Neural Networks from Math

Vu Hung Nguyen

2025-11-03

Contents

Preface

This brief book takes an example-first path from familiar school mathematics to the core ideas behind a simple neural network architecture. It is designed for motivated high-school students: short chapters, visual intuition, and concrete checkpoints.

Chapter 1

Overview

This book takes you from familiar mathematics to a working understanding of simple neural networks. It is written in clear British English, and designed for undergraduates and motivated high-school students.

Learning Objectives

1. Understand the book's roadmap and how chapters connect.
2. Recognise the key ingredients of a neural network.
3. Learn how examples, metaphors, and visuals support intuition.

Roadmap

We move from functions and graphs to data and error, then to vectors and matrices that compactly represent many inputs. Linear models give us a first predictive rule; nonlinear activations add the "kinks" that let us model more interesting patterns. Stacking layers composes these ideas. Finally, we discuss loss and optimisation, the training loop, a simple network, and practical limits and ethics.

Example 1.1

Consider predicting house price from size. A function maps size (input) to price (output). A linear model draws a straight line; a neural network allows bends via activations and layers, improving fit when reality is not a straight line.

Remark 1.1

Throughout, we prioritise intuition first, then formalism. Visuals accompany core definitions where helpful.

How to Use This Book

Skim the learning objectives at the start of each chapter. Work through examples; attempt exercises before revealing the upside-down hints. Use the glossary to refresh key terms.

Chapter 2

Functions and Graphs

Functions describe how inputs map to outputs. Graphs help us see this mapping.

Learning Objectives

1. Interpret a function as a rule from input to output.
2. Read tables and plots to understand trends.
3. Distinguish linear from nonlinear patterns visually.

2.1 From Tables to Rules

Suppose we record study time (hours) and score (percentage). A function f turns an input x into an output $y = f(x)$. A straight trend suggests a linear rule; bends suggest nonlinearity.

Definition 2.1

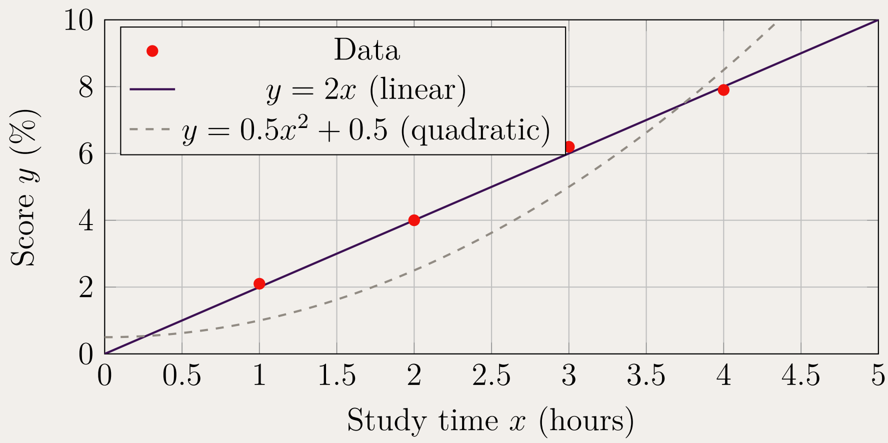
A *function* is a mapping assigning each input a single output. We often visualise $(x, f(x))$ as points on a graph.

Example 2.1

Measured study time (hours) and score (%):

x	1	2	3	4
y	2.1	4.0	6.2	7.9

If doubling x roughly doubles y, a straight line is sensible. Below we plot the points, a simple linear rule $y = 2x$, and a curved (quadratic) rule $y = 0.5x^2 + 0.5$ to contrast linear vs nonlinear behaviour.



The linear rule tracks the trend closely; the quadratic bends upward and will diverge from the roughly proportional pattern as x grows.

2.2 Exercises

Given points $(1, 2), (2, 4.1), (3, 5.9)$, would a straight line be a reasonable model? Explain briefly.

Hint: Turn the page upside down to readYes. The ratios are consistent with a near-linear trend; noise causes small deviations.

Chapter 3

Data and Error

Predictions meet reality through data. Error tells us how far off we are.

Learning Objectives

1. Define prediction, target, and error for a dataset.
2. Compute simple average absolute and squared error.
3. Explain why averaging error stabilises noisy measurements.

