Supervised Learning

EN5730 Machine Learning for Communications

Samiru Gayan

Department of Electronic and Telecommunication Engineering
University of Moratuwa

2024

Supervised Learning

- In supervised learning, we aim to build a model that takes an input $\mathbf{x}$ and outputs a prediction $\mathbf{y}$.
- For simplicity, we assume that both the input $\mathbf{x}$ and output $\mathbf{Y}$ are vectors of a predetermined and fixed size and that the elements of each vector are always ordered in the same way.
- As an example, the input $\mathbf{x}$ would always contain the age of the car and then the mileage, in that order.
- This is termed structured or tabular data.

Supervised Learning

- To make the prediction, we need a model $f[\cdot]$ that takes input $\mathbf{x}$ and returns $\mathbf{y}$.

$$\mathbf{y} = f[\mathbf{x}]$$

- When we compute the prediction y from the input $\mathbf{x}$, we call this inference.
- The model is just a mathematical equation with a fixed form.
- It represents a family of different relations between the input and the output.
- The model also contains parameters ϕ.
- The choice of parameters determines the particular relation between input and output.

$$\mathbf{y} = f[\mathbf{x}, \phi]$$

Supervised Learning

- Learning or training a model: finding parameters ϕ that make sensible output predictions from the input
- Training dataset: these parameters are learnt using a training dataset $\mathcal{I}$ of input and output pairs $\{\mathbf{x}_i, \mathbf{y}_i\}, i \in \mathcal{I}$
- Objective: finding parameters that map each training input to its associated output as closely as possible
- Loss: quantify the degree of mismatch in the mapping
- When we train the model, we seek parameters $\hat{\phi}$ that minimize the loss function $L[\phi]$

$$\hat{\phi} = \underset{\phi}{\arg\min} \quad L[\phi]$$

- Testing: to know how the model will perform in the real world by computing the loss on a separate set of test data

Linear Regression Example

- We consider a model$y = f[x, \boldsymbol{\phi}]$ that predicts a single output y from a single input x.
- A 1D linear regression model describes the relationship between input x and output y as a straight line:

$$y = f[x, \boldsymbol{\phi}] = \phi_0 + \phi_1 x$$

- This model has two parameters:

$$\boldsymbol{\phi} = [\phi_0, \phi_1]^T$$

  where ϕ_0 is the y-intercept of the line and ϕ_1 is the slope.
- Different choices for the y-intercept and slope result in different relations between input and output.

Linear Regression Example

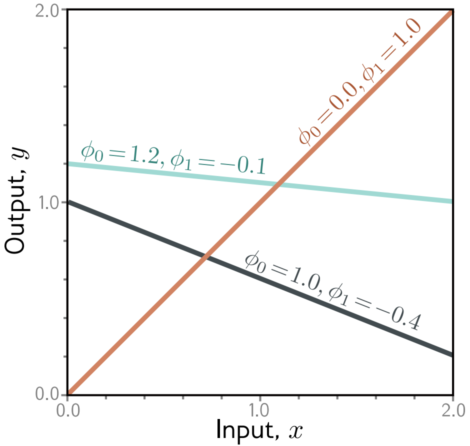

Figure: Linear regression model.

Loss

- The training dataset consists of I input/output pairs x_i, y_i.
- We need a principled approach for deciding which parameters ϕ are better than others.
- To this end, we assign a numerical value to each choice of parameters that quantifies the degree of mismatch between the model and the data.
- We term this value the loss; a lower loss means a better fit.
- The mismatch is captured by the deviation between the model predictions $f[x_i, \phi]$ and the ground truth outputs y_i.
- We quantify the total mismatch, training error, or loss as the sum of the squares of these deviations for all I training pairs:

$$L[\phi] = \sum_{i=1}^{I} \left(f\left[x_i, \phi\right] - y_i\right)^2 = \sum_{i=1}^{I} \left(\phi_0 + \phi_1 x_i - y_i\right)^2$$

- This is the least-squares loss.

