

# Classification and Regression

---

by Dr. Rishikesh Yadav

September 17, 2025

Assistant Professor, School of Mathematical and Statistical Sciences, IIT Mandi, India

# Table of Contents – Introduction to Deep Learning

- 1. Recap: Introduction to DL
- 2. Supervised Learning
  - 2.1 Supervised Learning for Predictive Tasks
  - 2.2 Regression
  - 2.3 Classification

## Recap: Introduction to DL

---

# What are AI, ML, and DL?

- **Artificial Intelligence (AI)**

- **Broadest field:** making machines think and act like humans.
- Includes: reasoning, problem-solving, planning, learning.
- Covers all approaches: symbolic AI, expert systems, ML, DL.

- **Machine Learning (ML)**

- **Subset of AI:** learns patterns from data without explicit rules.
- **Scope:** about 70–80% of modern AI progress comes from ML.
- Main paradigms:
  - **Supervised Learning:** Learn from labeled data ( $X, Y$ ) to predict outcomes.  
*Examples:* regression (predict house prices), classification (cat vs dog).
  - **Unsupervised Learning:** Discover hidden structure in unlabeled data ( $X$ ) only.  
*Examples:* clustering (customer segments), dimensionality reduction (PCA).
  - **Reinforcement Learning:** Learn by interacting with an environment to maximize cumulative reward. *Examples:* game-playing (chess, Go), robotics, self-driving cars.

- **Deep Learning (DL)**

- **Subset of ML:** neural networks with many layers (depth).
- **Scope:**  $\approx$ 80–90% of state-of-the-art ML today is DL.
- Excels with **high-dimensional data:** images, text, speech.
- Dominates practical applications today (vision, NLP, speech).

# Historical Development of Deep Learning

- **Early Origins (1940s–1960s):** McCulloch–Pitts (1943): first neuron model, Rosenblatt's Perceptron (1958): first trainable neural network.
- **First AI Winter (1970s):** Minsky & Papert (1969) show perceptron limits (XOR), Loss of funding & shift to symbolic AI.
- **Backpropagation Revolution (1980s):** Werbos (1974), Rumelhart–Hinton–Williams (1986) → training multi-layer NNs.
- **Second AI Winter (1987–1995):** Hardware & data insufficient, neural nets sidelined; symbolic/statistical AI rise.
- **Statistical Learning Era (1990s–2000s):** SVMs (Vapnik), Probabilistic Graphical Models (Pearl), strong math foundations (optimization, RKHS).
- **Deep Learning Boom (2006–2015):** DBNs (Hinton) in 2006, ImageNet/AlexNet (2012), Word2Vec, GANs.
- **Modern Era (2017–Present):** Transformers (2017), LLMs (GPT, BERT), AlphaGo, Diffusion Models, Multimodal AI.

# Applications of Deep Learning – At a Glance

- **Computer Vision**
  - Image recognition, object detection
  - Medical imaging, self-driving cars
  - CNNs, YOLO, U-Net
- **Natural Language Processing (NLP)**
  - Translation, chatbots, summarization
  - RNNs, Transformers, LLMs
- **Speech & Audio**
  - Speech recognition, assistants, music generation
  - RNNs, Transformers, GANs
- **Healthcare & Life Sciences**
  - Disease diagnosis, drug discovery, personalized medicine
  - CNNs, GNNs, Bayesian DL
- **Science & Engineering**
  - Climate modeling, physics simulations, astronomy
  - PINNs, Autoencoders
- **Daily Life & Industry**
  - Recommendations (Netflix, Amazon)
  - Finance & fraud detection
  - Social media filters, traffic prediction
  - Graph DL, CNNs, GANs

# Tools for Deep Learning

- **Mathematical Tools**
  - **Linear Algebra** – vectors, matrices, eigenvalues.
  - **Probability & Statistics** – distributions, Bayesian methods.
  - **Calculus** – differentiation, gradients, optimization.
  - **Optimization Theory** – convex and non-convex methods.
- **Programming Foundations**
  - **Python** (we will use throughout)
  - **Mathematical Libraries:** NumPy, SciPy, SymPy.
  - **Data Handling:** Pandas, SQL.
  - **Documentation:** Jupyter Notebooks (we will use throughout)
- **Deep Learning Frameworks**
  - **TensorFlow** (Google) (we will use throughout)
  - **PyTorch** (Meta), **JAX** (Google), **Keras**.
- **Computational Tools**
  - **GPUs & TPUs** – **NVIDIA CUDA**, **Google TPUs**.
  - **Cloud Platforms:** Google Colab (we will use throughout), AWS, Azure, Kaggle.
  - **Local Workstations:** CUDA-enabled NVIDIA GPUs.