3.1 Measuring Error

Given inputs x_i with observed outputs y_i and predictions $\hat{y}_i$, an error for item i is $e_i = \hat{y}_i - y_i$. Two popular summaries are the mean absolute error (MAE) and mean squared error (MSE):

$$\mathrm{MAE} = \frac{1}{n}\sum_i |e_i|, \quad \mathrm{MSE} = \frac{1}{n}\sum_i e_i^2.$$

Example 3.1

Using the study–score data from the previous chapter

x	1	2	3	4
y	2.1	4.0	6.2	7.9

consider the linear rule $\hat{y} = 2x$. The predictions and errors are:

x	1	2	3	4
$\hat{y} = 2x$	2.0	4.0	6.0	8.0
$e = \hat{y} - y$	0.1	0.0	-0.2	0.1
$\lvert e \rvert$	0.1	0.0	0.2	0.1
e^2	0.01	0.00	0.04	0.01

Thus

$$\mathrm{MAE} = \tfrac{0.1+0.0+0.2+0.1}{4} = 0.1, \qquad \mathrm{MSE} = \tfrac{0.01+0.00+0.04+0.01}{4} = 0.015.$$

The small MAE/MSE values indicate the line $\hat{y} = 2x$ matches the trend closely for these points.

Remark 3.1

Squaring emphasises large mistakes; absolute value treats all deviations proportionally. Choice depends on the application.

3.2 Exercises

For true values $y = (1, 2, 3)$ and predictions $\hat{y} = (1.2, 1.9, 3.4)$, compute MAE and MSE.

Hint: Errors: $(0.2, -0.1, 0.4)$. MAE $= (0.2 + 0.1 + 0.4)/3 = 0.233\ldots$; MSE $= (0.04 + 0.01 + 0.16)/3 = 0.07\ldots$.

Chapter 4

Vectors and Matrices

Vectors collect features; matrices collect many vectors and define linear transformations.

Learning Objectives

1. Represent data points as vectors of features.
2. Use matrix–vector multiplication to combine features linearly.
3. Interpret a matrix as a transformation of space.

4.1 Notation

Write a feature vector as $\boldsymbol{x} = (x_1, \ldots, x_d)^\top$. A weight vector $\boldsymbol{w}$ and bias b define a linear score $s = \boldsymbol{w}^\top \boldsymbol{x} + b$.

Definition 4.1

A *matrix* $W \in \mathbb{R}^{m \times d}$ applied to a vector $\boldsymbol{x} \in \mathbb{R}^d$ yields $W\boldsymbol{x} \in \mathbb{R}^m$, combining inputs into m outputs.

Example 4.1

With two features (height, width) and two outputs (sum, difference), $W = \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}$ maps a vector $\boldsymbol{x} = (x_1, x_2)^\top$ to $W\boldsymbol{x} = (x_1 + x_2,\ x_1 - x_2)^\top$.
For a concrete calculation, let $\boldsymbol{x} = (3, 2)^\top$. Then

$$W\boldsymbol{x} = \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \begin{bmatrix} 3 \\ 2 \end{bmatrix} = \begin{bmatrix} 1 \cdot 3 + 1 \cdot 2 \\ 1 \cdot 3 + (-1) \cdot 2 \end{bmatrix} = \begin{bmatrix} 5 \\ 1 \end{bmatrix}.$$

So the first output (sum) is 5 and the second output (difference) is 1.

4.2 Exercises

If $W = \begin{bmatrix} 2 & 0 \\ 0 & 3 \end{bmatrix}$ and $\boldsymbol{x} = (1, 2)^\top$, compute $W\boldsymbol{x}$.

Hint: $(2, 6)^\top$.

Chapter 5

Linear Models

A linear model scores inputs by a weighted sum plus a bias; decisions follow from thresholds or regression.

Learning Objectives

1. Write a linear model as $\hat{y} = \boldsymbol{w}^\top \boldsymbol{x} + b$.
2. Explain the geometric view: lines and hyperplanes.
3. Recognise when linear models are insufficient.

5.1 Form and Geometry

The set of points with constant score forms a hyperplane. When classes are linearly separable, a single hyperplane can divide them.