Loss

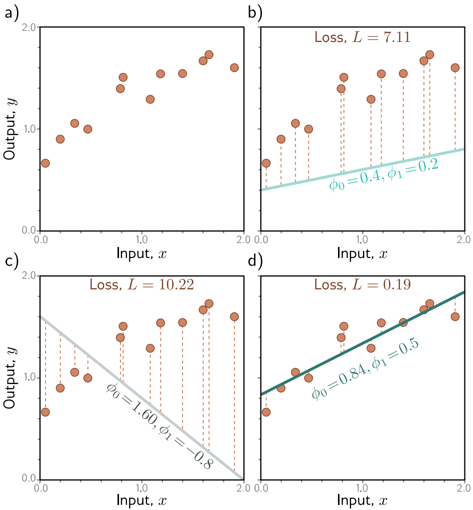

Figure: Linear regression training data, model, and loss.

Loss

- The loss L is a function of the parameters ϕ; it will be larger when the model fit is poor and smaller when it is good.
- The goal is to find the parameters $\hat{\phi}$ that minimize this quantity:

$$\hat{\phi} = \underset{\phi}{\arg\min}\left[L\left[\phi\right]\right] = \underset{\phi}{\arg\min}\left[\sum_{i=1}^{I}\left(\phi_0 + \phi_1 x_i - y_i\right)^2\right]$$

Loss

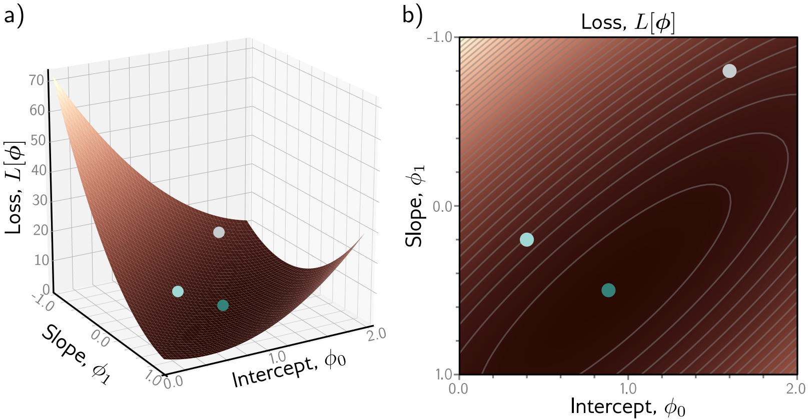

Figure: Loss function for linear regression model. The three circles represent the three lines.

Training

- The process of finding parameters that minimize the loss is termed model fitting, training, or learning.
- The basic method is to choose the initial parameters randomly and then improve them by walking down the loss function until we reach the bottom.

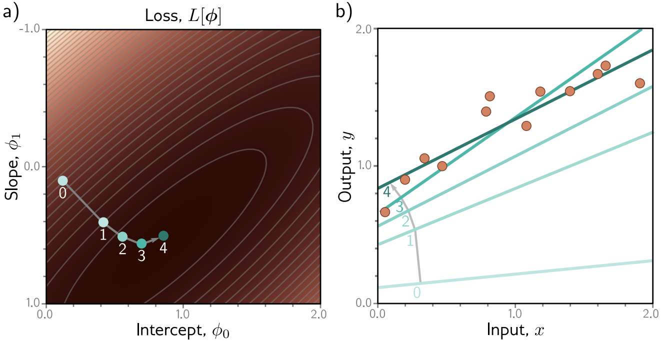

Figure: Linear regression training..

Testing

- Having trained the model, we want to know how it will perform in the real world.
- We do this by computing the loss on a separate set of test data.
- The degree to which the prediction accuracy generalizes to the test data depends in part on how representative and complete the training data is.
- However, it also depends on how expressive the model is.
- A simple model like a line might not be able to capture the true relationship between input and output.
- This is known as underfitting.
- Conversely, a very expressive model may describe statistical peculiarities of the training data that are atypical and lead to unusual predictions.
- This is known as overfitting.

Generative vs. Discriminative Models

Discriminative Models

- Make an output prediction $\mathbf{y}$ from real-world measurements $\mathbf{x}$:

$$\mathbf{y} = f\left[\mathbf{x}, \phi\right]$$

Generative Models

- Real-world measurements $\mathbf{x}$ are computed as a function of the output $\mathbf{y}$:

$$\mathbf{x} = g\left[\mathbf{y}, \phi\right]$$

- Does not directly predict $\mathbf{y}$.
- To perform inference, we invert the generative equation as

$$\mathbf{y} = g^{-1}\left[\mathbf{x}, \phi\right]$$

- This may be difficult

Discriminative models dominate modern machine learning.

References

1. Understanding Deep Learning, first edition by Simon J.D. Prince

- All the images have been taken from [1].

The End