



Introduction to Data Science

The Neural Network

22 Sep 2026

The Neural Network

Part 1 Roadmap

1. Why learn from examples?

Use handwritten digits to see where hand-written rules become difficult.

2. Build and read a model

Vectorize an image, then work through the sleep-and-study case.

3. Express the model as a neuron

Identify inputs, weights, bias, and activation; then check the full calculation.

We will turn an example into a prediction, one step at a time.

Building Intelligent Machines

Why Start with Recognition?

The textbook starts with familiar abilities: recognizing faces, separating objects from backgrounds, and understanding voices.

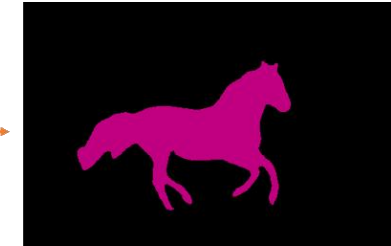
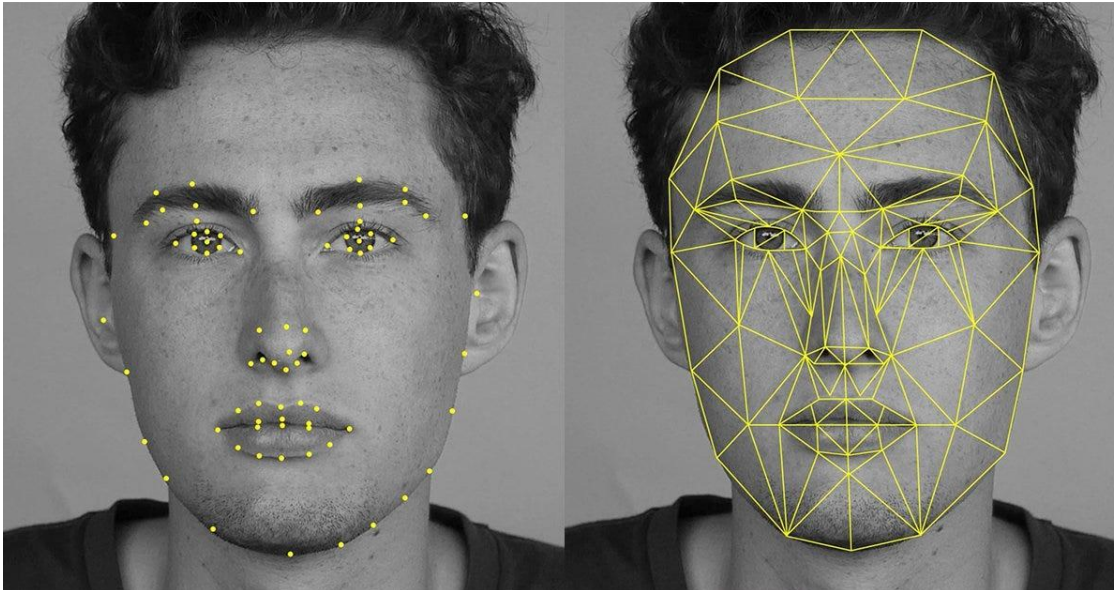
It then asks how machines could perform useful recognition tasks, such as reading an image or assisting a driver.

The central question is not only how fast a computer calculates. It is how we specify a useful recognition model.

The chapter begins with tasks that are easier to recognize than to describe as rules.

Building Intelligent Machines

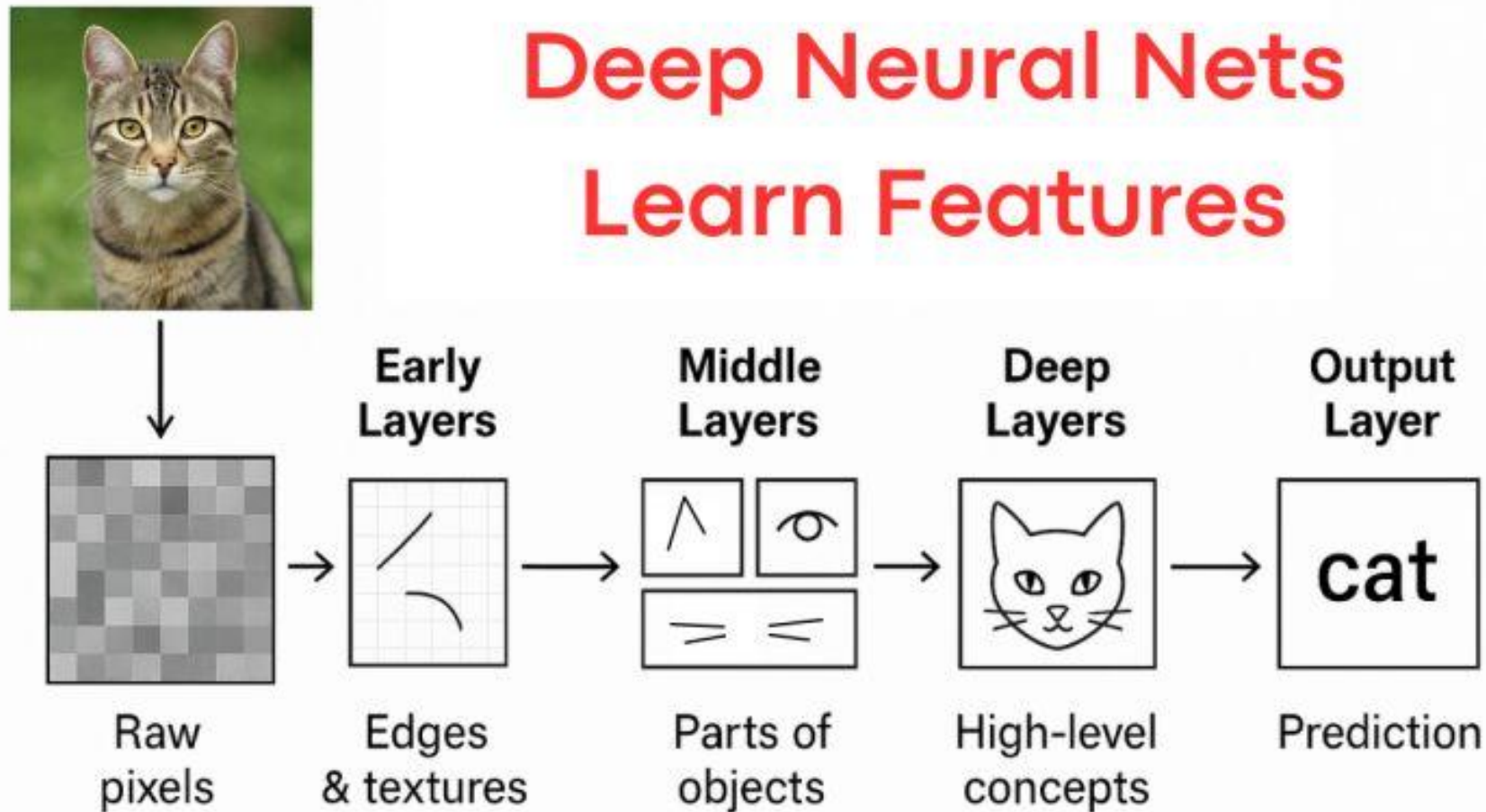
Why Start with Recognition?



The chapter begins with tasks that are easier to recognize than to describe as rules.