Example 5.1

Classify points above a line: take $\hat{y} = 2x_1 - x_2 + 0.5$ and predict positive when $\hat{y} > 0$.
For $(x_1, x_2) = (2, 1)$,

$$\hat{y} = 2 \cdot 2 - 1 + 0.5 = 3.5 > 0 \;\Rightarrow\; \text{positive.}$$

For $(x_1, x_2) = (0, 1)$,

$$\hat{y} = 2 \cdot 0 - 1 + 0.5 = -0.5 < 0 \;\Rightarrow\; \text{negative.}$$

The decision boundary $\hat{y} = 0$ is the straight line $x_2 = 2x_1 + 0.5$.

5.2 Exercises

Find a line that separates points A: $(0, 2), (1, 3)$ from B: $(2, 0), (3, 1)$ if possible.

Hint: Try $x_2 = x_1 + 1.5$. Points A lie above, B below.

Chapter 6

Nonlinearity and Activations

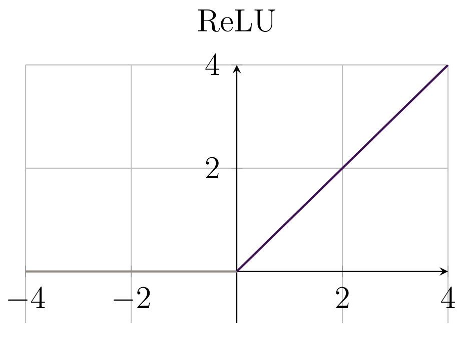
Nonlinear activations introduce bends that let models capture curved patterns.

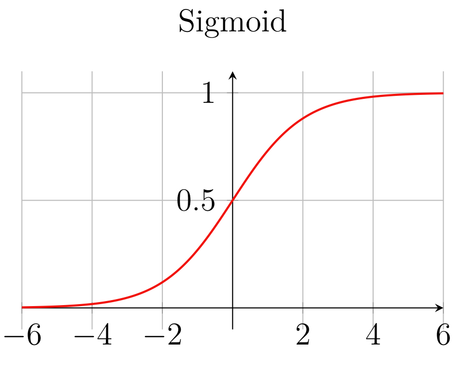
Learning Objectives

1. Describe common activations: ReLU, sigmoid, tanh.
2. Explain why stacking linear layers without activations stays linear.
3. Match activations to tasks (e.g., sigmoid for probabilities).

6.1 Common Activations

ReLU: $\mathrm{ReLU}(z) = \max(0, z)$ adds a kink at zero. Sigmoid $\sigma(z) = 1/(1 + e^{-z})$ squashes to $(0, 1)$. Tanh squashes to $(-1, 1)$.





Why this matters. Activations create nonlinearity so networks can approximate curved relationships. ReLU is simple and keeps gradients flowing for positive inputs, aiding optimisation. Sigmoid maps real numbers to $(0, 1)$, ideal for probabilities; but it can saturate (flat tails), slowing learning when inputs are very large in magnitude.

Bio example (sigmoid). The logistic (sigmoid) curve models population growth with carrying capacity and neuron firing rates as a function of membrane potential: response is low for small input, steep near threshold, and saturates at high input.

Remark 6.1

Linear ∘ linear is still linear. Nonlinearity between linear steps is essential.

6.2 Exercises

Sketch and compare $\tanh(z)$ and leaky ReLU $\mathrm{LReLU}_\alpha(z) = \max(\alpha z, z)$ with $\alpha = 0.1$. Where are they most/least sensitive, and what ranges do their outputs take?

Hint: ReLU is flat for $z < 0$ and slope 1 for $z > 0$. Sigmoid is steepest near 0, flat in tails.

Chapter 7

Composing Layers

Stack linear steps with activations to form complex functions.

Learning Objectives

1. Define a layer as $\boldsymbol{h} = g(W\boldsymbol{x} + \boldsymbol{b})$.
2. Explain how two layers can form piecewise linear curves.
3. Understand hidden units as learned features.

7.1 Two-Layer Example

Let $\boldsymbol{h} = \mathrm{ReLU}(W_1\boldsymbol{x} + \boldsymbol{b}_1)$ and $\hat{y} = W_2\boldsymbol{h} + b_2$. Even with ReLU, the output is a flexible piecewise linear function of $\boldsymbol{x}$.

