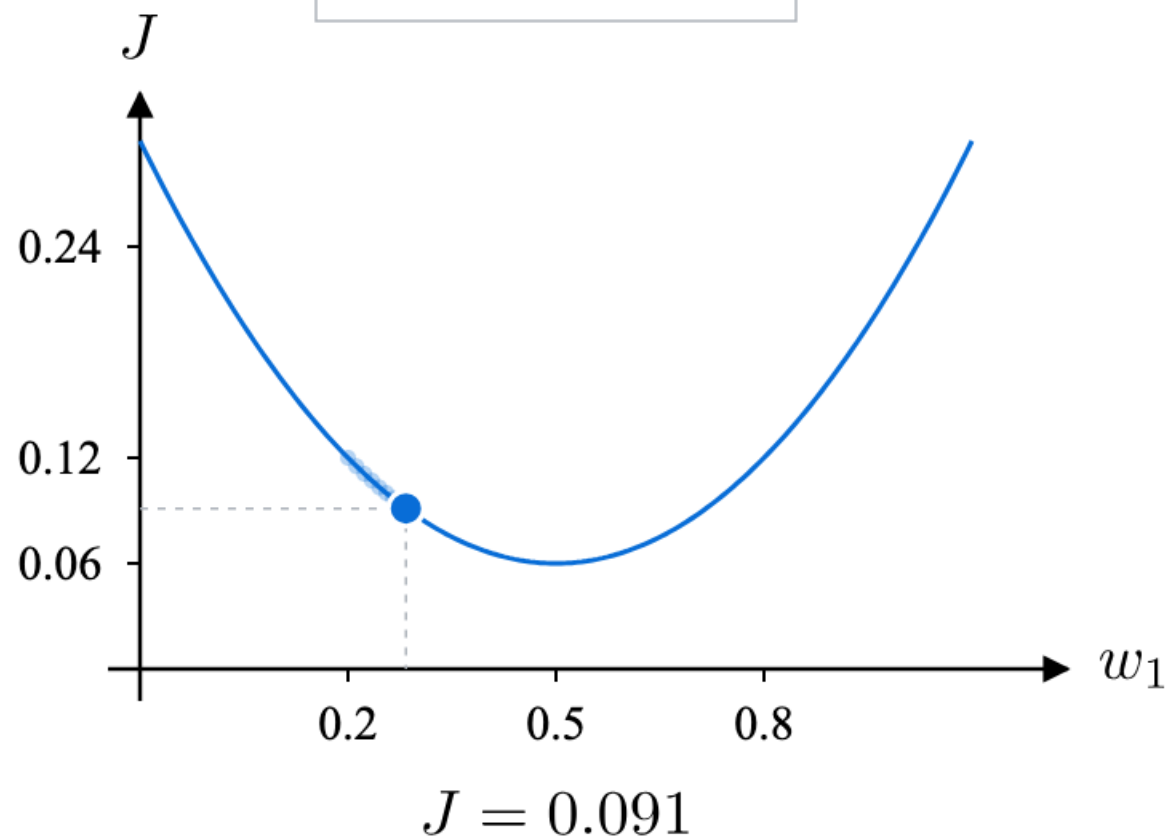
$$\eta = 1.55$$



Learning Rate Matters

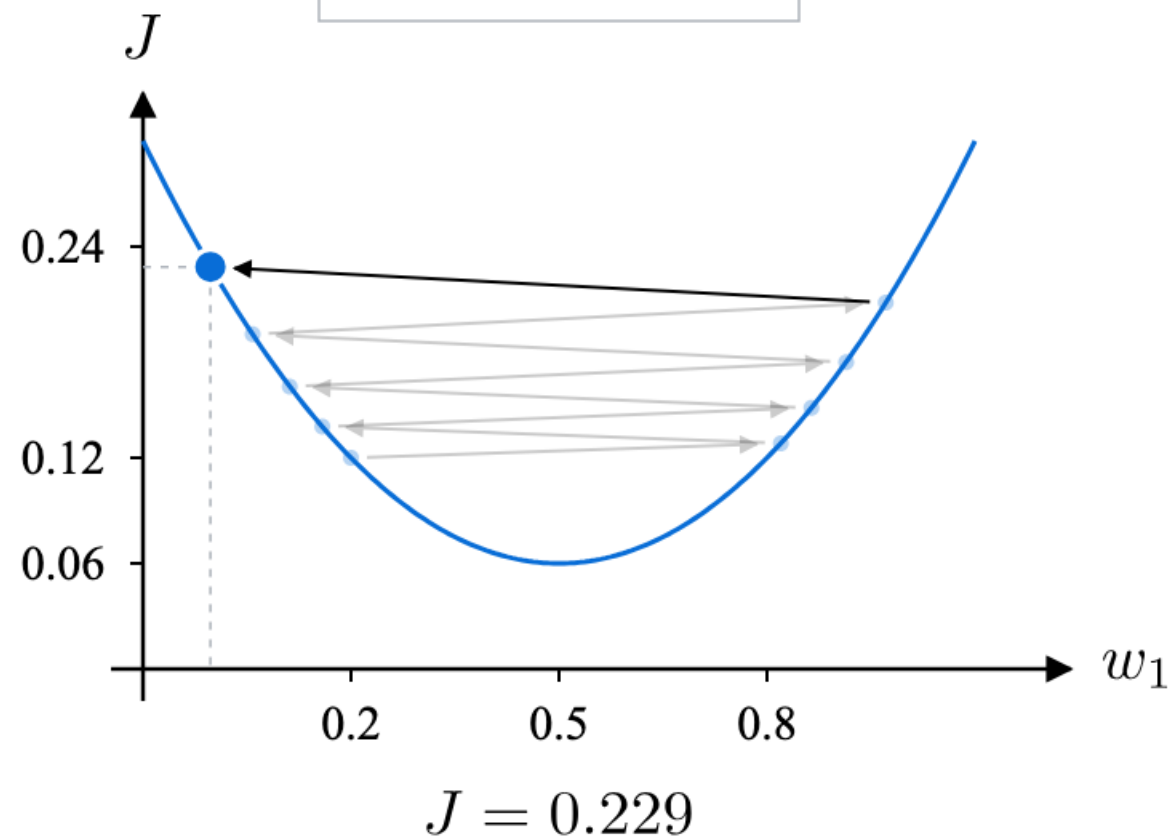
Too Small

$$\eta = 0.03$$



Too Large

$$\eta = 1.55$$



Batch Gradient Descent (BGD)

All Training Examples

$$\boldsymbol{w} \leftarrow \boldsymbol{w} - \eta \underbrace{\left[-\frac{2}{n} \sum_{i=1}^n (y^{(i)} - z^{(i)}) \boldsymbol{x}^{(i)} \right]}_{\frac{\partial J}{\partial \boldsymbol{w}}}$$

$$b \leftarrow b - \eta \underbrace{\left[-\frac{2}{n} \sum_{i=1}^n (y^{(i)} - z^{(i)}) \right]}_{\frac{\partial J}{\partial b}}$$

Gradient Descent with Two Weights

Demo 5

maib7002.dirk.hk/L2.zip

Demo 6: `adaline_full_batch.ipynb`

Stochastic Gradient Descent (SGD)

Stochastic Gradient Descent (SGD)

BGD

All Training Examples

$$\boldsymbol{w} \leftarrow \boldsymbol{w} - \eta \underbrace{\left[-\frac{2}{n} \sum_{i=1}^n (y^{(i)} - z^{(i)}) \boldsymbol{x}^{(i)} \right]}_{\frac{\partial J}{\partial \boldsymbol{w}}}$$