Building Intelligent Machines

Why Start with Recognition?



The chapter begins with tasks that are easier to recognize than to describe as rules.

The Neural Network

The Limits of Traditional Computer Programs

A CLEAR PROCEDURE

Given two numbers, multiply them.

The programmer can specify the arithmetic steps precisely.

$$\begin{array}{r}
 2435 \\
 \times 35 \\
 \hline
 12175 \\
 + 73050 \\
 \hline
 85225
 \end{array}$$

A RECOGNITION TASK

Given a handwritten image, identify the digit.

The programmer must decide which visual differences matter and which should be ignored.



Result: 3

A fast calculation is not the same as a good recognition rule.

Traditional Programs

Handwritten Digits Are Not Identical



Each row shows different versions of the same digit.

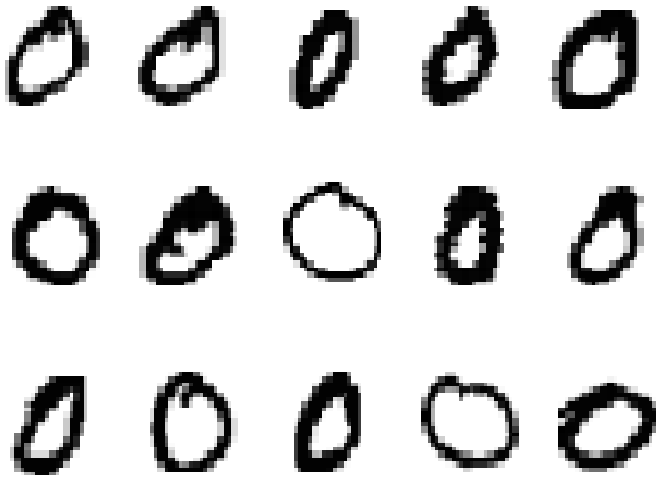
The shape, thickness, and tilt vary, but the intended label stays the same.

A useful model should recognize the label despite these variations.

Different pixel patterns can represent the same digit.

Traditional Programs

Handwritten Digits Are Not Identical



Each row shows different versions of the same digit.

The shape, thickness, and tilt vary, but the intended label stays the same.

A useful model should recognize the label despite these variations.

Different pixel patterns can represent the same digit.

Traditional Programs

A First Rule for Recognizing Zero



This is a possible rule.

It seems reasonable for neat zeros, but a rule must also work on messy handwriting.

What happens if the writer leaves a small gap in the loop?

A plausible rule can still fail on a small change in appearance.

Traditional Programs

Where Should the Rule Draw the Line?



It describes this as a zero that is difficult to distinguish from a six by a rule.

Should a small opening still count as a closed loop?

A distance threshold adds another rule, but we must choose where that threshold belongs.

The difficult part is deciding which visual changes should preserve the label.

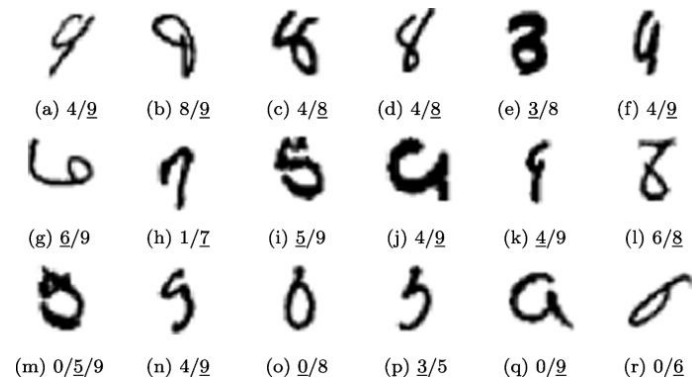
Traditional Programs

Why Not Keep Adding Rules?

ONE PAIR IS NOT ENOUGH

A rule that separates 0 from 6 does not automatically separate 3 from 5.

A rule for 3 and 5 does not automatically separate 4 from 9.



Recognizing many variations can require a large, fragile collection of rules.

MORE RULES, MORE EXCEPTIONS

We can keep adding hand-designed features and thresholds.

The textbook's concern is that this process becomes difficult to maintain for complex recognition tasks.

The Mechanics of Machine Learning

Learn from Examples and Corrections

1. Observe examples

The book uses the example of a child seeing different dogs.

2. Make a guess

A model maps the observed input to a proposed answer.

3. Use feedback

A wrong guess provides information for modifying the model.



Examples and feedback can guide the model instead of a long list of visual rules.

The Mechanics of Machine Learning

Two Ways to Build the Predictor

HAND-WRITTEN RULES

The programmer specifies the detailed decision rules.

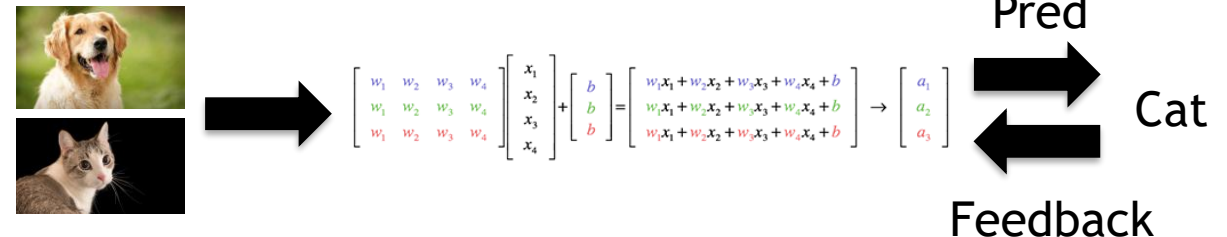
Each new exception may require another rule.

```
try {
    int buckets[] = new int[15];
    buckets[15] = 38 / 6;
} catch (ArithmeticException a) {
    System.out.println("Arithmetic Error!");
} catch (ArrayIndexOutOfBoundsException a) {
    System.out.println("Array out of bounds!");
} catch (Exception a) {
    System.out.println("Generic error");
}
```

LEARNING FROM EXAMPLES

The programmer chooses a model and a way to update it.

Examples guide the numerical settings used by the model.



Machine learning changes how the prediction rule is obtained.

The Mechanics of Machine Learning

Read the Model Notation

$$\text{prediction} = h(x, \theta)$$

Symbol	Meaning
x	One input example, written as a vector.
θ	The numerical parameters used by the model.
h	The rule that combines the input and parameters.

The model h uses an input x and parameter values θ to produce a prediction.

The Mechanics of Machine Learning

What Changes During Learning?

PREDICT ONE EXAMPLE

Choose one input x .

Use the current parameter values θ .

Compute the model's answer $h(x, \theta)$.

IMPROVE THE MODEL

Compare predictions with the example labels.

