Fundamentals of Machine Learning

Lecture 1

Resources:

1. Deep Learning with Python, Francois Chollet, Manning Publications, ISBN 9781617294433, 2018)
2. https://www.statisticshowto.com/probability-and-statistics/regression-analysis/find-a-linear-regression-equation/
3. https://www.researchgate.net/publication/371492486_The_Fundamentals_of_Machine_Learning

AI Model for Common Use (1)

courtesy to https://www.wikipedia.org/

ChatGPT

ChatGPT is a generative artificial intelligence chatbot developed by the American company OpenAI and launched in 2022.

Microsoft Copilot

Microsoft Copilot (or **Copilot**) is a generative artificial intelligence chatbot developed by Microsoft. Based on the GPT-4 series of large language models, it was launched in 2023

deepseek

Founded 17 July 2023

DeepSeek is a Chinese artificial intelligence company that develops large language models (LLMs).

Claude

Large language model
Generative pre-trained transformer Foundation model

The first model was released in March 2023 by Anthropic.

cohere

Cohere Inc. is a Canadian multinational technology company focused on artificial intelligence for the enterprise, specializing in large language models. Cohere was founded in 2019

Gemini

Gemini is a generative artificial intelligence chatbot developed by Google. Based on the large language model (LLM), it was launched in 2023

AI Model for Common Use (2)

courtesy to https://www.wikipedia.org/

Current logo since February 24, 2025

Grok is a generative artificial intelligence chatbot developed by xAI. Based on the large language model (LLM) of the same name, it was launched in November 2023 as an initiative by Elon Musk.

Mistral AI SAS

Current logo since 2025

Mistral AI SAS is a French artificial intelligence (AI) startup, headquartered in Paris. Founded in 2023, it specializes in open-weight large language models (LLMs).

Generative Pre-trained Transformer 4 (GPT-4)

Developer(s)	OpenAI
Initial release	March 14, 2023; 2 years ago
Preview release	gpt-4-turbo-2024-04-09 / April 9, 2024; 13 months ago
Predecessor	GPT-3.5
Successor	GPT-4o
Type	Multimodal Large language model Generative pre-trained transformer Foundation model

Llama (**Large Language Model Meta AI**, formerly stylized as **LLaMA**) is a family of large language models (LLMs) released by Meta AI starting in February 2023. The latest version is Llama 4, released in April 2025.

Nova AI refers to **multiple AI-powered products, including a chatbot app and an online video editor**.

AI, Machine Learning, and Deep learning

- In the past few years, **Artificial Intelligence** (**AI**) has been the subject of intense media hype
  - **Machine learning (ML), deep learning (DL)**, and **AI** come up in countless articles, often outside technology-minded publications.
- First, we need to define clearly when we mention AI.
- What are *artificial intelligence*, *machine learning*, and *deep learning*?
- How do they relate to each other?
  - See **Figure 1.1** for the answer.

AI, Machine Learning, and Deep learning

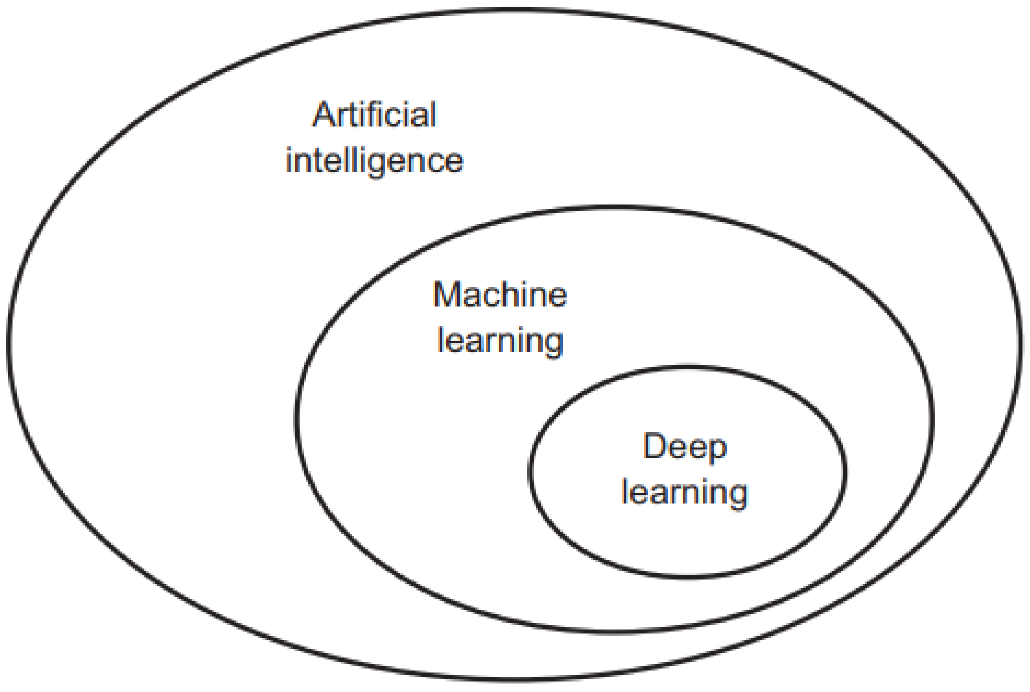


Figure 1.1 Artificial intelligence, machine learning, and deep learning

Artificial Intelligence

- AI was born in the 1950s when a handful of pioneers from the nascent (starting) field of computer science started asking whether computers could be made to "**think**."
- A concise definition of AI *is the effort to* ***automate intellectual tasks*** *usually performed by humans*.
- **AI** is a general field that encompasses ***machine learning*** and ***deep learning*** but includes many more approaches that don't involve **learning**.

Machine Learning

- In the past, until the 1980s, ***symbolic AI*** was popular, and it was handcrafted with a large set of ***explicit rules*** for manipulating the **knowledge domain**.
  - This approach, called the **expert system,** was the dominant AI paradigm from the 1950s to the late 1980s.
    - It turned out to be ***intractable*** to figure out explicit rules for solving more complex, fuzzy problems, such as *image classification*, *speech recognition*, and *language translation*.
- At present, a new approach has emerged to replace **symbolic AI**, called **machine learning (ML)**.

Machine Learning

- **Machine learning (ML)** is a subfield of **artificial intelligence (AI)**
  - ML involves the development of algorithms and statistical models that allow computer systems to learn automatically from and make predictions or decisions based on data.
  - Rather than being explicitly programmed to perform specific tasks, **ML systems** use algorithms to *analyze data*, *identify patterns*, and *make predictions* or *decisions* based on what they learn.

Machine Learning

- There are several types of ML techniques, mainly including ***supervised***, ***unsupervised***, and ***reinforcement*** learning:
  - In ***<u>supervised learning</u>***, the machine learning algorithm is trained on **labeled data**, with the ***input and output variables*** already known, to learn to recognize patterns and make accurate predictions on **new or unseen data**.
  - In ***<u>unsupervised learning</u>***, the algorithm is trained on **unlabeled data** to identify patterns and structures in the data.
  - ***<u>Reinforcement learning</u>*** is a technique that allows the algorithm to learn through ***interactions with an environment***, **receiving feedback in the form of *rewards or punishments***.

Machine Learning

- The applications of ML are vast and proliferating:
  - Some typical applications include *speech recognition*, *image recognition*, *natural language processing*, *recommendation systems*, and *predictive analytics*.
  - ML is used in various industries, including *healthcare*, *finance*, *marketing*, and *manufacturing*.

History of Machine Learning

- **Analytical Engine**: the first-known general-purpose, mechanical computer by Charles Babbage, designed in the 1830s and 1840s.
- At that time, the concept of **general-purpose computation** had not yet been invented.
  – It was merely meant to use mechanical operations to automate certain computations from the field of mathematical analysis—hence, the name Analytical Engine.

