

# Lecture 01 & 02

Introduction to Machine Learning, Vector and Matrix Representation

May 23, 2025

## 1 Supervised & Unsupervised Learning

**Supervised learning** is a type of machine learning algorithm that takes in **labelled** data, whereby inputs are mapped to the corresponding correct outputs. A model learns the mapping from the inputs to outputs based on the provided data. Examples of supervised learning algorithms are **linear regression** and **classification**.

**Unsupervised learning** is another type of machine learning algorithm that takes in **unlabelled** data. This algorithm uses the provided dataset to discover hidden patterns. Examples of unsupervised learning algorithms are **clustering** and **Principal Component Analysis**.

## 2 Linear Regression

**Linear regression** learns a continuous function from datasets to predict outcomes with continuous values. Optimization in linear regression involves finding the set of weights that results in the algorithm to have the minimum Mean Squared Error (MSE) value.

A linear model refers to a model that follows a linear function, involving input features and their respective weights. The optimum model refers to a linear function that results in a linear regression model to have the minimum MSE value.

### 2.1 Mean Squared Error

**Loss:** The difference between the predicted and the actual value at a given training point.

**MSE:** The average loss across all the training points in a dataset.

$$\mathcal{L} = \frac{1}{N} \sum_{n=1}^N (t_n - f(x_n; w_0, w_1))^2$$

## 2.2 Vectorizing Models

Vectorizing models allow us to simplify the mathematical representation of complex models (e.g. a linear regression model with an 8-order polynomial).

We can encapsulate a complex model like this:

$$t = w_0 + w_1x + w_2x^2 + \cdots + w_Kx^K$$

Into this:

$$t = \mathbf{w}^T \mathbf{x}$$

By utilizing vectors, where:

$$\mathbf{w} = \begin{bmatrix} w_0 \\ w_1 \\ \vdots \\ w_K \end{bmatrix}, \quad \mathbf{x}_n = \begin{bmatrix} 1 \\ x_n \\ x_n^2 \\ \vdots \\ x_n^K \end{bmatrix}$$

Therefore, we can turn the mathematical expression of finding the Mean Squared Error (MSE) from this:

$$\mathcal{L} = \frac{1}{N} \sum_{n=1}^N (t_n - f(x_n; w_0, w_1))^2$$

Into this:

$$\mathcal{L} = \frac{1}{N} (\mathbf{t} - \mathbf{X}\mathbf{w})^\top (\mathbf{t} - \mathbf{X}\mathbf{w})$$

When minimizing loss, we need to do partial derivatives with respect to vector  $\mathbf{w}$  and equal it to zero:

$$\frac{\partial \mathcal{L}}{\partial \mathbf{w}} = \mathbf{0}$$

Allowing us to derive the vectorized equation to find the minimum weights, which requires the feature inputs and their corresponding labelled data:

$$\hat{\mathbf{w}} = (\mathbf{X}^\top \mathbf{X})^{-1} \mathbf{X}^\top \mathbf{t}$$

A vectorized expression of the weights and feature inputs of an 8-order polynomial model:

$$t = w_0 + w_1x + w_2x^2 + \cdots + w_8x^8$$
$$\mathbf{w} = \begin{bmatrix} w_0 \\ w_1 \\ \vdots \\ w_8 \end{bmatrix}, \quad \mathbf{X} = \begin{bmatrix} 1 & x_1 & x_1^2 & \cdots & x_1^8 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & x_N & x_N^2 & \cdots & x_N^8 \end{bmatrix}$$