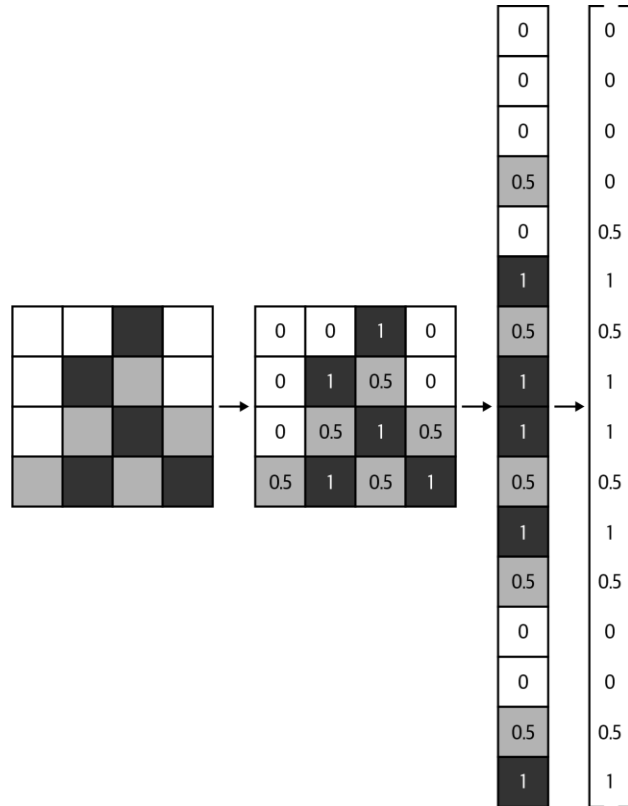
Change θ to improve the predictions.

Use the updated model on further examples.

Learning adjusts parameters; prediction uses the current parameters.

The Mechanics of Machine Learning

From an Image to an Input Vector



An image must be represented numerically before the model can calculate with it.

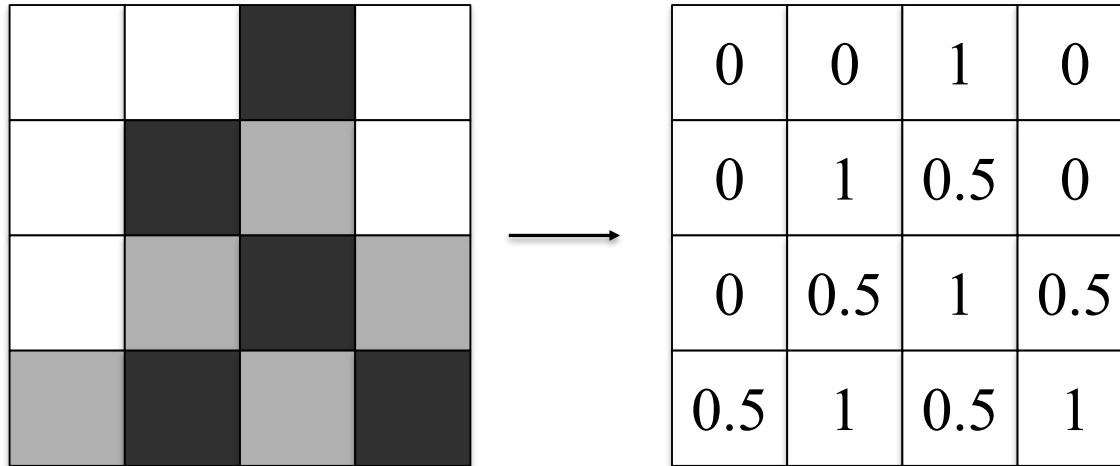
Figure shows a small grayscale image, a grid of pixel values, and a vector containing those values.

The picture and the vector carry the same listed pixel values in different arrangements.

Vectorization changes the arrangement of the pixel values, not the values themselves.

The Mechanics of Machine Learning

Read the Pixel Values First



In the drawing, 0 is white, 0.5 is gray, and 1 is dark.

The entry at row 1, column 3 is 1.

There are $4 \times 4 = 16$ pixel values.

Each cell contributes one numerical input value.

The Mechanics of Machine Learning

Follow the Order of Figure

Read each column from top to bottom, then move to the next column.

Column 1

$$\begin{bmatrix} 0 \\ 0 \\ 0 \\ 0.5 \end{bmatrix}$$

Column 2

$$\begin{bmatrix} 0 \\ 1 \\ 0.5 \\ 1 \end{bmatrix}$$

Column 3

$$\begin{bmatrix} 1 \\ 0.5 \\ 1 \\ 0.5 \end{bmatrix}$$

Column 4

$$\begin{bmatrix} 0 \\ 0 \\ 0.5 \\ 1 \end{bmatrix}$$

0	0	1	0
0	1	0.5	0
0	0.5	1	0.5
0.5	1	0.5	1

Stack these groups in order. The resulting vector has 16 entries.

Choose an order for the pixels and use that order consistently.

The Mechanics of Machine Learning

The Final Vector Still Represents the Image

$$\mathbf{x} = \begin{bmatrix} 0 & 0 & 0 & 0.5 & 0 & 1 & 0.5 & 1 & 1 & 0.5 & 1 & 0.5 & 0 & 0 & 0.5 & 1 \end{bmatrix}$$

The first four entries came from the first image column.

The next four came from the second image column, and so on.

An entry's position tells the model which pixel that value represents.

0	0	1	0
0	1	0.5	0
0	0.5	1	0.5
0.5	1	0.5	1

A vector entry is meaningful only when its position has a consistent interpretation.

The Mechanics of Machine Learning

Do Not Confuse x with θ

$$\text{prediction} = h(x, \theta)$$

Quantity	Where does it come from?	What does it do?
Input x	The current example.	Describes what the model sees.
Parameters θ	Values selected during learning.	Control how the model responds.
Prediction $h(x, \theta)$	The current model calculation.	Provides the proposed answer.

x contains the example; θ controls how the model interprets it.

Inputs describe an example; parameters describe the model's current settings.

The Mechanics of Machine Learning

Quick Check: Vectorizing the Image

A 4×4 grayscale image is written as one input vector. How many entries does it contain?

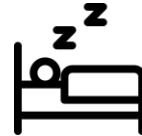
If the same pixel values are placed in a different order, is that necessarily the same input to the model?

Which quantity is learned: the image x or the parameter vector θ ?

Count the values and preserve their order before interpreting them.

The Mechanics of Machine Learning

The Exam Prediction Example

Recorded quantity	Meaning
x_1	Hours of sleep on the previous day. 
x_2	Hours spent studying on the previous day. 
Observed outcome	Whether exam performance was below or at/above the class average.

We will keep these meanings of x_1 and x_2 fixed throughout the example.

Two recorded features become the two coordinates of the input vector.

The Mechanics of Machine Learning

What Do -1 and $+1$ Mean?

OUTPUT -1

Predict performance below the class average.

The number -1 is a label for this category.



OUTPUT $+1$

Predict performance at or above the class average.

The number $+1$ is a label for the other category.



The perceptron returns a class label, not an exam score or a probability.

The Mechanics of Machine Learning

A Perceptron Uses a Score and a Threshold

$$z = \theta_1 x_1 + \theta_2 x_2 + \theta_0$$

FIRST: COMPUTE A SCORE

Multiply each input by its parameter.

Add both contributions and the constant term θ_0 .

THEN: CHOOSE THE LABEL

If $z < 0$, output -1 . 

If $z \geq 0$, output $+1$. 

The weighted score and the final class label are different quantities.

The Mechanics of Machine Learning

Use the Parameter Values

$$\theta = (-24, 3, 4)$$

$$z = 3x_1 + 4x_2 - 24$$

The order is θ_0 , θ_1 , θ_2 : constant term first, then the two input weights.

The book supplies these parameter values for the displayed example.

For now, we use them to calculate predictions.