## Some Good References

- Bishop, C. M. (2007). *Pattern Recognition and Machine Learning*. Springer.
- Géron, A. (2022). *Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow* (3rd ed.). O'Reilly. [[GitHub companion code](#)]
- Goodfellow, I., Bengio, Y., & Courville, A. (2016). *Deep Learning*. MIT Press. [[Free online](#)]
- Chollet, F. (2021). *Deep Learning with Python* (2nd ed.). Manning.
- Nielsen, M. A. (2015). *Neural Networks and Deep Learning*. [[Free online book](#)]
- Online learning platforms:
  - [Coursera - Deep Learning Specialization](#)
  - [fast.ai - Practical Deep Learning for Coders](#)
  - [DeepLearning.AI](#)



# Supervised Learning

---

# Supervised Learning

- **Definition:** Learn a mapping from inputs  $x$  to outputs  $y$  using labeled data.
  - **Data:**  $(x_i, y_i)$  pairs.
  - **Goal:** minimize prediction error on unseen data.
- **Examples:**
  - **Image classification** – cat vs. dog.
  - **Speech recognition** – audio  $\rightarrow$  text.
  - **Medical diagnosis** – MRI  $\rightarrow$  disease label.
- **Why important?**
  - Powers most current **deep learning applications**.
  - Used in computer vision, NLP, speech, healthcare, finance.
  - **70–80% of deep learning success comes from supervised learning.**
- **Limitations:** requires large labeled datasets, costly to annotate.

# Common Terminology in Supervised Learning

- **Target (label, output, response):**
  - The variable we aim to predict.
  - Also called: **dependent variable** (in statistics), **output variable**, **ground truth**.
  - Can be:
    - **Continuous** → regression (e.g., price, temperature)
    - **Categorical** → classification (e.g., spam vs. not, disease yes/no)
- **Features (inputs, attributes, predictors):**
  - The variables used to predict the target.
  - Also called: **independent variables**, **explanatory variables**, **covariates** (in statistics).
  - Represent measurable characteristics or properties of the data.
- **Example - Predicting Diabetes Diagnosis (Classification):**
  - **Target:** Diabetic (Yes = 1, No = 0)
  - **Features:** Age, BMI, glucose level, blood pressure
- **Notation:**  $\mathbf{X} = [\mathbf{x}_1, \dots, \mathbf{x}_p]$ :  $n \times p$  feature matrix of  $p$  variables, and  $\mathbf{Y} = (Y_1, \dots, Y_n)^\top$ : Target vector

## 2. Supervised Learning

### 2.1 Supervised Learning for Predictive Tasks

#### 2.2 Regression

#### 2.3 Classification

# Supervised Learning in Predictive Modeling

**Predictive Modeling:** Use available data to build models that can predict outcomes for new, unseen cases.

- **Supervised Learning = backbone of predictive modeling.**
  - Input: data with features ( $\mathbf{X}$ ) and labels ( $\mathbf{Y}$ ).
  - Task: learn mapping  $f : \mathbf{X} \rightarrow \mathbf{Y}$ .
  - Output: predictions for unseen inputs.
- **Prediction may mean:**
  - Future value (e.g., next week's rainfall).
  - Class assignment (e.g., cat vs. dog image).
  - Probability score (e.g., disease risk).
- **Note:** Other paradigms can also involve prediction:
  - Unsupervised learning: predict cluster membership, latent features.
  - Reinforcement learning: predict rewards for actions.

**Two main types of Supervised Learning tasks:**

- Regression
- Classification

# Regression vs. Classification

- **Regression:**

- Predicts a **continuous numeric** response variable.
- **Example:** Predicting a person's blood pressure based on age and weight.

- **Classification:**

- Predicts a **categorical or binary** class label.
- **Example:** Predicting whether a tumor is benign or malignant based on imaging features.

- **Shared foundation:**

- Both use observed variables (predictors or features) to model outcomes.
- Both aim to learn a function  $f$  to map inputs to outputs.

**Exercise:** Given a target variable of interest (e.g., house price, disease type, rainfall amount, spam email detection), decide whether it is a **Regression** or **Classification**.

- House price  $\Rightarrow$  Regression
- Disease type  $\Rightarrow$  Classification
- Rainfall amount  $\Rightarrow$  Regression
- Spam email detection  $\Rightarrow$  Classification

# General Formulation

- We aim to learn the relationship between  $p$  predictors  $\mathbf{x} = (x_1, \dots, x_p)^\top$  and a response  $Y$  through a function  $f$ :

$$Y = f(\mathbf{x}) + \varepsilon$$

- For **regression**:
  - $Y \in \mathbb{R}$ : continuous output
  - E.g., house price, blood pressure
- For **classification**:
  - $Y \in \{0, 1\}$  or more general classes
  - E.g., fraud detection, disease status
- $f(\mathbf{x})$  may be:
  - Linear (e.g., linear regression, logistic regression)
  - Non-linear (e.g., decision trees, neural networks)
- Error  $\varepsilon$  accounts for uncertainty.