History of Machine Learning

- In 1843, **Ada Lovelace** remarked, "*The Analytical Engine has no pretensions whatever to originate anything. It can do whatever we know how to order it to perform*

- AI pioneer **Alan Turing,** in his landmark 1950 paper "Computing Machinery and Intelligence," introduced the **Turing test** and key concepts that would shape **AI**.
  - **Turing** was quoting **Ada Lovelace** while pondering whether general-purpose computers could be capable of learning and originality, and he concluded that they could.

History of Machine Learning

- The history of ML can be traced back to the early days of computing in the **mid-20th century**.
- The first **artificial neural network**, the **Perceptron**, was developed in 1957 by **Frank Rosenblatt**.
- It was a **single-layer neural network** that could recognize simple patterns and was used in *image and speech recognition*.
- In the 1960s and 1970s, machine learning saw significant statistical advances, with decision trees and linear regression development.
- In the 1980s and 1990s, the focus shifted to **expert and knowledge-based systems**, which used **rule-based systems** to make decisions.

History of Machine Learning

- The **2000s** faced a rise of ***big data*** and the need for more sophisticated machine learning algorithms to handle ***vast amounts of data***.
- This led to the development of **deep learning**, a *subset of machine learning* that uses ***neural networks*** *with many layers to learn from data.*
  - Today, **ML** is a rapidly growing field with applications in various industries, including *healthcare*, *finance*, *marketing*, and *manufacturing*
  - *S*ee various **AI platforms made from ML techniques** at the beginning of this lecture slide
- The availability of ***large datasets***, ***faster computing power***, and ***more sophisticated algorithms*** has led to significant advances in the field of ML.

Machine Learning

- **Machine learning** arises from this question: could a computer go beyond "*what we know how to order it to perform*" and *learn how to perform a specified task*?
- *Could a computer surprise us?*
- Rather than programmers crafting data-processing rules by hand, ***could a computer automatically learn these rules by looking at data***?
  - This question opens the door to a new programming paradigm(see **Figure 1.2**).
  - Many current AI platforms, including **ChatGPT**, have answered this question

Machine Learning

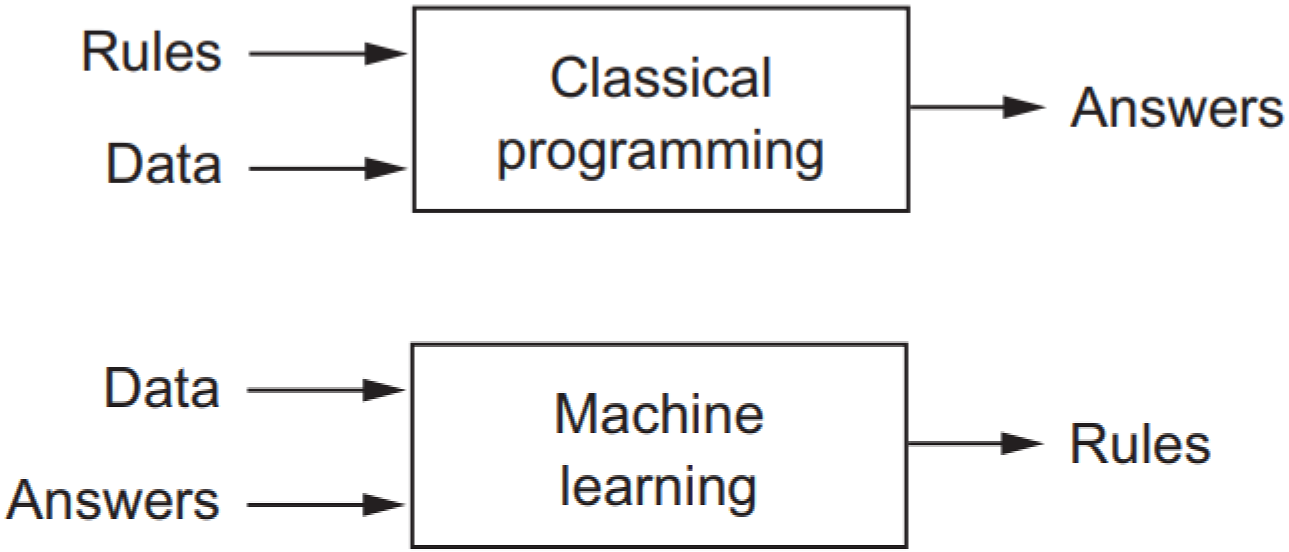


Figure 1.2 Machine learning: a new programming paradigm

Machine Learning

- In **classical programming**, the paradigm of **symbolic AI** involves **input rules** (a program) and **data** to be processed according to these rules, and outcome answers.
- With **machine learning**, humans input data as well as the answers expected from the data and the **outcome of the rules** (see **Figure 1.2**).
  - These rules can then be applied to new data to produce original answers.
  - An ML system is **trained** rather than explicitly programmed.

Machine Learning

- For instance, *if you wished to automate the task of tagging your vacation pictures*, you could present a **machine-learning** system with many examples of pictures already tagged by humans, and the system would ***learn* how to tag** a specific picture from the examples.
- **Machine learning** is tightly related to ***mathematical statistics*** but differs from statistics in several important ways
  - Unlike traditional statistics, **ML** deals with large, **complex datasets** (such as millions of images with thousands of pixels) for which classical statistical analysis, such as **Bayesian analysis,** would be impractical.

Learning Representations from Data

- To understand the difference between **deep learning** and **machine learning**, first, we need some idea of what ML algorithms do.
- Let us see how a **machine learning** algorithm discovers **rules** to execute a data-processing task
  - So, to do **machine learning**, we need **the following** things:
    1. **Input data**
    2. **Past Data**
    3. **Expected output**
    4. **A way to measure the performance**

Learning Representations from Data

- ***Input and past data*:** For instance, if the task is *speech recognition*, **past data** could be ***sound files*** of people speaking, **input data** is the input sound to the system to find its owner, or if the task is *image tagging*, they could be pictures.
- ***Examples of the expected output*:** In a speech-recognition task, these could be *human-generated transcripts* of sound files. In an image task, expected outputs could be tags such as "*dog*," "*cat*," and so on.
- ***A way to measure the algorithm*:** This is necessary to determine the distance between the algorithm's ***current output*** and its ***expected output***.
  - The measurement is a feedback signal to adjust the algorithm's work. This adjustment step is what we call ***learning.***

Learning Representations from Data

- An **ML** model transforms **input data** into meaningful **outputs**, a “ ***learned*** “ process from exposure to ***known examples*** of inputs and outputs.
- Therefore, the central problem in **ML** and **deep learning** is to transform data meaningfully*:*
  - In other words, properly representing the **input data** would provide a **closer expected output**.

Learning Representations from Data

- **What's a data representation?**
  - It's a different way to look at data—to ***represent*** or ***encode*** data.
    - For instance, a **color image** can be encoded in the **RGB** (red-green-blue) format or the **HSV** (hue-saturation-value) **format:** these are two different representations of the same data.
  - Some tasks that may be difficult with one representation can become easy with another.
    - For example, the task "***select all red pixels in the image***" is more straightforward in the **RGB format**, whereas "make the image less saturated" is more straightforward in the **HSV** format
  - **ML models** are all about <u>*finding appropriate representations for their input data*</u>; more explicit

Learning Representations from Data

- Let's make finding appropriate data representations concrete.
- Consider an **x-axis**, a **y-axis**, and some points represented by their coordinates in the (x, y) system, as shown in **Figure 1.3**.