Keep the parameter order clear before substituting the numbers.

The Mechanics of Machine Learning

Worked Prediction: A Positive Score

$x = (6, 2)$: 6 hours asleep, 2 hours studying

Step	Calculation
Sleep contribution	$3 \times 6 = 18$ 
Study contribution	$4 \times 2 = 8$ 
Score including the bias	$z = 18 + 8 - 24 = 2$

$$2 \geq 0 \rightarrow \text{prediction} = +1$$
 

The score is positive, so the rule selects the +1 class.

The Mechanics of Machine Learning

Worked Prediction: A Negative Score

$x = (4, 1)$: 4 hours asleep, 1 hour studying

Step	Calculation
Sleep contribution	$3 \times 4 = 12$ 
Study contribution	$4 \times 1 = 4$ 
Score including the bias	$z = 12 + 4 - 24 = -8$

$$-8 < 0 \rightarrow \text{prediction} = -1$$



The same rule gives a different label when the input changes.

The Mechanics of Machine Learning

What Happens Exactly at Zero?

$$\mathbf{x} = (4, 3)$$

$$z = 3 \times 4 + 4 \times 3 - 24 = 0$$

The rule uses $z \geq 0$ for the +1 class.

Therefore, a score of exactly zero is assigned to +1.

This is a boundary convention in the model; zero is not a third class.

At the threshold, use the inequality written in the model.

The Mechanics of Machine Learning

Change One Input at a Time

Fix sleep at 4 hours. Change only the study time.



Input (sleep, study)	Score z	Prediction
(4, 1)	-8	-1
(4, 2)	-4	-1
(4, 3)	0	+1
(4, 4)	4	+1

One extra study hour adds 4 to this model's score.

Holding one coordinate fixed makes the other coordinate's effect easy to see.

The Mechanics of Machine Learning

What Do the Weights Mean Here?

SLEEP WEIGHT = 3

Holding study time fixed, adding 1 to x_1 increases z by 3.

This describes the model's arithmetic.

STUDY WEIGHT = 4

Holding sleep time fixed, adding 1 to x_2 increases z by 4.

It is not proof of a causal effect on a real exam.

$$z = 3x_1 + 4x_2 - 24$$

A weight controls an input's contribution to this model's score.

The Mechanics of Machine Learning

What Does the Bias Change?

For $x = (4, 3)$, the weighted input sum is 24. $z = 3x_1 + 4x_2 - 24$

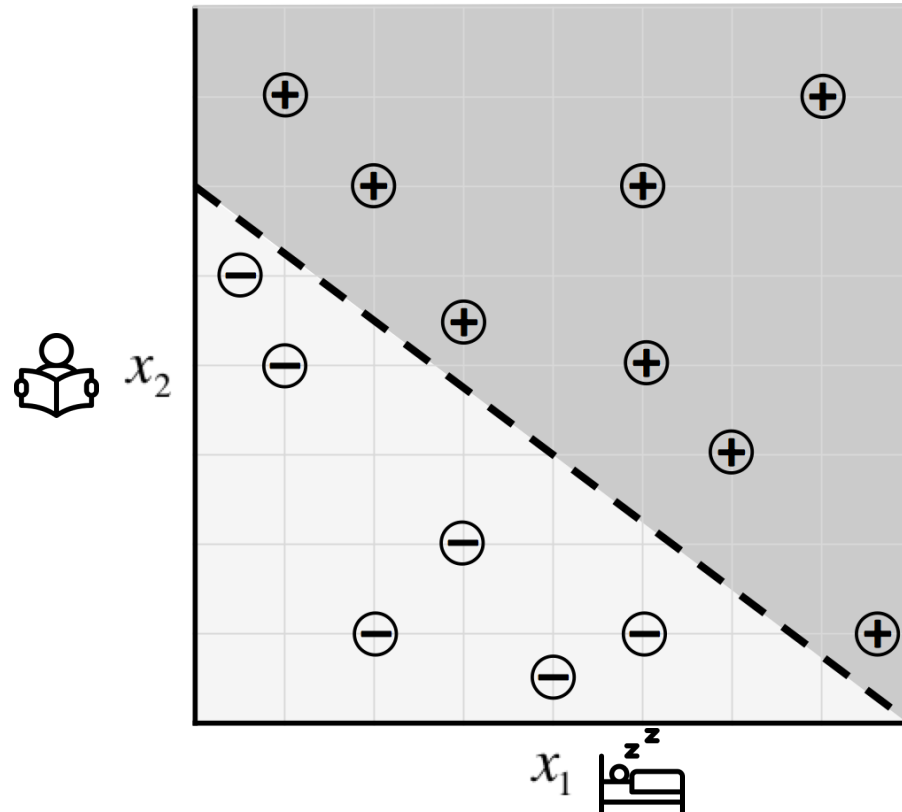
Bias value	Score	Predicted label
-20	$24 - 20 = 4$	+1
-24	$24 - 24 = 0$	+1
-28	$24 - 28 = -4$	-1

The input weights stay fixed; only the constant offset changes.

A bias shifts the score before the threshold is applied.

The Mechanics of Machine Learning

Classifier Diagram



Each plotted sample has a sleep coordinate x_1 and a study coordinate x_2 .

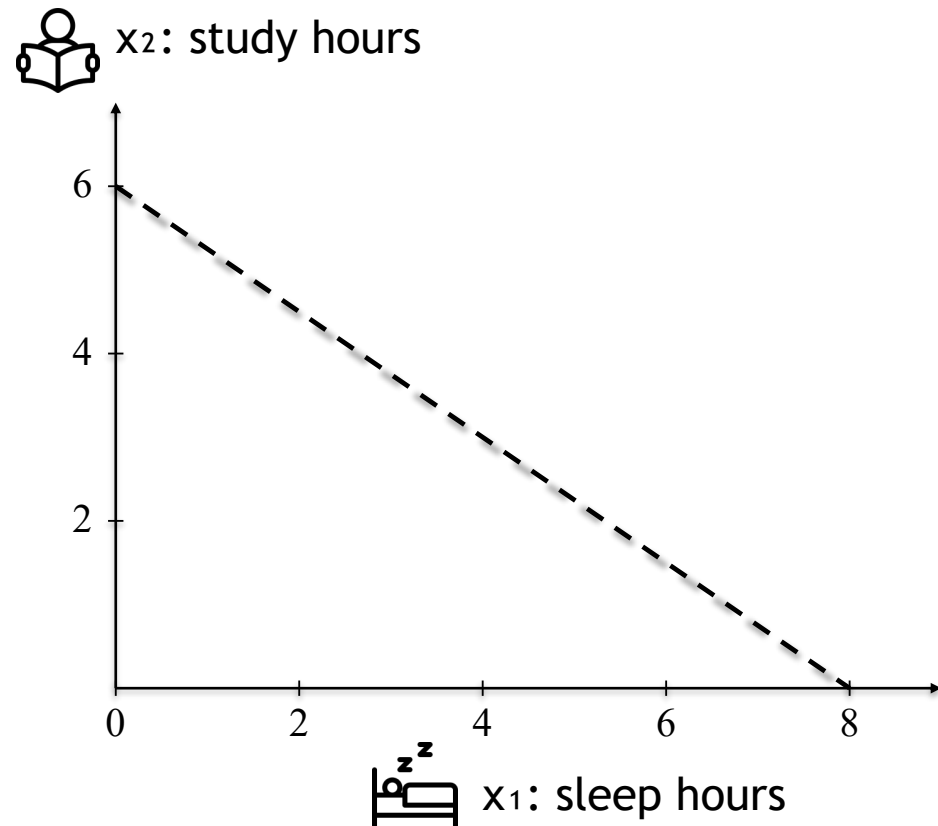
The plus and minus signs indicate the two observed classes. PASS FAIL

The dashed line is a possible classifier separating the samples in this illustration.