## Example 1: Regression Task — Housing Prices

- **Goal:** Predict house prices using features of the property.
- **Target  $Y$ :** Sale price (continuous)
- **Predictors  $x$ :**
  - Square footage
  - Number of bedrooms
  - Location
  - House age
- **Key Questions:**
  - How does location affect price?
  - What is the price increase per added bedroom?



## Example 2: Classification Task — Medical Study

- **Goal:** Predict patient outcome (e.g., disease presence).
- **Target  $Y$ :** Treatment success (Yes/No)
- **Predictors  $x$ :**
  - Age
  - Sex
  - Treatment type
  - Comorbidities
- **Key Questions:**
  - Which features are most predictive of outcome?
  - Can we flag high-risk patients before treatment?

## 2. Supervised Learning

### 2.1 Supervised Learning for Predictive Tasks

### 2.2 Regression

### 2.3 Classification

# Linear Regression (1)

- **Linear Regression** is a statistical method for modeling the relationship between a dependent variable and one or more independent variables.
- The relationship is modeled using a linear predictor function whose unknown parameters are estimated from the data.
- In its simplest form, with one dependent variable  $y$  and one independent variable  $x$ , the linear regression model, so-called **Simple Linear Regression (SLR)** is:

$$y = w_0 + w_1x + \varepsilon = \mathbf{w}^\top \mathbf{x} + \varepsilon, \quad \mathbf{x} = (1, x)^\top, \quad \mathbf{w} = (w_0, w_1)^\top,$$

where:

- $y$  is the output variable.
- $x$  is the input variable.
- $w_0$  is the intercept parameter which represents the expected value of  $y$  when  $x = 0$ .
- $w_1$  is the slope, represents change in  $y$  for a one-unit change in  $x$ .
- $\varepsilon$  is the error term, which accounts for the variability in  $y$  that cannot be explained by the linear relationship with  $x$ . Also,  $\mathbb{E}(\varepsilon) = 0$ .

## Linear Regression (2)

- In a classical regression setting, the parameters  $w_0$  and  $w_1$  are estimated from the data using methods such as **Ordinary Least Squares (OLS)**, which minimizes the sum of the squared differences between the observed values and the predicted values.
- The **OLS** estimate of the parameters can be obtained by solving the following normal equations:

$$\begin{pmatrix} \hat{w}_0 \\ \hat{w}_1 \end{pmatrix} = (\mathbf{X}^\top \mathbf{X})^{-1} \mathbf{X}^\top \mathbf{y},$$

where:

- $\mathbf{X} = (\mathbf{1}, \mathbf{x})$ : design matrix of feature variables, including a column of ones,  $\mathbf{1} = (1, \dots, 1)^\top$ , for the intercept and a column of the predictor variable values,  $\mathbf{x} = (x_1, \dots, x_n)^\top$ .
- $\mathbf{y} = (y_1, \dots, y_n)^\top$ : vector of observed values of dependent variable.
- SLR can be extended to include multiple independent (features) variables, resulting in **Multiple Linear Regression (MLR)**:

$$y = w_0 + w_1 x_1 + w_2 x_2 + \dots + w_p x_p + \epsilon.$$

# Interpretation of Regression Model Parameters

- **Estimated (Predicted) Regression Line:**

$$\hat{y} = \hat{w}_0 + \hat{w}_1 x$$

- **Interpretation of  $\hat{w}_1$  (Slope):**

- Represents the estimated change in  $y$  for a one-unit increase in  $X$ .
- **Example:** If  $\hat{w}_1 = 2$ , then for each additional unit of  $X$ ,  $Y$  is expected to increase by 2 units.

- **Interpretation of  $\hat{w}_0$  (Intercept):**

- Represents the estimated value of  $Y$  when  $X = 0$ .
- **Note:** The intercept may not always have a meaningful interpretation, especially if  $X = 0$  is outside the range of observed data.