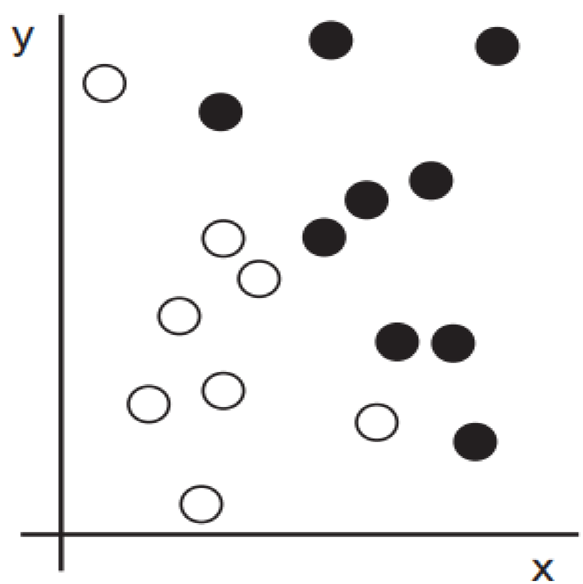


Figure 1.3 Some sample data

Learning Representations from Data

- in **figure 1.3**, you can see a **few white points** and a **few black points.**
- Let's say we want to develop an **algorithm** that can take the **coordinates (x, y)** of a point and output whether that point is likely **black** or **white**. In this case:
  1. The inputs are the coordinates of our points.
  2. The expected outputs are the colors of our points.
  3. A way to measure whether our algorithm is doing a good job could be, for instance, the percentage of points that are being correctly classified.

Learning Representations from Data

- **ML** is, technically, searching for valuable representations of some **input data** within a predefined space of possibilities, using guidance from a **feedback signal**.
  - This simple idea enables the solution of an extensive range of intellectual tasks, from *speech recognition* to *autonomous driving*.

Types of Machine Learning

- There are **three** main machine learning types**: supervised, unsupervised,** and **reinforcement**. Here are some examples of each type:
  - ***Supervised learning*: *Supervised learning*** involves training an ML model (for example, a *neural network*) on **labeled data**, with ***input variables*** (input features) and ***output variables*** (target features) already known.
    - *The algorithm aims to learn to recognize patterns and accurately predict **new data** or **unseen** data (meaning the data is not involved in training)*. Examples include:
      - **Image classification**: identifying whether an image contains a *cat* or a *dog*
      - **Sentiment analysis**: classifying text as *positive*, *negative*, or *neutral*
      - **Fraud detection**: identifying *fraudulent credit card transactions*

Types of Machine Learning

– ***Unsupervised Learning****:* Unsupervised learning involves training a machine learning model on ***unlabeled data*** (*data has no input and output classification*) to identify patterns and structure in the data.
  - *The algorithm aims to independently figure out the underlying patterns without* ***predefined labels or targets***. Examples include:
    – **Clustering**: grouping similar data points together
    – **Anomaly detection**: identifying *unusual patterns* in data
    – **Dimensionality reduction**: reducing the number of features in a dataset

Types of Machine Learning

- ***<u>Reinforcement Learning (RL)</u>****:* Reinforcement learning involves training an **agent** to learn through ***interactions with an environment, receiving feedback*** as rewards or punishments.
  - *The algorithm aims to find the* ***optimal policy that maximizes the cumulative reward***. Applications of RL include:
    - **Game playing**: training an agent to play games like chess or Go
    - **Robotics**: teaching a robot to perform tasks in the real world
    - **Recommender systems**: suggesting products or services to users based on their previous actions

Applications of Machine Learning

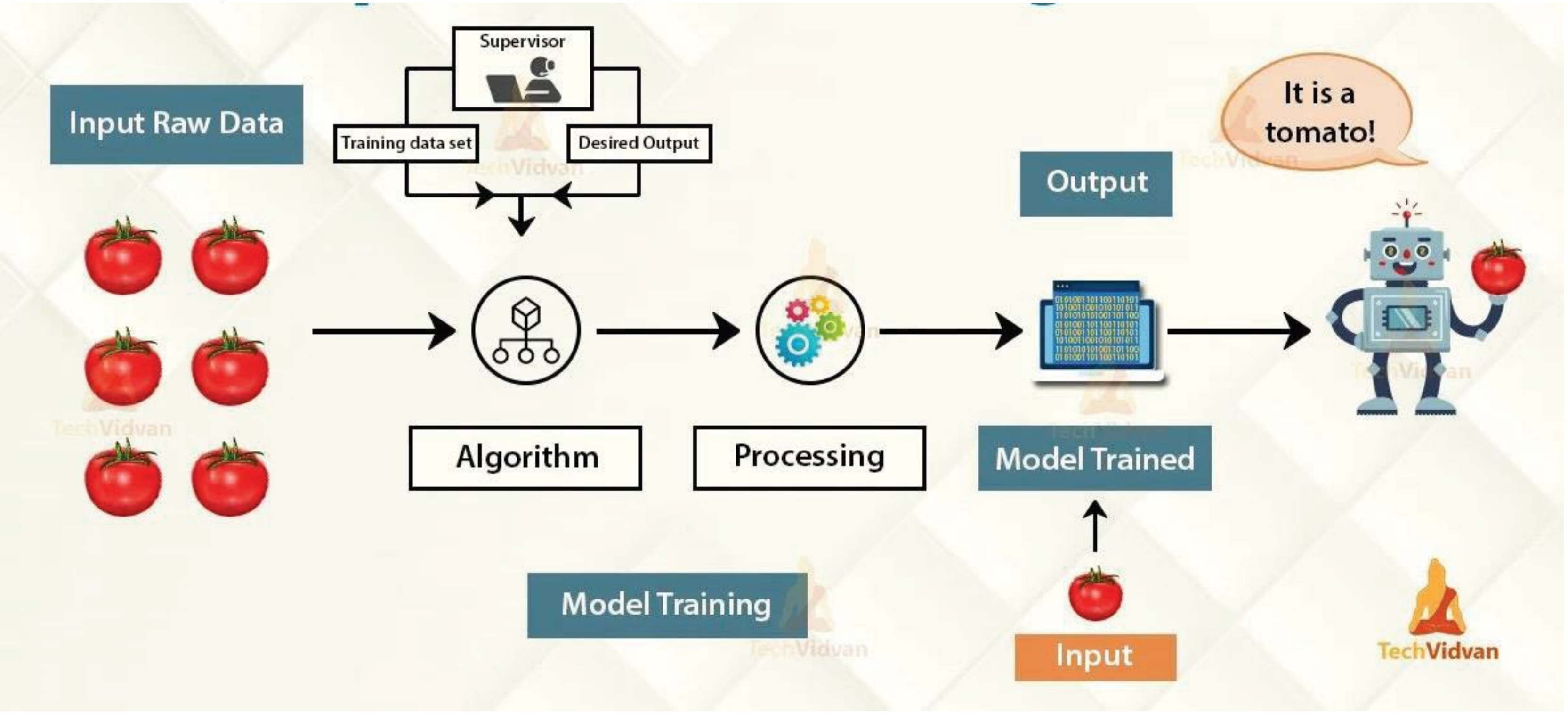
- Machine learning has a wide range of applications; here are some examples:
  - **Healthcare:**
    - Predictive modeling to identify patients at risk of developing certain diseases
    - Image analysis to diagnose medical images like X-rays and MRIs
    - Natural language processing to extract information from medical records
  - **Finance:**
    - Fraud detection in credit card transactions
    - Predictive modeling for loan approvals and credit risk assessment
    - Stock price forecasting
  - **Marketing:**
    - Customer segmentation to target specific demographics
    - Personalized recommendations for products and services
    - Churn prediction to identify customers at risk of leaving