The same prediction rule can be interpreted as a boundary in the input plane.

The Mechanics of Machine Learning

Why Is the Boundary a Straight Line?



$$3x_1 + 4x_2 - 24 = 0$$

The decision boundary contains the inputs whose score is exactly zero.

If $x_1 = 0$, then $x_2 = 6$.

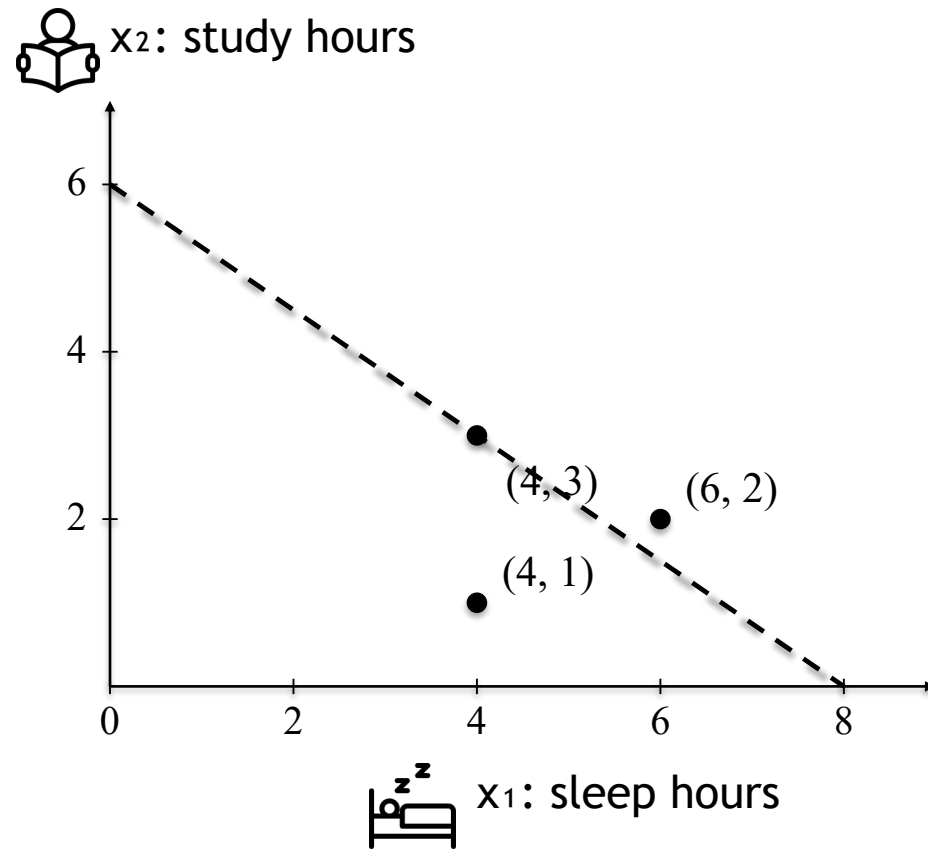
If $x_2 = 0$, then $x_1 = 8$.

These two intercepts locate the straight line.

The boundary is the set of inputs with score zero.

The Mechanics of Machine Learning

Connect the Numbers to the Geometry



$(6, 2)$ gives $z = 2$,
so it is assigned to $+1$.

$(4, 1)$ gives $z = -8$,
so it is assigned to -1 .

$(4, 3)$ lies on the boundary and is
assigned to $+1$ by the ≥ 0 rule.

The sign of the score tells us which side of the boundary an input lies on.

The Mechanics of Machine Learning

Changing x Is Not the Same as Changing θ

CHANGE THE INPUT

Keep $\theta = (-24, 3, 4)$.

Test another sleep-and-study pair.

The classifier stays fixed.

CHANGE THE PARAMETERS

Keep the recorded examples.

Try different weights or a different bias.

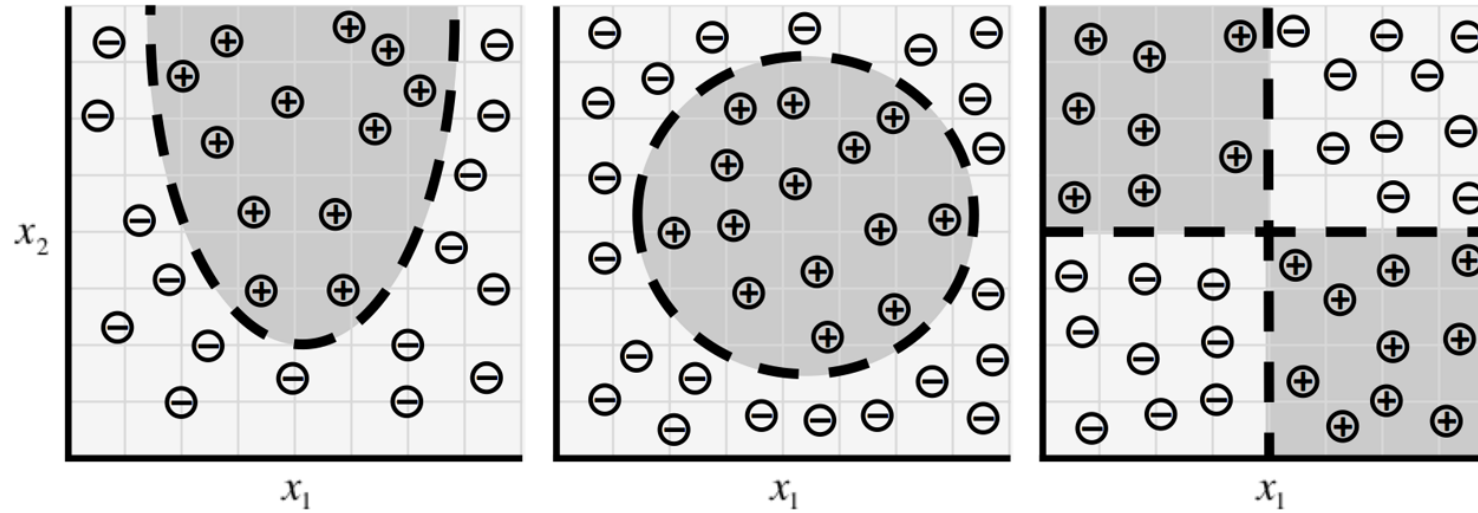
The classifier itself changes.

$$z = 3x_1 + 4x_2 - 24$$

Inputs move the point; parameter changes move or rotate the decision boundary.

The Mechanics of Machine Learning

When One Line Is Not Enough



The patterns need curved, closed, or multiple-region boundaries.

A single straight boundary cannot match all of these shapes.

A model's form limits which relationships its parameters can represent.