## 2. Supervised Learning

### 2.1 Supervised Learning for Predictive Tasks

### 2.2 Regression

### 2.3 Classification

# What is Classification?

- **Classification** is a supervised learning task where the goal is to predict a **categorical target variable** (label).
- The output is a class label rather than a continuous number.
- **Binary classification**: Two classes (e.g., spam vs. not spam, disease vs. no disease)
- **Multiclass classification**: More than two classes (e.g., handwritten digit recognition: 0 to 9)
- Given input features  $\mathbf{x}$ , the goal is to learn a function that predicts class membership:

$$f(\mathbf{x}) \in \{0, 1\} \quad (\text{or more generally, } \{1, 2, \dots, K\})$$

- **Example**: Predicting if a patient has diabetes based on age, BMI, and glucose levels.

# Logistic Regression (1)

- **Logistic Regression** is a classical statistical model used for **binary classification**.
- The model estimates the **probability** that a given input belongs to the positive class (e.g.,  $Y = 1$ ).
- The relationship between the predictors  $\mathbf{x}$  and the probability  $P(Y = 1 \mid \mathbf{x})$  is modeled as:

$$P(Y = 1 \mid \mathbf{x}) = \sigma(\mathbf{w}^\top \mathbf{x}) = \frac{1}{1 + e^{-\mathbf{w}^\top \mathbf{x}}}$$

- Here,  $\sigma(\cdot)$  is the logistic (sigmoid) function.
- **Extension to multiple classes:** when there are  $K > 2$  categories, the probability of input  $\mathbf{x}$  belonging to class  $k$  is modeled using the **softmax function**:

$$P(Y = k \mid \mathbf{x}) = \frac{\exp(\mathbf{w}_k^\top \mathbf{x})}{\sum_{j=1}^K \exp(\mathbf{w}_j^\top \mathbf{x})}, \quad k = 1, 2, \dots, K$$

- Outputs are interpreted as class probabilities.
  - In the binary case: threshold (e.g., 0.5) is used to make class predictions.
  - In the multiclass case: the predicted class is the one with the highest probability.



# No Closed-Form Solution for Logistic Regression

- The estimated parameters  $\mathbf{w}$  of logistic regression are obtained by minimizing **log-loss** (or cross-entropy loss), given by:

$$\mathcal{L}(\mathbf{w}) = - \sum_{i=1}^n [y_i \log \hat{p}_i + (1 - y_i) \log(1 - \hat{p}_i)]$$

where  $\hat{p}_i = \sigma(\mathbf{w}^\top \mathbf{x}_i)$  and  $n$  is the number of data points.

- Unlike linear regression, logistic regression does **not** have a closed-form solution for estimating parameters  $\mathbf{w}$ .
- Therefore, classically we opt for **optimization approach** such as Newton-Raphson, Fisher scoring, and Iteratively Reweighted Least Squares (IRLS) to find the parameter estimates  $\mathbf{W}$ .
- Since this is a convex optimization problem, we use **iterative algorithms** such as **gradient descent** to find the optimal  $\mathbf{w}$ .

# Newton–Raphson / IRLS vs. Gradient Descent

- **What is Newton–Raphson?** An iterative optimization method using both the **gradient** and the **Hessian**:

$$\theta^{(t+1)} = \theta^{(t)} - \left[ \nabla^2 \ell(\theta^{(t)}) \right]^{-1} \nabla \ell(\theta^{(t)})$$

- For logistic regression, a variant of it known as **Iteratively Reweighted Least Squares (IRLS)** is used.
- **Newton–Raphson / IRLS (Classical choice)**
  - **Good:**
    - Quadratic convergence near optimum  $\rightarrow$  fast for small/medium datasets. Standard in classical statistics software (R, SAS, Stata).
    - Well-suited for convex problems like logistic regression.
  - **Bad:**
    - Requires computing and inverting the Hessian ( $O(p^3)$ )  $\rightarrow$  costly for high-dimensional data.
    - Memory heavy for very large  $n$  or  $p$ .
- **Why Gradient Descent took over (Modern ML era)**
  - Scales easily to massive datasets ( $n, p$  very large).
  - Works with **stochastic/mini-batch updates**  $\rightarrow$  efficient for streaming data.
  - Integrates naturally with neural networks and non-convex models.
  - Slower per-iteration convergence than Newton–Raphson, but cheaper steps.

# Regression vs. Classification Summary

Aspect	Regression	Classification
Output type	Continuous	Categorical
Model example	Linear / Polynomial Regression	Logistic Regression
Loss function	$\mathcal{L}_{\text{MSE}} = \frac{1}{n} \sum (y_i - \hat{y}_i)^2$	$\mathcal{L}_{\text{log}} = -\frac{1}{n} \sum [y_i \log \hat{y}_i + (1 - y_i) \log(1 - \hat{y}_i)]$
Closed-form solution	Yes (e.g., OLS)	No
Estimation method	Matrix inversion or GD	Gradient descent
Goal	Predict a number	Predict a class or probability