Applications of Machine Learning

- **Manufacturing:**
  - Quality control and defect detection
  - Predictive maintenance to minimize machine downtime
  - Supply chain optimization
- **Transportation:**
  - Traffic flow optimization
  - Autonomous driving technology
  - Predictive maintenance for vehicles
- **Education:**
  - Adaptive learning to personalize education for students
  - Automatic grading and feedback
  - Learning analytics to track student performance and identify areas for improvement

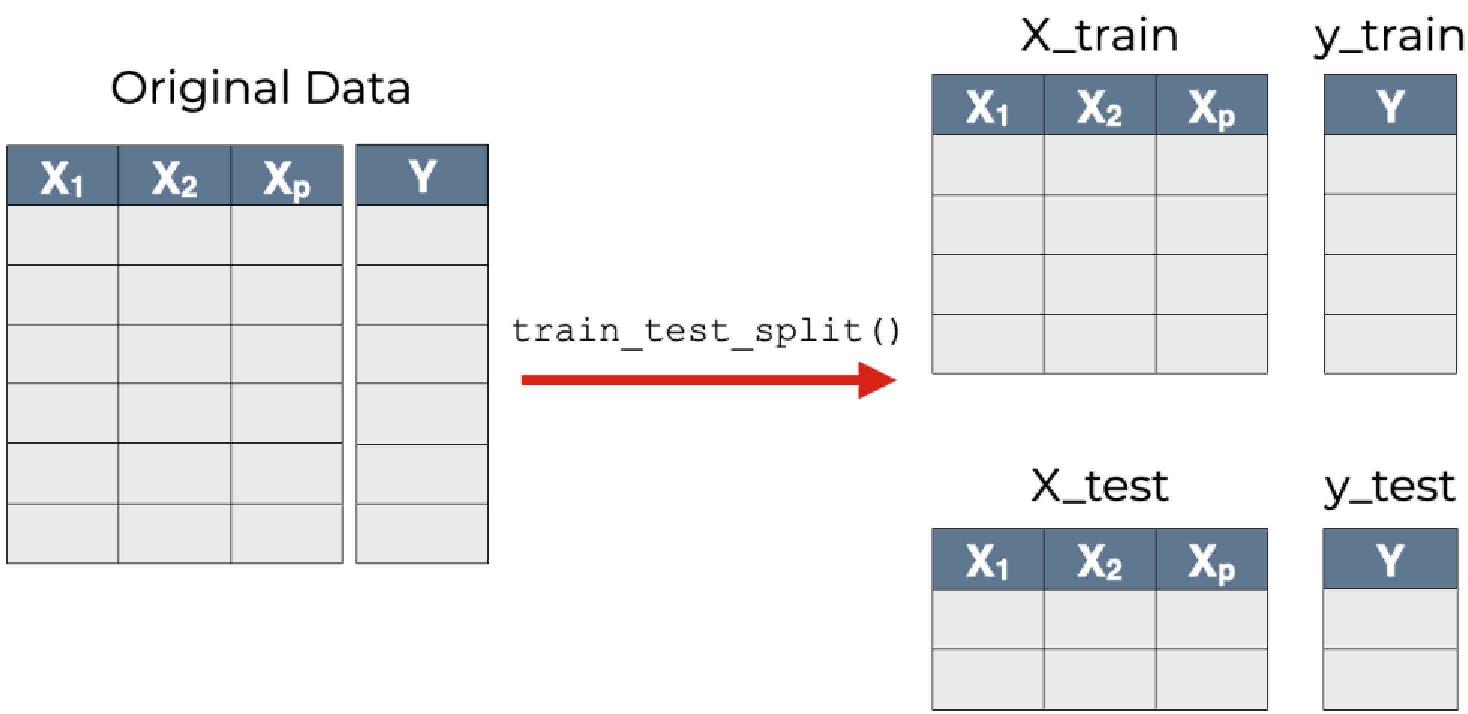
Supervised Learning

- **Supervised learning** is a type of ML where the algorithm is trained on a ***labeled dataset***, meaning that the input data is accompanied by the correct output (also called *target* or *label*) for each data point.
- The goal of supervised learning is to learn a mapping function that can predict the correct output for new, ***unseen input data***. The following diagram illustrates the process of supervised learning:



Supervised Learning

- As shown in the diagram below, the **original data** (labeled data) is split into the ***training set*** (***X_train*** and ***y_train***) and the ***test set*** (***X_test*** and ***y_test***).
- The ***training set*** is used to train the **ML model**, while the ***test set*** is used to ***evaluate its performance***.

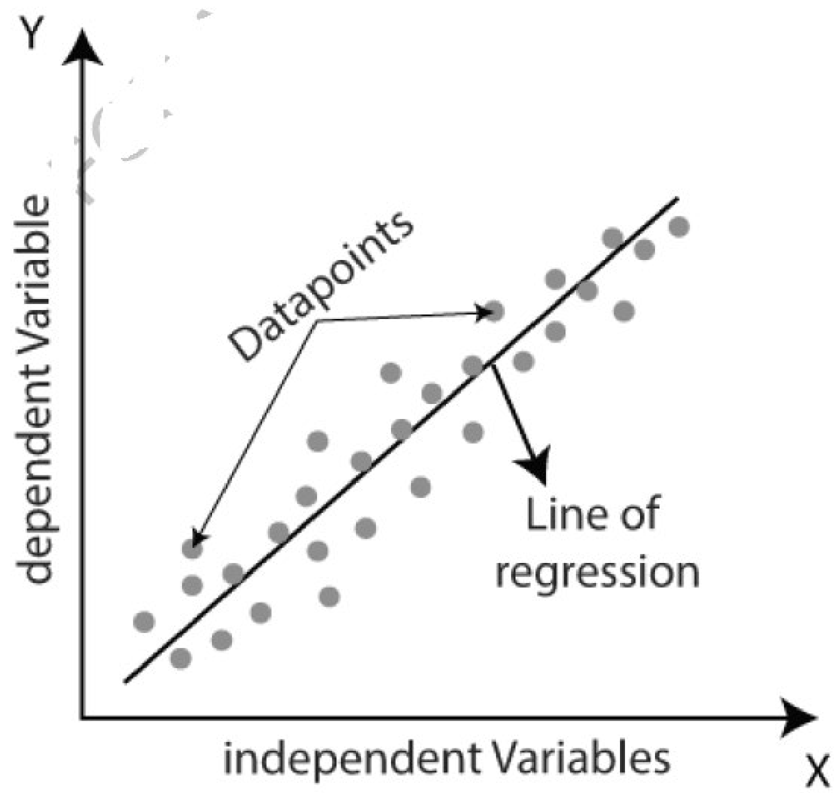


Supervised Learning

- The **input data** (features or predictors) are fed into the ML model, which uses algorithms to learn the mapping function between the input and output data.
- *The goal is to minimize the difference between* ***predicted*** *and* ***actual*** *output*.
  - ***What is the predicted output?***
  - ***What is the actual output?***
- **Once the model is trained**, it is evaluated on the ***test set*** to measure its performance.
  - The **performance metrics** may include ***accuracy***, ***precision***, ***recall***, ***F1 score***, or other metrics, depending on the problem and output data type.
  -

Linear Regression

- **Linear regression** is a basic **supervised learning** model used for predictive learning
  - **Linear regression** predicts a variable's value based on another variable's value.
  - The variable you want to predict is called the ***dependent variable***.
  - The variable you are using to predict the other variable's value is called the ***independent variable***.
  - Linear regression predicts a ***dependent variable (y)*** based on the given ***independent variable (x)***. Hence, the name is ***linear regression***.