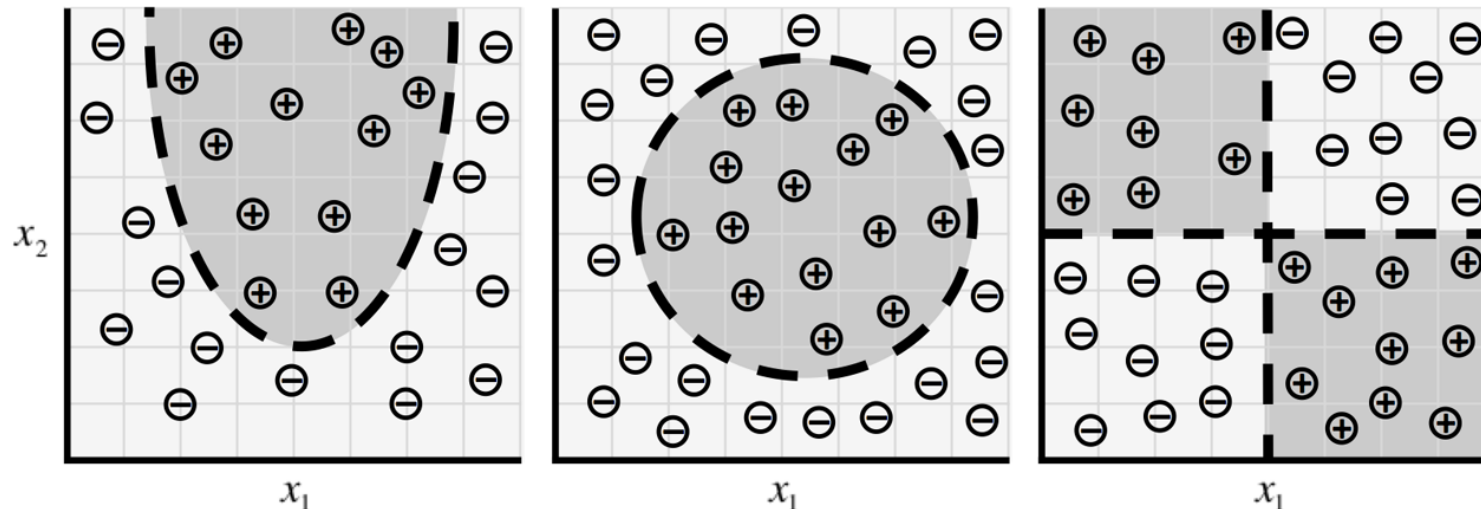
The Mechanics of Machine Learning

Why Parameter Tuning Cannot Fix Every Pattern

Changing θ changes the location and orientation of a linear boundary.

It does not turn that one line into a circle or a set of separated regions.

For more complex relationships, the chapter introduces neurons and then networks of neurons.



Better parameters cannot remove a limitation of the model's structure.

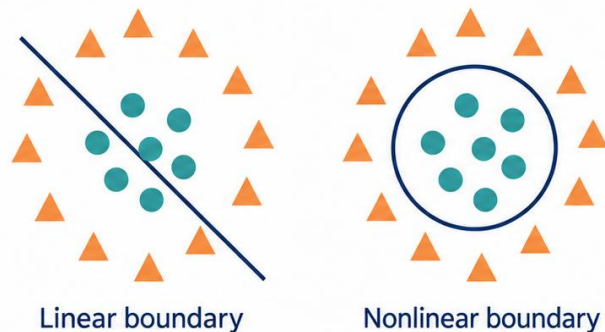
The Mechanics of Machine Learning

Two Different Questions

WHAT CAN THE MODEL EXPRESS?

Choose a structure that can represent the relationship.

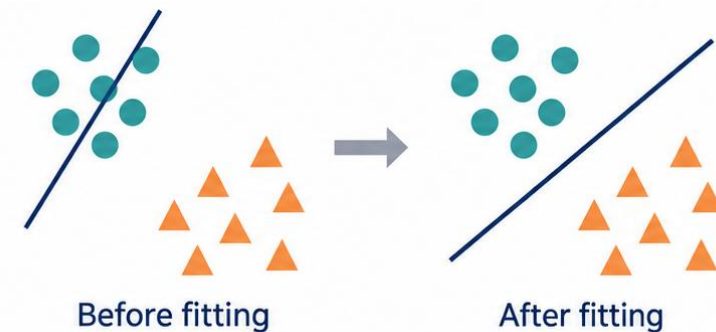
This is the focus of the neuron and network discussion.



HOW DO WE FIND PARAMETERS?

Adjust the model's numerical settings to reduce prediction errors.

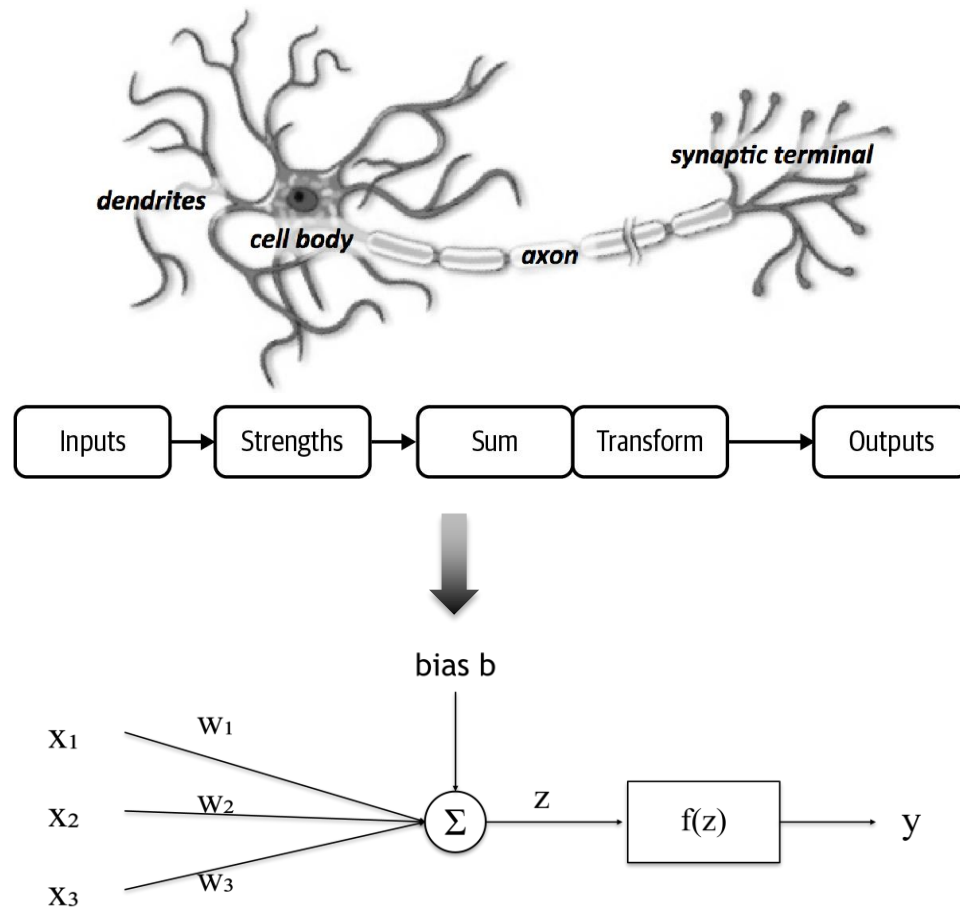
We will discuss optimization and gradient next week.