Example 7.1

In one dimension, two ReLUs can create a "bump" by combining kinks at different locations. Consider a tiny network with one input x, two hidden units, and one output:

$$\boldsymbol{h} = \mathrm{ReLU}(W_1 x + \boldsymbol{b}_1), \qquad \hat{y} = W_2\, \boldsymbol{h} + b_2,$$

where

$$W_1 = \begin{bmatrix} 1 \\ -1 \end{bmatrix},\ \boldsymbol{b}_1 = \begin{bmatrix} -1 \\ -3 \end{bmatrix},\ W_2 = \begin{bmatrix} 1 & -1 \end{bmatrix},\ b_2 = 0.$$

Thus the hidden units are

$$h_1 = \mathrm{ReLU}(x-1), \qquad h_2 = \mathrm{ReLU}(-x-3) = \mathrm{ReLU}(-(x+3)).$$

The output becomes $\hat{y} = h_1 - h_2$. Evaluate at a few inputs:

x	$h_1 = \max(0, x-1)$	$h_2 = \max(0, -x-3)$	$\hat{y} = h_1 - h_2$
-5	0	2	-2
-3	0	0	0
0	0	0	0
2	1	0	1
4	3	0	3

Interpretation:

- For $x \le -3$, only the second unit is active, so $\hat{y} = -h_2$ decreases linearly.
- For $-3 < x < 1$, both units are off, so $\hat{y} = 0$.
- For $x \ge 1$, only the first unit is active, so $\hat{y} = h_1$ increases linearly.

This piecewise linear shape forms a flat region with rising and falling flanks—one way to build a "bump" by composing ReLUs.

7.2 Exercises

How many linear regions can a sum of two shifted ReLUs create on the line?

Hint: Up to three regions separated by the two kink locations.

Chapter 8

Loss and Optimisation

Loss quantifies mismatch between predictions and targets; optimisation reduces loss.

Learning Objectives

1. Define a loss function suitable for a task.
2. Describe gradient descent at a high level.
3. Relate learning rate to step size and stability.

8.1 Loss Choices

For regression, MSE is common; for binary classification, logistic loss pairs well with sigmoid outputs.

Example 8.1: Regression loss (MSE)

Suppose a model predicts house prices (in thousands) for three homes: $\hat{y} = (210, 195, 250)$, and true prices are $y = (200, 205, 240)$. Errors are $e = \hat{y} - y = (10, -10, 10)$. The mean squared error is

$$\mathrm{MSE} = \tfrac{1}{3}(10^2 + (-10)^2 + 10^2) = \tfrac{1}{3}(100 + 100 + 100) = 100.$$

Units are squared (here, thousands^2). A smaller value indicates closer predictions.

Example 8.2: Logistic loss

For a binary label $y \in \{0, 1\}$ with sigmoid output $p = \sigma(z)$ interpreted as $\Pr(y = 1 \mid x)$, the logistic loss for a single example is

$$\ell(y, p) = -\Big(y \, \log p + (1 - y) \, \log(1 - p)\Big).$$

Consider two cases with $p = 0.9$:

- If $y = 1$: $\ell = -(1 \cdot \log 0.9 + 0 \cdot \log 0.1) \approx 0.105$ (small penalty).
- If $y = 0$: $\ell = -(0 \cdot \log 0.9 + 1 \cdot \log 0.1) \approx 2.303$ (large penalty for confident wrong prediction).

This asymmetry encourages calibrated probabilities: confident and correct is rewarded; confident and wrong is penalised heavily.

8.2 Gradient Descent (Conceptually)

Imagine standing on a landscape where height is loss. At each step, move a little in the downhill direction; repeat until you are low enough.

Example 8.3: One variable

Let $f(x) = (x - 3)^2$. Its derivative is $f'(x) = 2(x - 3)$. Starting at $x_0 = 0$ with learning rate $\eta = 0.5$, gradient descent updates are

$$x_{k+1} = x_k - \eta \, f'(x_k) = x_k - 0.5 \cdot 2(x_k - 3) = 3 - (x_k - 3).$$

Numerically: $x_1 = 1.5$, $x_2 = 2.25$, $x_3 = 2.625 \ldots$ which approaches the minimiser $x^* = 3$.

