What Is Deep Learning?

Deep learning is machine learning with many-layered neural networks. What a neural

Introduction

In the Machine Learning series you learned that a model finds patterns in examples. Deep learning is the same idea, taken further: instead of a simple model like a decision tree, you build a **neural network** — a chain of small computing units arranged in layers. Data goes in one side, passes through layer after layer, and a prediction comes out the other side. Each layer transforms the data a little, so by the last layer the network has built up a rich picture of what it saw.

The word "deep" does not mean clever or mysterious. It means **many layers**. A network with one or two layers is "shallow". A network with ten, fifty or a hundred layers is "deep". Depth is what lets the network learn complicated patterns on its own — for example, it can learn from raw pixels that edges form shapes, shapes form parts, and parts form a cat — without anyone hand-writing features like "pointy ears" for it.

Deep learning shines where data is huge and messy: photos, speech, video and language. It struggles where data is small and tidy: a spreadsheet with 200 rows and clear columns is usually handled better, faster and cheaper by classic machine learning. Deep learning is a tool, not a replacement for everything you learned before.

In this lesson you will learn what a neural network looks like, how depth helps, and a simple rule for choosing between classic ML and deep learning.

Where beginners get stuck: they imagine the network "understands" a photo the way a person does. It does not. It only learns number patterns — which pixels tend to go together, which shapes tend to appear near each other. When you keep that picture in mind, the rest of the series makes sense: training is just finding number patterns that predict well, and testing is checking whether those patterns hold on photos the network has never seen. *(Roman Urdu: Deep learning ka matlab hai kai layers wali neural network — data aik taraf se dakhil hota hai, har layer usay thora badalti hai, aur aakhir main prediction nikalta hai.)*

Key points

- **A neural network is layers of small units.** Each unit takes numbers in, multiplies them by weights, adds them up, and passes the result on. Alone a unit does almost nothing; together, in layers, they can learn complex patterns.
- **Deep = many layers.** Depth lets the network build features step by step: edges → shapes → parts → whole objects. Nobody has to hand-design those features.
- **It learns from examples, like all ML.** Training still means showing labelled examples and adjusting the model until its predictions improve. Deep learning just uses a bigger, layered model for that job.
- **It loves big, messy data.** Photos, audio, video and text have far too many raw inputs for hand-written features. Deep networks handle that raw input directly.

- **It is hungry: data, compute and time.** Training a deep network needs lots of examples and often a GPU (a chip built for parallel number-crunching). On a small dataset it will simply memorise — overfitting, the same enemy from ML.7.
- **Classic ML still wins on small, tidy data.** If your data fits in a spreadsheet and the patterns are clear, a simple model trains in seconds, explains itself, and often scores higher.
- **One big layer is not the same as many layers.** A single layer can only draw simple boundaries. Many layers let each layer build on the last one, so the final decision can be curved and complex.
- **Labels decide what is learned.** If the training photos are labelled wrong, or if they all come from one place, the network learns those mistakes too. Good labels matter more than a fancy network.

Examples

Example 1 — A tiny network, layer by layer

Imagine a network that decides whether a 2×2 black-and-white picture is a "corner" shape.

- **Input layer:** 4 numbers, one per pixel (1 = black, 0 = white).
- **Hidden layer:** 2 units. One learns "top row is black", the other learns "left column is black". Each unit just weighs its inputs and fires when its pattern is present.
- **Output layer:** 1 unit that says "corner" only when both hidden units fire.

That is a whole neural network in miniature: layers turning raw pixels into a decision. A real image network does the same thing with millions of pixels and many more layers.

Example 2 — When depth matters, and when it does not

- **Photo tagging (deep wins):** a phone sorting 10,000 holiday photos into "beach", "food" and "people". The input is raw pixels; nobody can hand-write features for every possible beach photo. A deep network learns its own features from the photos.
- **House-price table (classic wins):** 300 rows with columns for size, rooms and location. A linear regression or tree trains in seconds and you can explain every prediction. A deep network would be slower, harder to explain, and no more accurate.

Example 3 — Depth building features, step by step

In a network trained on animal photos, researchers can look inside and see roughly what each layer responds to: early layers fire for edges and colours, middle layers for fur texture and eyes, late layers for whole animals. Nobody programmed "look for eyes" — the layers discovered it because it helped the final prediction. That self-built ladder of features is deep learning's superpower.