This chapter builds the model; the next chapter explains how to fit it.

The Neuron

A Biological Analogy



The biological diagram motivates a sequence: receive inputs, weight them, combine them, and transmit an output.

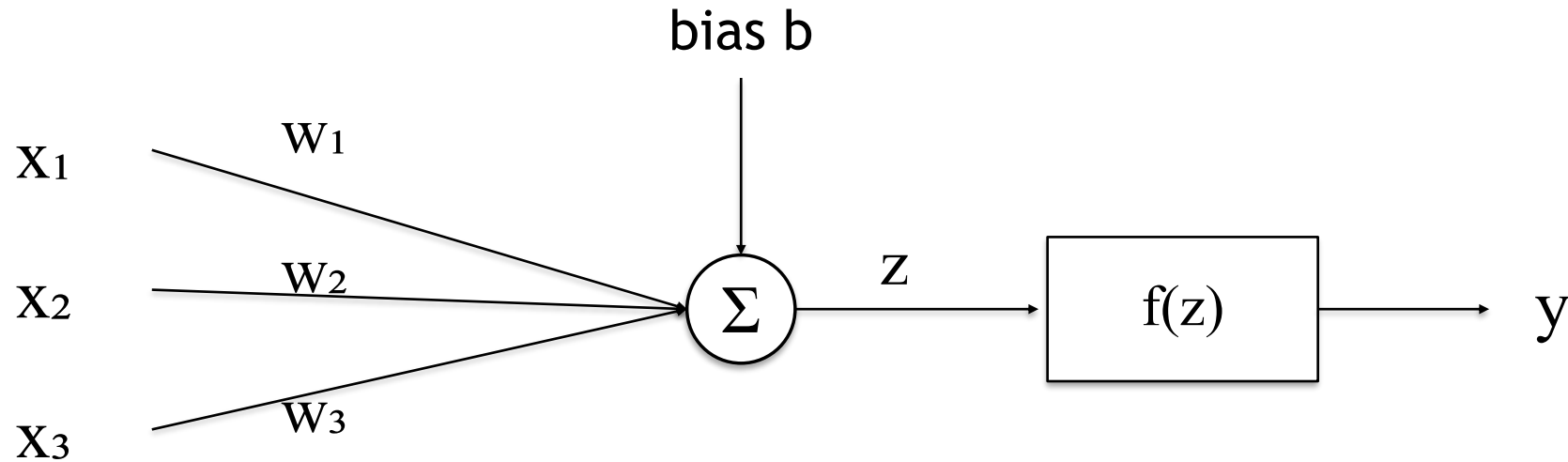
The artificial neuron will use a simplified mathematical version of this sequence.

The analogy motivates the model; it is not a complete simulation of a brain cell.

The useful computational idea is weighted inputs followed by a transformation.

The Neuron

The Artificial Neuron



Multiply each input by its weight, then add the contributions and bias.

The score z passes through f to produce the output y .

Separate the weighted score z from the transformed output y .

The Neuron

Five Symbols, Five Different Jobs

Symbol	Plain meaning
x_i	An input value from the current example.
w_i	The weight multiplying that input.
b	A constant added to the weighted sum.
z	The score before applying the activation.
$y = f(z)$	The output after applying the activation.

The input, weight, bias, score, and output should not be confused.

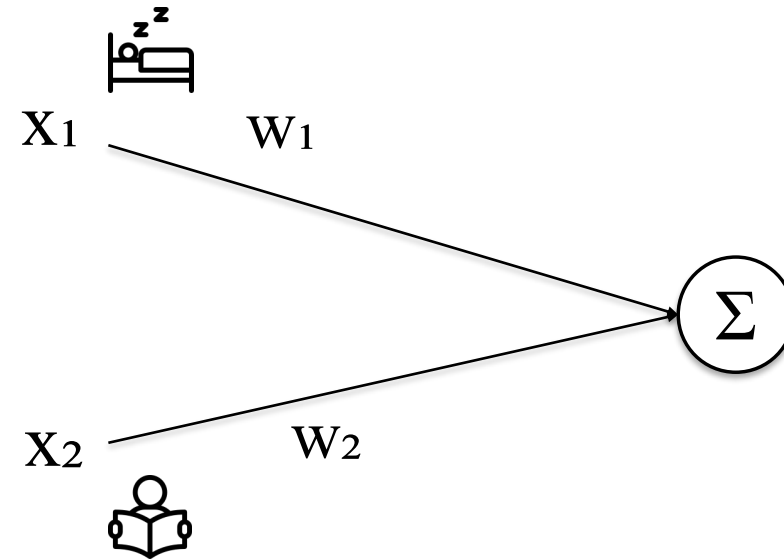
The Neuron

Step 1: Form the Weighted Sum

$$\mathbf{x} = (6, 2), \quad \mathbf{w} = (3, 4)$$

$$\mathbf{x} \cdot \mathbf{w} = 6 \times 3 + 2 \times 4$$

$$= 18 + 8 = 26$$



This is the dot product from the earlier linear-algebra lesson.

We have combined the inputs, but have not yet added the bias or applied f .

The dot product adds the contribution of every input.

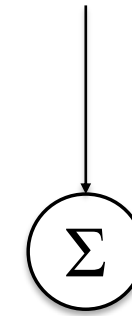
The Neuron

Step 2: Add the Bias

$$z = x \cdot w + b$$

$$z = 26 + (-24) = 2$$

bias b



The bias shifts the weighted sum by a fixed amount.

The book calls this pre-activation value z the logit.

The score 2 is not yet the class label. We still need the activation function.

A raw score becomes an output only after the chosen transformation is applied.

The Neuron

Step 3: Apply the Activation

THE ACTIVATION RULE

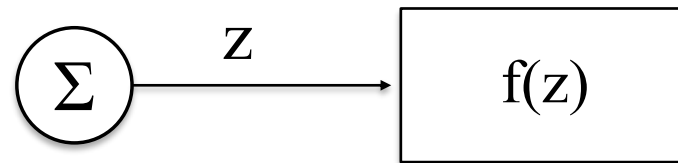
If $z < 0$, $f(z) = -1$. 

If $z \geq 0$, $f(z) = +1$. 

APPLY THE RULE

For our score $z = 2$, $f(2) = +1$.

For the earlier score $z = -8$, $f(-8) = -1$.



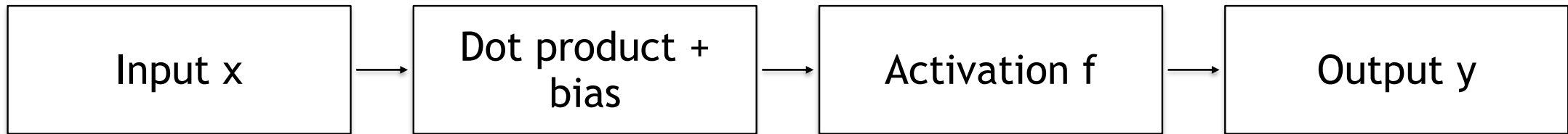
The same activation rule is used for every input to this model.

Here, f turns a numerical score into one of two labels.

The Neuron

Write the Whole Neuron in One Formula

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



The expression inside the parentheses is calculated first.

Then f acts on that single score.