Linear Regression

- **Linear regression** is used to model the relationship between a ***dependent variable*** (also called the *response variable*) and one or more ***independent variables*** (also called *predictors* or *features*) that are assumed to be ***linearly related*** (as a straight line).
- **Input** and **output** of a simple **linear regression** are:
  - **Input:**
    - Independent variable ***x***
    - Dependent variable ***y***
  - **Prediction output: $y' = a + bx$**
    - ***a*** = ***Intercept*** of the line
    - ***b*** = ***slope*** or ***linear regression coefficient*** of the line
    - How to calculate ***a*** and ***b***?

Linear Regression Steps

- Here is the algorithm for a **simple linear regression**:
  1. Set **labeled dataset** with marked **input (*x*)** and **output (*y*)**.
  2. Calculate the corresponding xy, x^2, and y^2 of each row of dataset.
  3. Calculate Σx, Σy, $\Sigma(xy)$, Σx^2, and Σy^2
  4. Calculate the **intercept *a*** using the following formula:
     $a = [((\Sigma y)*(\Sigma x^2)) - (\Sigma x * \Sigma(xy))] / [(n*\Sigma x^2) - (\Sigma x)^2]$, where n is the number of data samples
  5. Calculate the **slope or coefficient *b*** using the formula:
     $b = [(n*\Sigma xy) - (\Sigma x * \Sigma y)] / [(n*\Sigma x^2) - (\Sigma x)^2]$, where n is the number of data samples
  6. Insert the values of ***a*** and ***b*** into the **prediction output** $y' = a + bx$
  7. If x is a **new input** (meaning, *not from the dataset*), its output y' is generated.

Linear Regression Steps

x	y	x*y	x^2	y^2
0.7	1.7	1.19	0.49	2.89
1.1	2.4	2.64	1.21	5.76
1.3	2.5	3.25	1.69	6.25
1.9	2.7	5.13	3.61	7.29
2.6	2.9	7.54	6.76	8.41
3.1	3.4	10.54	9.61	11.56
3.9	3.7	14.43	15.21	13.69
$\Sigma x = 14.6$	**$\Sigma y = 19.3$**	**$\Sigma(xy) = 44.72$**	**$\Sigma x^2 = 38.58$**	**$\Sigma y^2 = 55.85$**

$(\Sigma x)^2 = 213.16$

Linear Regression Steps

- Calculate the line **intercept *a*** using the following formula:
  $\boldsymbol{a} = [((\Sigma \boldsymbol{y}) \times (\Sigma \boldsymbol{x^2})) - (\Sigma \boldsymbol{x} \times \Sigma(\boldsymbol{xy}))] \,/\, [(\boldsymbol{n} \times \Sigma \boldsymbol{x^2}) - (\Sigma \boldsymbol{x})^2]$, where ***n*** is the number of data samples

  $\boldsymbol{a} = [(19.3 \times 38.58) - (14.6 \times 44.72)] \,/\, [(7 \times 38.58) - 213.16]$

  $= [744.594 - 652.912) \,/\, [\,270.06 - 213.16]$

  $= 91.682 \,/\, 56.9$

  $= \mathbf{1.611283}$

Linear Regression Steps

- Calculate the line **slope or coefficient *b*** using the formula: $\boldsymbol{b} = [(\boldsymbol{n} \times \Sigma \boldsymbol{xy}) - (\Sigma \boldsymbol{x} \times \Sigma \boldsymbol{y})] / [(\boldsymbol{n} \times \Sigma \boldsymbol{x}^2) - (\Sigma \boldsymbol{x})^2]$, where ***n*** is the number of data samples

  $\boldsymbol{b} = [(7 \times 44.72) - (14.6 \times 19.3)] / [(7 \times 38.58) - (213.16)]$

  $= [313.04 - 281.78] / [270.06 - 213.16]$

  $= 31.26/56.9$

  $= \mathbf{0.549385}$

- <u>**Next**</u>, insert the values of ***a*** and ***b*** into the **prediction output** $\boldsymbol{y' = a + bx}$ (***x*** is the new (unseen) value for the prediction)
- If $\boldsymbol{x}$ = <u>**3.76**</u>, the predicted value, $\boldsymbol{y'} = \mathbf{1.611283 + (0.549385 \times 3.76)}$

  $\boldsymbol{y'}$ = <u>**3.68**</u>, Similarly $\boldsymbol{x}$ = <u>**0.89**</u>, then predicted value, $\boldsymbol{y'}$ **=** <u>**2.1**</u>

Linear Regression

- Here is an example of **a simple linear regression problem**:
  - Suppose we want to **predict a house's price y** based on its **size *x*** (in **square feet, *SqFt***).
  - We have the ***house-price.csv*** dataset of houses with their corresponding sizes and prices.
  - The goal is to learn a **linear regression model that can predict the price of a new house**, given its **size**.
  - To solve this problem, we can use the following linear regression equation: ***y' = bx + a***.
  - where ***y*** is the dependent variable (***price***), ***x*** is the independent variable (size in ***SqFt***), ***b*** is the slope (the change in **y** per unit change in **x**), and ***a*** is the ***y*-intercept** (the value of **y** when **x** is **0**).
  - Predict the house price when **x = 2000** Square feet, **predicted price y': 130361.50657**

Linear Regression

```
import pandas as pd
from sklearn.linear_model import LinearRegression
# load the dataset
#df = pd.read_csv(r"C:\Users\Ron\Desktop\my_products.csv")
data = pd.read_csv(r"D:\Pytorch-Code-2023\house-prices.csv")
print(data)
# split the dataset into training and testing sets
X_train = data['SqFt'].values.reshape(-1, 1)
y_train = data['Price'].values.reshape(-1, 1)
# create a linear regression model
model = LinearRegression()
# train the model
model.fit(X_train, y_train)
# predict the price of a new house with size 2000 sq.ft.
X_test = [[2000]]
y_pred = model.predict(X_test)
print('Predicted price:', y_pred[0][0])
```

Linear Regression

```
     Home   Price  SqFt  Bedrooms  Bathrooms  Offers Brick Neighborhood
0       1  114300  1790         2          2       2    No         East
1       2  114200  2030         4          2       3    No         East
2       3  114800  1740         3          2       1    No         East
3       4   94700  1980         3          2       3    No         East
4       5  119800  2130         3          3       3    No         East
..    ...     ...   ...       ...        ...     ...   ...          ...
123   124  119700  1900         3          3       3   Yes         East
124   125  147900  2160         4          3       3   Yes         East
125   126  113500  2070         2          2       2    No        North
126   127  149900  2020         3          3       1    No         West
127   128  124600  2250         3          3       4    No        North

[128 rows x 8 columns]
Predicted price: 130361.50657664731
```

Multiple Regression

- **Multiple Regression** is a **supervised learning method** used to analyze the relationship between **a dependent variable** and **more than one independent variable**.
  - It is commonly used for **classification tasks**, such as *predicting whether a customer will buy a product* or *a patient will be diagnosed with a disease*.
  - See <u>the attached file for understanding the mathematics of **multiple regression with two independent variables and one dependent variable**</u>

Multiple Regression

- **<u>Multiple Regression Homework</u>**
- Predict the value of **BMI** (Body Mass Index) from the ***Height*** and ***Weight*** of a person using logistic regression.
  – Download "**Gender-Height-Weight-BMI**" **CSV dataset** from https://www.kaggle.com/datasets/yersever/500-person-gender-height-weight-bodymassindex/
  – **<u>Use the idea and calculations shown in the given file</u>**

The "Deep" in Deep Learning

- **Deep learning** is a subfield of **ML**, a new take on learning representations that emphasizes **learning successive layers** of increasingly meaningful representations.
- The "**deep**" in deep learning isn't a reference to any deeper understanding achieved by the approach; instead, it is the idea of involving ***successive layers of representations***.
  - The **number of layers** contributing to a data model is called the **model's depth**.
    - Modern **deep learning** often involves **tens or even hundreds of successive layers** of representations; they've all learned automatically from exposure to **training data**.