Example 8.4: Two variables

Let $f(x, y) = (x - 1)^2 + (y + 2)^2$. The gradient is $\nabla f = (2(x - 1), \, 2(y + 2))$. With $(x_0, y_0) = (0, 0)$ and $\eta = 0.25$:

$$(x_1, y_1) = (0, 0) - 0.25 \, (2(-1), \, 2(2)) = (0.5, \, -1).$$

Next step uses the new gradient: $\nabla f(x_1, y_1) = (2(-0.5), \, 2(1)) = (-1, 2)$, thus

$$(x_2, y_2) = (0.5, -1) - 0.25 \, (-1, 2) = (0.75, \, -1.5).$$

The iterates head toward the minimiser $(1, -2)$.

Many inputs (neural nets). For networks, parameters form a long vector $\boldsymbol{\theta}$ (all weights and biases). The loss $L(\boldsymbol{\theta})$ is averaged over a mini-batch. We compute the gradient $\nabla L(\boldsymbol{\theta})$ efficiently by backpropagation and update

$$\boldsymbol{\theta}_{k+1} = \boldsymbol{\theta}_k - \eta \, \nabla L(\boldsymbol{\theta}_k).$$

Different layers receive different components of the same gradient, moving all parameters a little in the direction that most reduces loss on the current batch.

If steps are too large, what failure can occur?

Hint: You can overshoot and bounce around (diverge) instead of settling.

Chapter 9

Training Loop Concepts

Predict, measure, adjust, repeat. Split data to measure progress fairly.

Learning Objectives

1. Describe the predict–loss–update loop.
2. Explain train/validation/test splits.
3. Recognise overfitting and how validation helps.

9.1 The Loop

For each mini-batch: compute predictions, compute loss, update parameters a little. After each epoch, check validation loss.

Example 9.1

Consider a simple linear model $\hat{y} = wx + b$ trained with mean squared error on one mini-batch of two points: $(x, y) \in \{(1, 2), (2, 4)\}$. Start from $w_0 = 1.0$, $b_0 = 0.0$, learning rate $\eta = 0.1$.

Predictions. With $(w, b) = (1, 0)$, predictions are $\hat{y} = (1, 2)$. Errors $e = \hat{y} - y = (-1, -2)$.

Loss. $\text{MSE} = \frac{1}{2}((-1)^2 + (-2)^2) = \frac{1}{2}(1 + 4) = 2.5$.

Gradients. For MSE with this batch,

$$\frac{\partial L}{\partial w} = \frac{1}{2} \sum 2e_i\, x_i = e_1 x_1 + e_2 x_2 = (-1) \cdot 1 + (-2) \cdot 2 = -5,$$

$$\frac{\partial L}{\partial b} = \frac{1}{2} \sum 2e_i = e_1 + e_2 = -3.$$

Update. Gradient descent gives

$$w_1 = w_0 - \eta \frac{\partial L}{\partial w} = 1 - 0.1 \cdot (-5) = 1.5, \qquad b_1 = b_0 - \eta \frac{\partial L}{\partial b} = 0 - 0.1 \cdot (-3) = 0.3.$$

Repeating on the next mini-batch (or another epoch) will continue to reduce loss; validation loss is checked after full-dataset passes to monitor overfitting.

Remark 9.1

When validation loss rises while training loss falls, you may be overfitting.

9.2 Exercises

Why is a separate test set needed if validation loss looks good?

Hint: To estimate performance on truly unseen data and avoid tuning to the validation set.

Chapter 10

A Simple Neural Network Architecture

A minimal feed-forward network: inputs, a hidden layer, and an output.

Learning Objectives

1. Sketch a tiny network with one hidden layer.
2. Track shapes through layers.
3. Explain forward pass at a high level.

10.1 Shapes and Flow

With d inputs, h hidden units, and one output: $W_1 \in \mathbb{R}^{h \times d}$, $\boldsymbol{b}_1 \in \mathbb{R}^h$, activation g; then $W_2 \in \mathbb{R}^{1 \times h}$, $b_2 \in \mathbb{R}$.

Example 10.1

Let $d = 2, h = 3$. Then W_1 has 3 rows and 2 columns; W_2 has 1 row and 3 columns. Compute $\boldsymbol{h} = g(W_1\boldsymbol{x} + \boldsymbol{b}_1)$, then $\hat{y} = W_2\boldsymbol{h} + b_2$.