The formula is a compact description of the same three steps.

Read the formula from the inner computation to the final output.

The Neuron

Expand the Compact Notation

$$\mathbf{x} \cdot \mathbf{w} + \mathbf{b} = x_1 w_1 + x_2 w_2 + \mathbf{b}$$

$$= 3x_1 + 4x_2 - 24$$

The compact dot-product notation does not change the arithmetic.

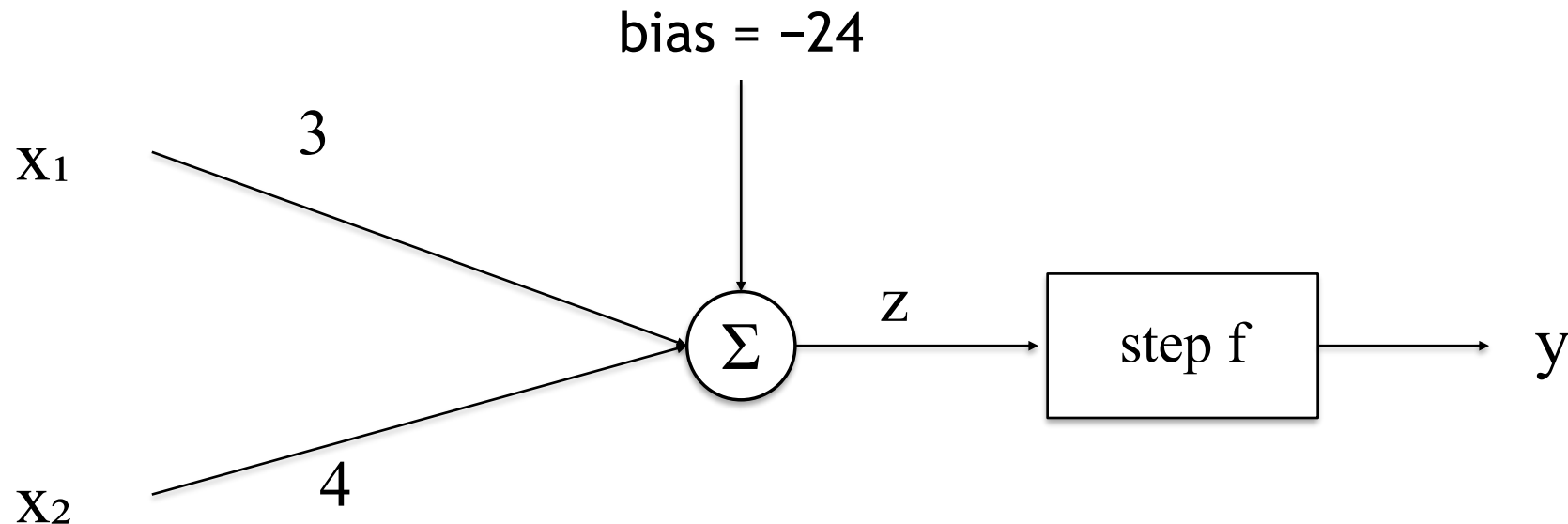
It simply collects the input values in $\mathbf{x}$ and the matching weights in $\mathbf{w}$.

For this example, $\mathbf{w} = (3, 4)$ and $\mathbf{b} = -24$.

The vector formula and the two-input calculation are the same model.

The Neuron

Expressing Linear Perceptrons as Neurons



Weights: 3 for sleep and 4 for study. Bias: -24.

The step function returns -1 for $z < 0$ and +1 for $z \geq 0$.

The earlier exam classifier is exactly a neuron with this step activation.

The Neuron

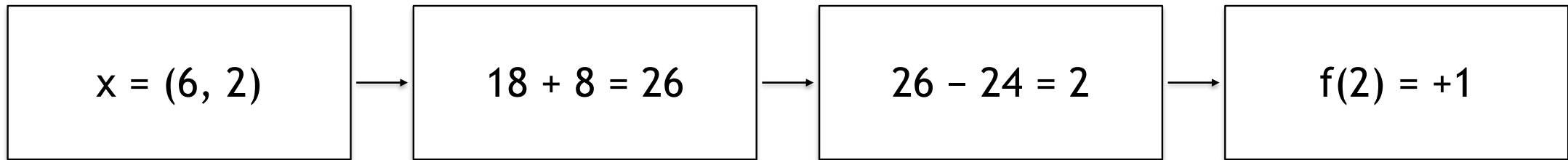
Match the Old Symbols to the New Ones

Earlier perceptron	Neuron notation
$\theta_1 = 3$ and $\theta_2 = 4$	$w = (3, 4)$
$\theta_0 = -24$	$b = -24$
$3x_1 + 4x_2 - 24$	$z = x \cdot w + b$
Choose -1 or $+1$ from the sign	$y = f(z)$, using the step function

The notation changes, but the predictions do not.

The Neuron

Trace One Input All the Way Through



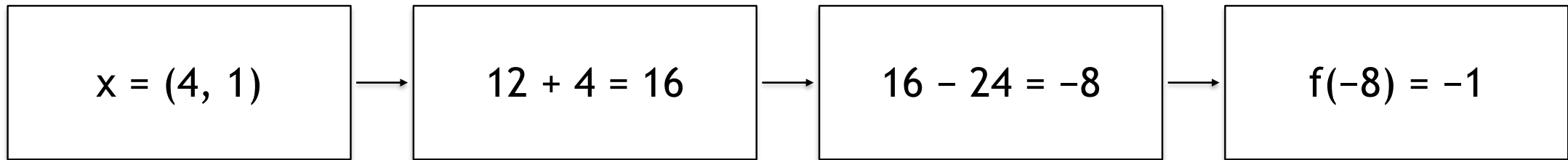
Read from left to right: input, weighted sum, score, and class label.

The last number is a prediction of the category, not a predicted exam mark.

Following a single example makes every intermediate value visible.

The Neuron

Now Trace the Other Input



Only the input changes. The weights, bias, and activation remain the same.

The different score leads the same model to choose the other class.

A fixed model can produce different outputs for different inputs.

The Neuron

What We Have Built—and What Comes Next

With the step activation, the neuron reproduces our linear perceptron exactly.

A different activation can change the neuron's output behavior.

The chapter next connects multiple neurons into layers to build richer models.

A single neuron is a building block,
not a complete solution to every recognition task.

Summary

Quick Check

1. In $x = (6, 2)$, which value is sleep time and which is study time?
2. For $z = 3x_1 + 4x_2 - 24$, what label is selected when $z = 0$?
3. Does changing the input x mean that the model has learned new parameters?

Explain what each intermediate value means, not just how to calculate it.

Summary

Check: Explanations

1. Read the input convention

The first coordinate records sleep; the second records study.

2. Read the inequality

At $z = 0$, the nonnegative branch selects +1.

3. Separate prediction from learning

Prediction computes with current parameters. Learning adjusts those parameters.

The input, parameters, score, and label have distinct roles.

Summary

From Examples to a Single Neuron

Recognition problem

Handwritten rules become difficult when input appearances vary.

Parameterized model

Use $h(x, \theta)$, then interpret the exam perceptron as a weighted score and threshold.

Neuron and next step

Compute $z = x \cdot w + b$ and $y = f(z)$. Next, connect neurons into layers.

Examples define the task; a neuron turns numerical inputs into an output.