The “Deep” in Deep Learning

- Other approaches to **ML** tend to focus on learning only **one or two layers of representations** of the data, called **shallow learning**.
- In **deep learning**, these **layered data representations** are learned via models called **neural networks**, structured by layers stacked on top of each other.
  – In learning, the term **neural network** is the central concept of **deep learning** was developed by drawing inspiration from our understanding of the **brain**,
    - **Deep-learning models are not models of the brain.**

The “Deep” in Deep Learning

- **Deep learning models** are not models of the brain; it is just a **mathematical framework** for learning representations from data.
- What do the representations learned by a **deep-learning** algorithm look like?
  - Let’s examine how a network of several layers deep transforms an image of a digit to recognize what digit it is (see **figure 1.5**).

The “Deep” in Deep Learning

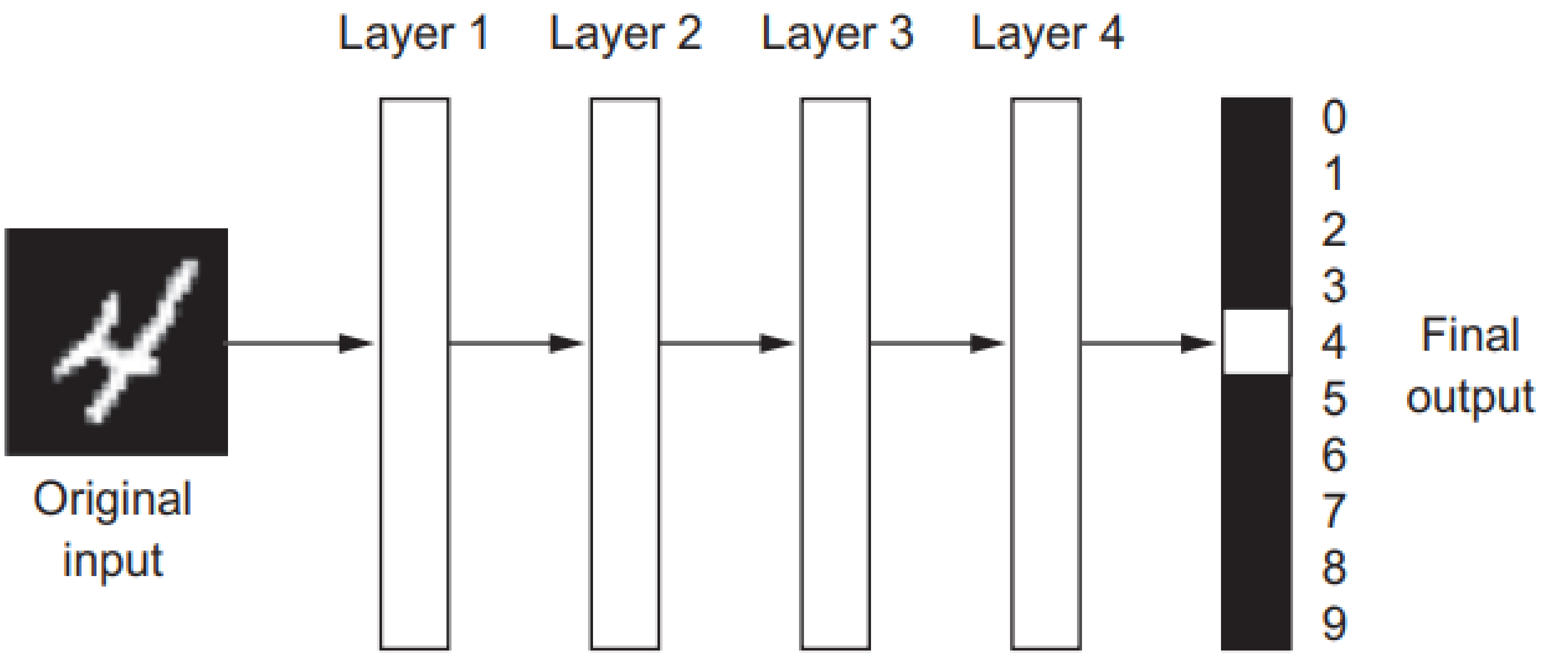


Figure 1.5 A deep neural network for digit classification

The “Deep” in Deep Learning

- The **neural network** transforms the digit image into increasingly different representations from the original image and is informative about the final result shown in **figure 1.6**.
  – You can think of a **deep neural network** as a ***multistage information-distillation*** operation, where information goes through ***successive filters*** and becomes increasingly purified.
  – So **deep learning** is, technically, ***a multistage way to learn data representations***.
    - It’s a simple idea—but, as it turns out, elementary mechanisms, sufficiently scaled, can end up looking like magic.