$$b \leftarrow b - \eta \underbrace{\left[-\frac{2}{n} \sum_{i=1}^n (y^{(i)} - z^{(i)}) \right]}_{\frac{\partial J}{\partial b}}$$

Stochastic Gradient Descent (SGD)

BGD

All Training Examples

$$\boldsymbol{w} \leftarrow \boldsymbol{w} - \eta \underbrace{\left[-\frac{2}{n} \sum_{i=1}^n (y^{(i)} - z^{(i)}) \boldsymbol{x}^{(i)} \right]}_{\frac{\partial J}{\partial \boldsymbol{w}}}$$

$$b \leftarrow b - \eta \underbrace{\left[-\frac{2}{n} \sum_{i=1}^n (y^{(i)} - z^{(i)}) \right]}_{\frac{\partial J}{\partial b}}$$

SGD

One Training Example

$$\boldsymbol{w} \leftarrow \boldsymbol{w} - \eta \underbrace{\left[-2(y^{(i)} - z^{(i)}) \boldsymbol{x}^{(i)} \right]}_{\frac{\partial J_i}{\partial \boldsymbol{w}}}$$

$$b \leftarrow b - \eta \underbrace{\left[-2(y^{(i)} - z^{(i)}) \right]}_{\frac{\partial J_i}{\partial b}}$$