Take a numerical instance with ReLU activation $g = \mathrm{ReLU}$:

$$\boldsymbol{x} = \begin{bmatrix} 2 \\ 1 \end{bmatrix}, \quad W_1 = \begin{bmatrix} 1 & -1 \\ 0.5 & 1 \\ 2 & 0 \end{bmatrix}, \quad \boldsymbol{b}_1 = \begin{bmatrix} 0 \\ -1 \\ 0.5 \end{bmatrix}.$$

First layer pre-activation and activation:

$$\boldsymbol{z}_1 = W_1\boldsymbol{x}+\boldsymbol{b}_1 = \begin{bmatrix} 1 & -1 \\ 0.5 & 1 \\ 2 & 0 \end{bmatrix} \begin{bmatrix} 2 \\ 1 \end{bmatrix} + \begin{bmatrix} 0 \\ -1 \\ 0.5 \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \\ 4.5 \end{bmatrix}, \qquad \boldsymbol{h} = g(\boldsymbol{z}_1) = \mathrm{ReLU}(\boldsymbol{z}_1) = \begin{bmatrix} 1 \\ 1 \\ 4.5 \end{bmatrix}.$$

Output layer parameters and prediction:

$$W_2 = \begin{bmatrix} 1 & -2 & 0.5 \end{bmatrix}, \quad b_2 = 0.2, \qquad \hat{y} = W_2\boldsymbol{h}+b_2 = 1\cdot1+(-2)\cdot1+0.5\cdot4.5+0.2 = 1.45.$$

Thus with this small network and input $\boldsymbol{x} = (2, 1)^\top$, the hidden representation is $\boldsymbol{h} = (1, 1, 4.5)^\top$ and the scalar output is $\hat{y} = 1.45$.

10.2 Exercises

If $g = \mathrm{ReLU}$ and $\boldsymbol{h} = (0, 2, 0)^\top$, what is $\hat{y}$ when $W_2 = (1, 1, 1)$ and $b_2 = 0$?

Hint:2.

Chapter 11

Limits and Ethics

Models have limits. Responsible practice matters.

Learning Objectives

1. Identify common failure modes: overfitting, data shift, bias.
2. Describe simple mitigations students can apply.
3. Reflect on responsible communication of model results.

11.1 Limits

Overfitting memorises training noise; distribution shift breaks assumptions; biased data leads to unfair outcomes.

Example 11.1: Overfitting

Fit a 9th-degree polynomial to 10 noisy points sampled from the straight line $y = 2x$. On the training points, the curve can wiggle to achieve near-zero error; on fresh test points from the same line, error is high because the wiggles chase noise rather than signal.

Example 11.2: Underfitting

Fit a straight line to data generated by a clear U-shaped quadratic like $y = x^2$ with small noise. Both training and test errors remain large because a line cannot capture the curved relationship: the model is too simple for the pattern present.

11.2 Ethics

Use diverse, representative data when possible; report uncertainty; avoid over-claiming; consider impacts on people.

Give one reason a model trained on last year's data may underperform this year.

Hint: Data distribution can shift: the relationship between inputs and outputs changes.

Chapter 12

Glossary

Plain-language definitions for quick recall.

Activation A nonlinear function applied to a neuron's input (e.g., ReLU, sigmoid).

Error The difference between a prediction and the true value.

Gradient Descent An iterative method that adjusts parameters to reduce loss.

Loss A number that measures how far predictions are from targets.

Matrix A rectangular array of numbers; represents a linear transformation.

Neural Network A function built by composing linear steps with activations.

Overfitting Learning noise or specifics of training data that do not generalise.

Vector An ordered list of numbers representing features.

Appendix A

Worksheets: Checkpoints

Short comprehension checks per chapter. Replace this text with specific checkpoint prompts when ready.

Appendix B

Worksheets: Exercises

Small numeric and drawing tasks designed for practice without calculus.

References

Bibliography

About the Author

Vu Hung Nguyen is the author of this book. For more information and updates:

- Website: https://vuhung16au.github.io/
- GitHub: https://github.com/vuhung16au/
- LinkedIn: https://www.linkedin.com/in/nguyenvuhung/