The “Deep” in Deep Learning

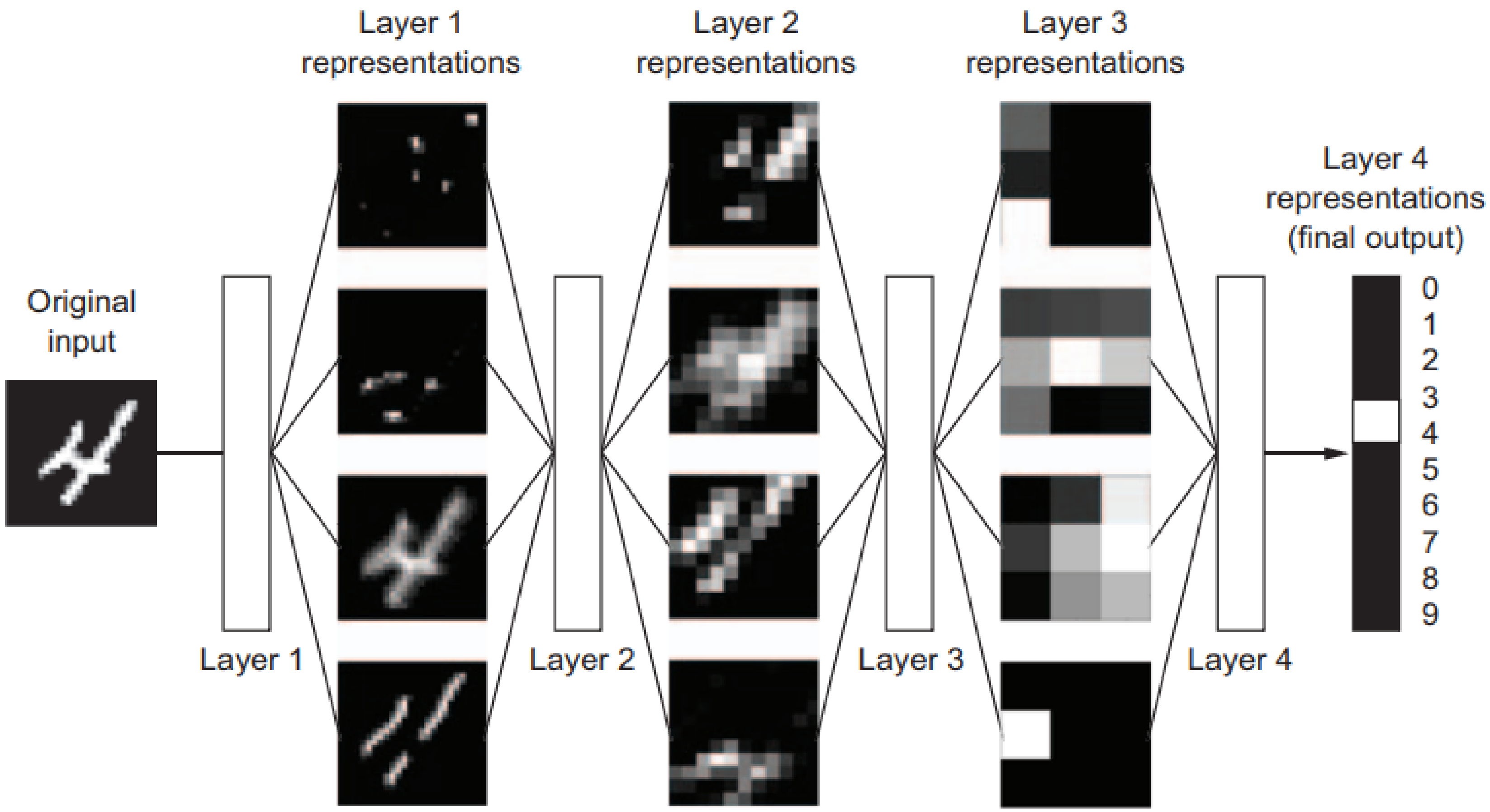


Figure 1.6 Deep representations learned by a digit-classification model

How Does Deep Learning Work?

- We know that **ML** is about mapping **inputs** (such as data/images) to **expected targets** (such as the label "cat"), which is done by observing many **examples of inputs and targets**.
- **Deep neural networks** do this **input-to-target mapping** via a ***deep sequence of simple data transformations*** (**layers**), and these data transformations are learned by exposure to examples.

How Does Deep Learning Work?

- The specification of **what a layer** does to its **input data** is stored in the layer's ***weights*** (which are a bunch of numbers).
- In technical terms, we'd say that the transformation implemented by a layer is **parameterized** by its **weights** (**weights** are also sometimes called the **parameters of a layer**); see **Figure 1.7**.

How Does Deep Learning Work?

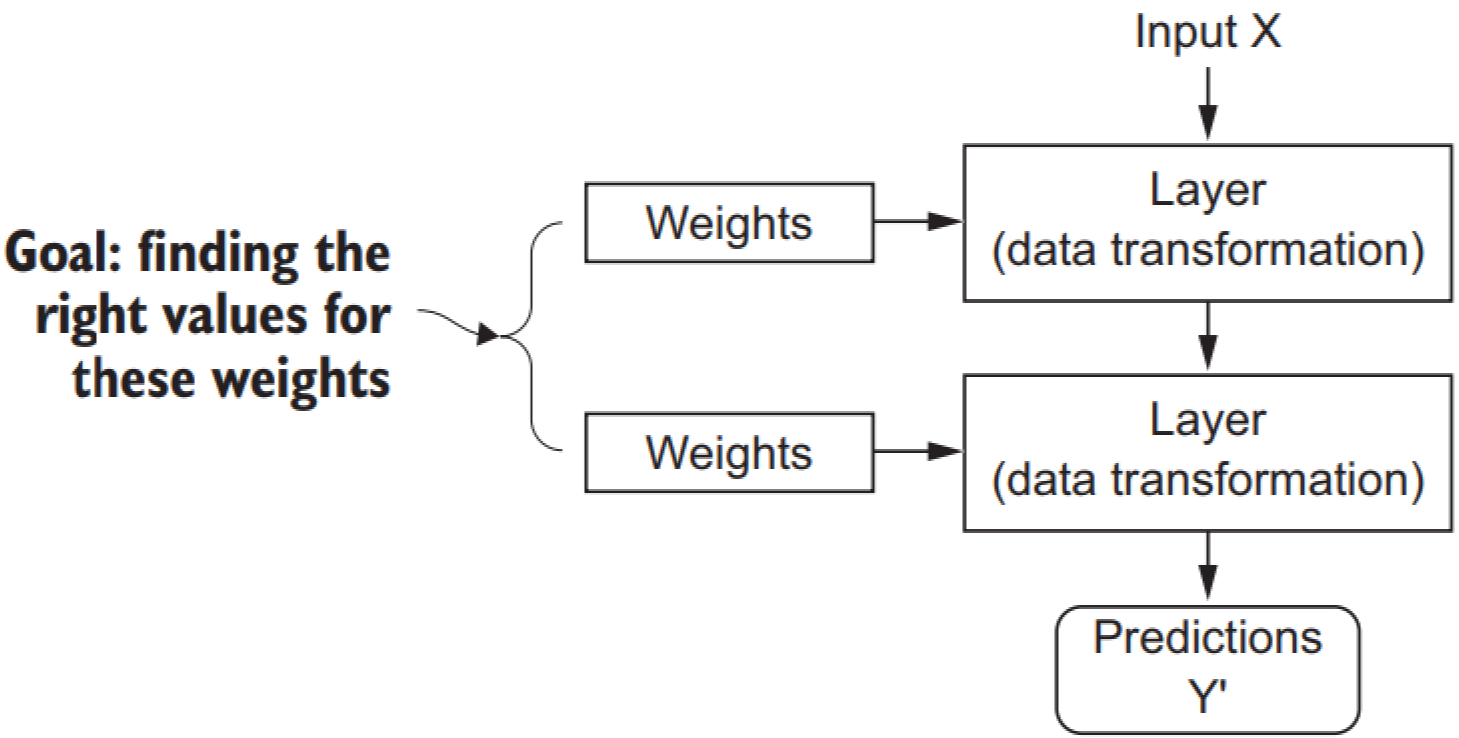


Figure 1.7 A neural network is parameterized by its weights.

How Does Deep Learning Work?

- In this context, ***learning*** *means finding proper values for the* ***weights*** *of all layers in a neural network such that the network will correctly map* ***example inputs*** *to their associated* ***target outputs*** (given as a dataset).
- A **deep neural network** can contain tens of millions of **parameters**.
  – Finding the correct value for all parameters may seem like a daunting (conquering) task, especially given that modifying one parameter's value will affect all the others' behavior!

How Does Deep Learning Work?

- To control the **output** of a **neural network** (called **calculated/predicted output**), you need to be able to measure how far this **output** is from what you expected (called **target output**).
- This is the job of the **loss function** that defines the difference between the **target output** and the network's **calculated/predicted output**.
  - The **loss score** of the network is estimated by the **loss function** from the **target output** and **predicted output** (see **Figure 1.8**).

How Does Deep Learning Work?