Stochastic Gradient Descent (SGD)

BGD

All Training Examples

$$\boldsymbol{w} \leftarrow \boldsymbol{w} - \eta \underbrace{\left[-\frac{2}{n} \sum_{i=1}^n (y^{(i)} - z^{(i)}) \boldsymbol{x}^{(i)} \right]}_{\frac{\partial J}{\partial \boldsymbol{w}}}$$

$$b \leftarrow b - \eta \underbrace{\left[-\frac{2}{n} \sum_{i=1}^n (y^{(i)} - z^{(i)}) \right]}_{\frac{\partial J}{\partial b}}$$

MBGD

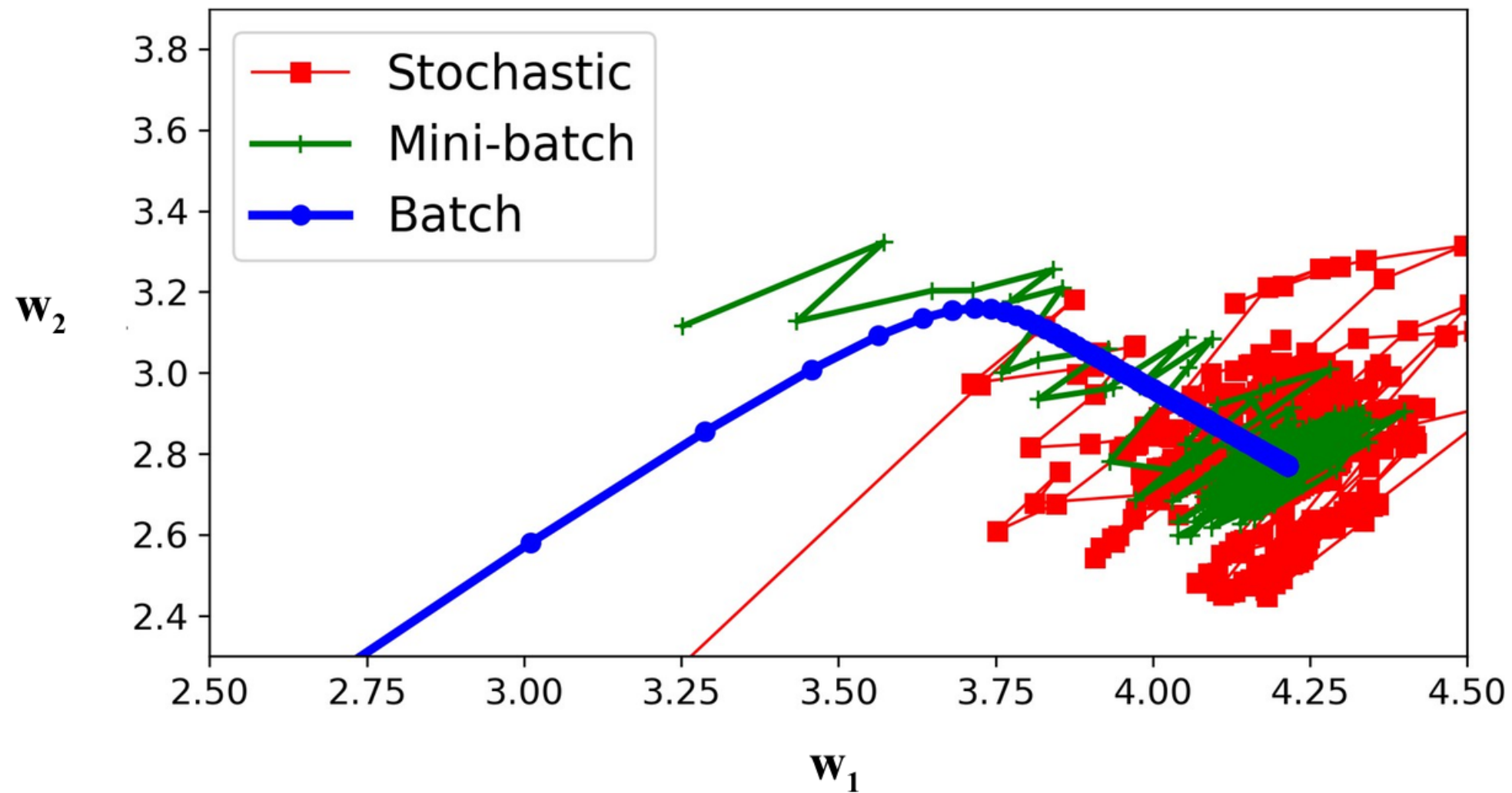
SGD

One Training Example

$$\boldsymbol{w} \leftarrow \boldsymbol{w} - \eta \underbrace{\left[-2(y^{(i)} - z^{(i)}) \boldsymbol{x}^{(i)} \right]}_{\frac{\partial J_i}{\partial \boldsymbol{w}}}$$

$$b \leftarrow b - \eta \underbrace{\left[-2(y^{(i)} - z^{(i)}) \right]}_{\frac{\partial J_i}{\partial b}}$$