Example 4 — Counting the weights in a tiny network

Take the corner network from Example 1: 4 inputs, 2 hidden units, 1 output.

- Input → hidden weights: 4 × 2 = 8.
- Hidden → output weights: 2 × 1 = 2.

- Total weights: 8 + 2 = 10 (plus one small bias per unit, 3 in total).

Now scale up: a photo network with millions of inputs and hundreds of units per layer quickly reaches millions of weights. That is why deep learning needs lots of data — every weight is something the network must learn from examples, and with too few examples it just guesses.

Example 5 — Choosing the tool: three mini-cases

- **5,000 voice clips to sort by speaker:** deep learning. The input is raw, messy sound, and there is enough data for a network to learn its own sound features.
- **200 rows of shop sales to forecast next month:** classic ML. The data is small and tidy, a simple model trains in seconds, and you can explain its answer to the shop owner.
- **50,000 product photos to sort into categories:** deep learning. Nobody can hand-write features for 50,000 different photos, but a network can learn them from the photos themselves.

Rule of thumb: messy raw input + lots of data → deep learning; tidy table + little data → classic ML.

Example 6 — Real world: photo search on your phone

Open a phone gallery and search for "beach" or "dog", and it finds matching photos without anyone tagging them by hand. Behind that search is a deep network trained on millions of labelled photos. It learned edges, textures and objects layer by layer, exactly as in this lesson — and the phone runs a small, fast version of that network on the device itself. That is deep learning doing a job no hand-written rule could do: understanding raw pixels at scale.

Quick recap

- Deep learning = machine learning with many-layered neural networks.
- A network turns raw input into a prediction, layer by layer.
- Depth builds features automatically: edges → shapes → parts → objects.
- It wins on big, messy data (images, speech, language); classic ML wins on small, tidy data.
- It needs lots of data and compute, and it can overfit like any model.

Common mistakes

1. **Thinking deep learning replaces all ML** — it does not; on small tabular data, classic models are usually better. Fix: match the tool to the data, not to the fashion.
2. **Thinking "deep" means smart or human-like** — it only means many layers. Fix: judge a network by its test score, never by its size.
3. **Training a deep network on tiny data and trusting it** — it will memorise the examples. Fix: with little data, use a simple model, or reuse a network trained on big data (transfer learning, coming in DL.6).

1. **Expecting deep learning to work without enough data or time** — a deep network trained on 100 photos in ten minutes will disappoint. Fix: either gather much more data, or use transfer learning (DL.4, DL.6), or pick a simpler model.

2. **Judging a network only by its training score** — a network can score 99% on photos it trained on and fail on new ones. Fix: always judge by a separate test set you never trained on.

Practice

1. Name one problem from your daily life where deep learning would help, and one where classic ML would be better. Write one sentence each explaining why.
2. Sketch a 3-layer network for deciding "is this photo a cat or a dog?" — write what you think each layer might learn.
3. A friend has 150 rows of shop sales data and wants to predict next month's sales. Would you suggest deep learning? Write your answer and one reason.

1. Count the weights (no biases) in a network with 6 inputs, 3 hidden units and 2 outputs. Show your working: inputs × hidden + hidden × outputs.
2. A school has 400 labelled photos of classroom objects and wants an app to name them. Deep learning or classic ML? Give your answer and one reason, using the rule of thumb from Example 5.
3. Explain in two sentences why a network trained only on daylight photos might fail on night photos. What is the fix — more night photos, or a bigger network?

Practice quiz

1. What does 'deep' mean in deep learning?

- Very smart
- Many layers in the neural network
- A large computer
- Learning for a long time

Answer: Many layers in the neural network

2. Where does deep learning win over classic ML?

- Small tidy spreadsheets
- Big messy data like photos, speech and language
- Simple counting tasks
- Any task with a calculator

Answer: Big messy data like photos, speech and language

3. What do early layers of an image network learn?

- Whole animals at once
- Edges and colours, which later layers build into shapes and objects
- File names
- Camera settings

Answer: Edges and colours, which later layers build into shapes and objects

---

Watch the video lesson: https://youtu.be/QGoVSlsH97U

Read online: https://scopetreasure.com/notes/dl-what-is-deep-learning · More free tools and notes at https://scopetreasure.com