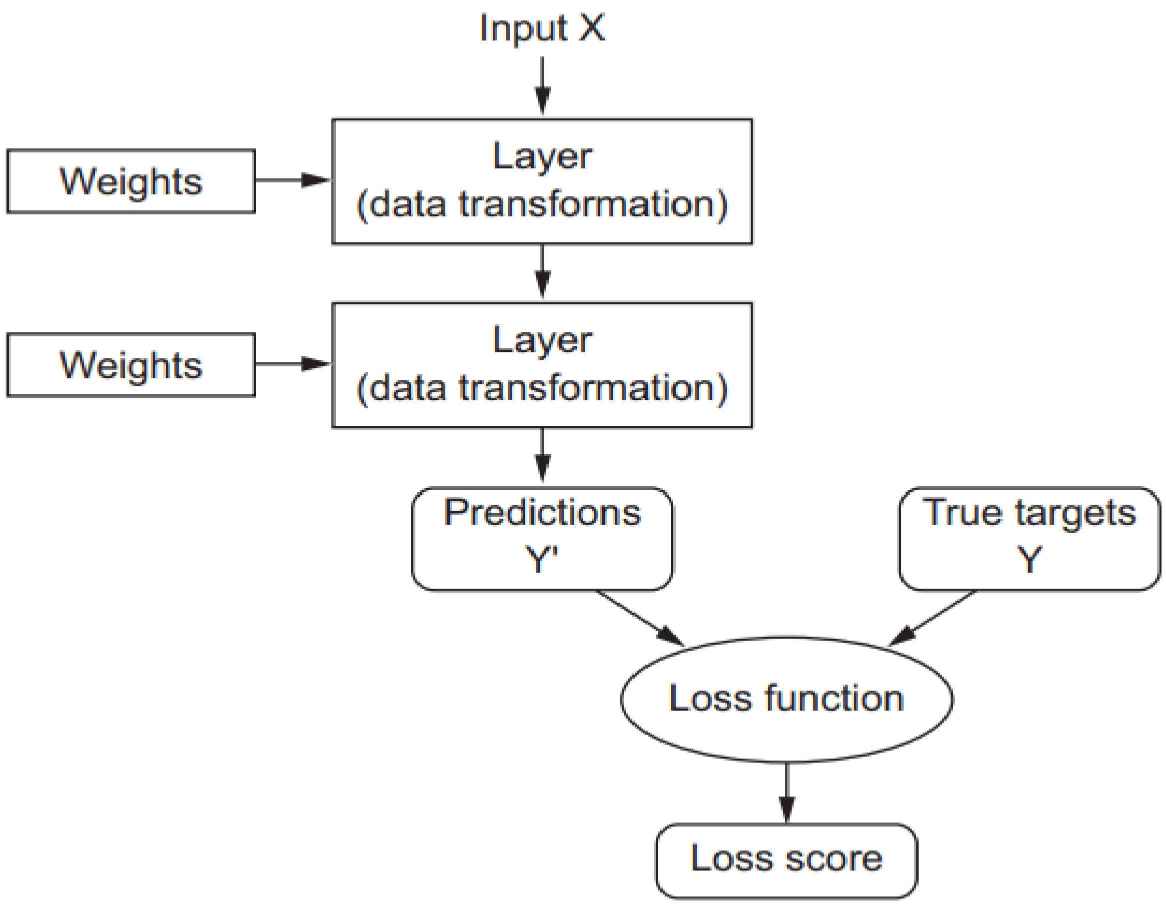


Figure 1.8 A loss function measures the quality of the network's output.

How Does Deep Learning Work?

- The fundamental trick in **deep learning** is to use the **loss score** as **a feedback signal** to adjust the **weights** in a direction that will **lower the loss score** for the current example (see **Figure 1.9**).
- This adjustment is the **optimizer's job** and is implemented by the **Backpropagation algorithm** (one of the central algorithms in deep learning).

How Does Deep Learning Work?

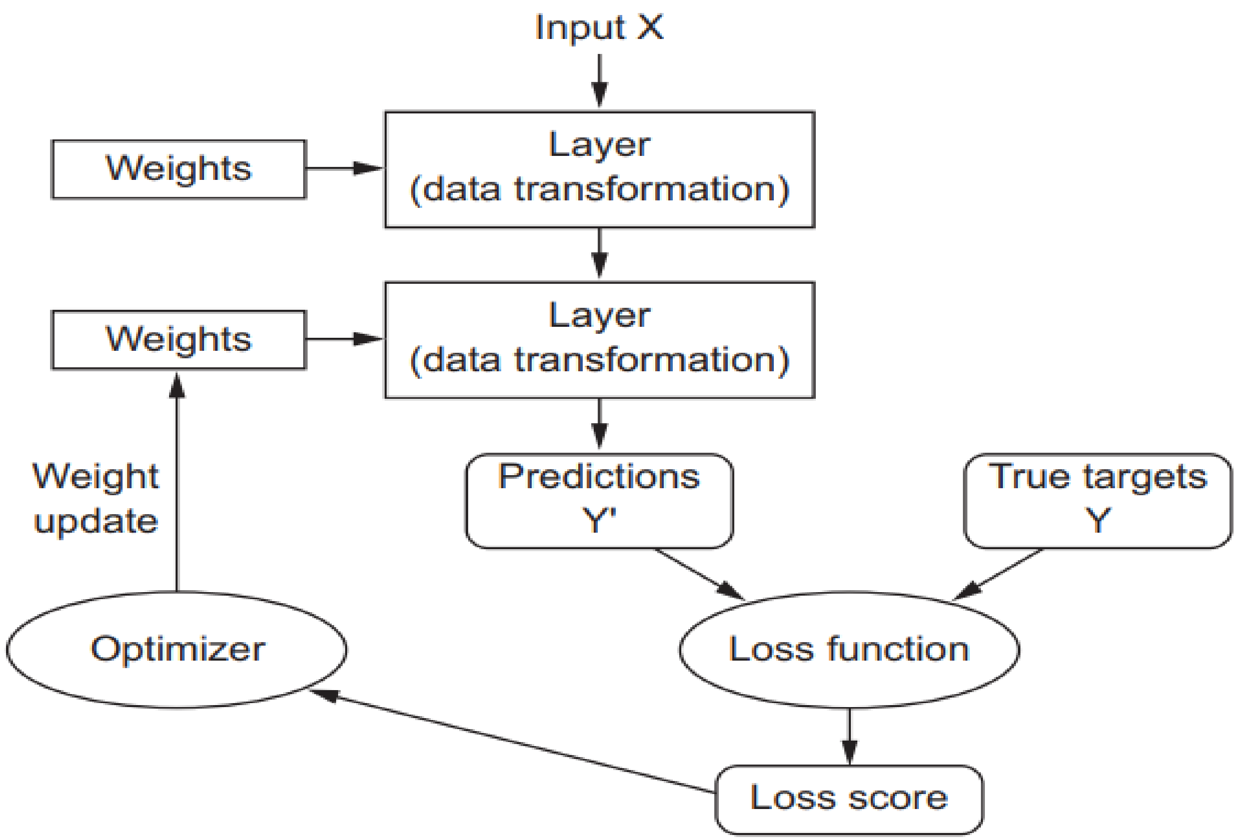


Figure 1.9 The loss score is used as a feedback signal to adjust the weights.

How Does Deep Learning Work?

- Initially, the **weights** are randomly assigned, so the network implements a series of random transformations.
- Naturally, its **output is far from what it should ideally be**, and the **loss score** is very high.
- But with every example of the network processes, the **weights** are **adjusted** in the correct direction, and the **loss score decreases**.
- This **training loop** is repeated several times until it reaches an appropriate **weight value set** that minimizes **the loss function**.
- A neural network with a **minimal loss** is the trained one for the given dataset.

What Deep Learning Has Achieved?

- Although **deep learning** is a fairly old subfield of **machine learning**, it only rose to prominence in the early 2010s.
- **Deep learning** has achieved the following breakthroughs:
  - Near-human-level image classification
  - Near-human-level speech recognition
  - Near-human-level handwriting transcription
  - Improved machine translation
  - Improved text-to-speech conversion
  - Digital assistants such as Google Now and Amazon Alexa
  - Near-human-level autonomous driving
  - Improved ad targeting, as used by Google, Baidu, and Bing
  - Improved search results on the web
  - Ability to answer natural-language questions
  - Superhuman Go playing

Why is deep learning? Why now?

- The two fundamental **Deep learning** ideas for computer vision are **convolutional neural networks (CNN)** and **backpropagation neural networks (BPNN)**.
- In general, **three technical forces** are driving advances in ML:
  1. Hardware
  2. Datasets and benchmarks
  3. Algorithmic advances