Mini-Batch Gradient Descent (MBGD) Uses a Small Group of Examples per Update

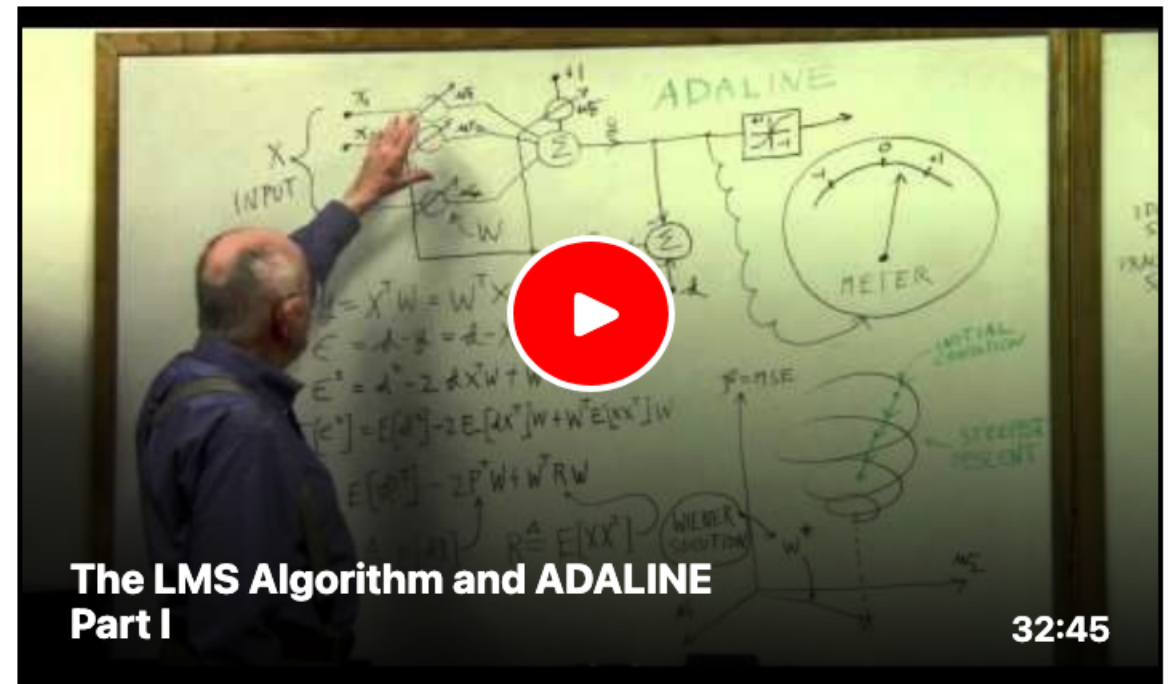


Adaline's Origins

Bernard Widrow and Ted Hoff

Adaline · 1960

<https://www.youtube.com/watch?v=hc2Zj55j1zU>



References and Suggested Reading

Core Reading

Machine Learning with PyTorch and Scikit-Learn

Raschka, Liu and Mirjalili · 2022

Chapter 1: Giving Computers the Ability to Learn from Data

Chapter 2: Training Simple Machine Learning Algorithms for Classification

Further Reading

Hands-On Machine Learning with Scikit-Learn and PyTorch

Aurélien Géron · 2025

Math Refresher

[Linear Algebra Review and Reference](#)

Zico Kolter, updated by Chuong Do

Python Refreshers

[NumPy Introduction](#)

[10 Minutes to pandas](#)

[Matplotlib Quick Start](#)

Exit Ticket