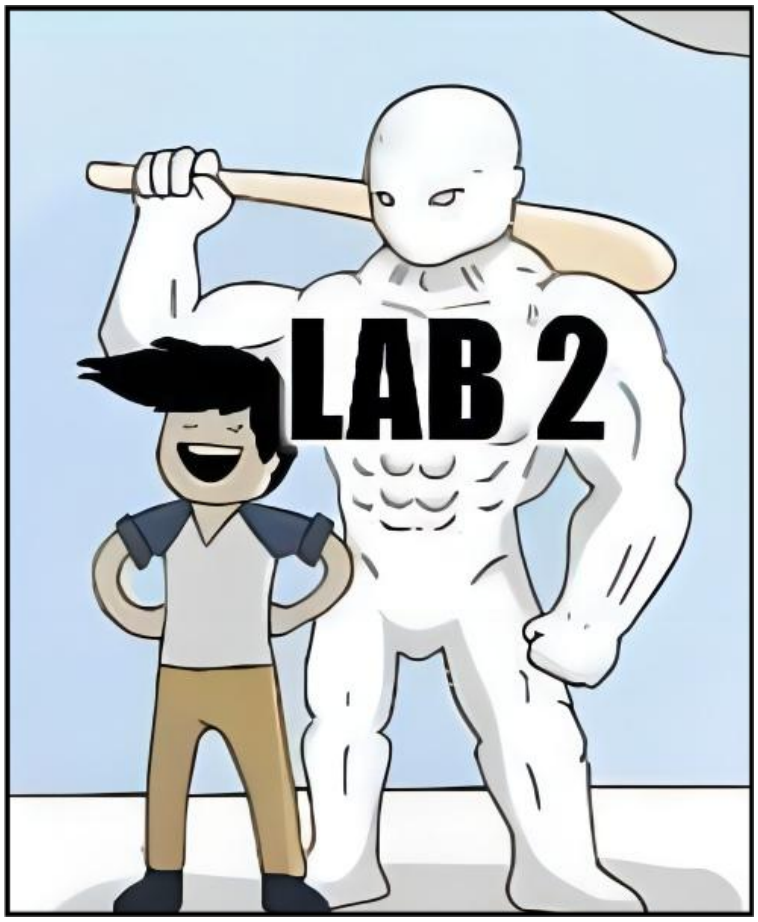
ECE4179 - Lab 2
Model Training with Linear Regression and Logistic Regression



Contents

Academic integrity and the Use of Generative AI

Academic integrity

Every lab submission will be screened for any collusion and/or plagiarism. Breaches of academic integrity will be investigated thoroughly and may result in a zero for the assessment and interviews with the plagiarism officers at Monash University.

Lab Instructions and the Use of Generative AI

- The marks of the lab is based on the Moodle 'Lab 1 Quiz' attempt.
- You need to also submit both Jupyter Notebooks (.ipynb file) to the Moodle submission box 'Lab 1 Submission'. We do not directly mark your notebook, but it is required to show your work out for the questions. More details can be found in the notebook and Moodle quiz instructions.
- You may use NumPy for array handling, and vectorizing your code (reducing the number of for-loops) is encouraged.
- You should use Matplotlib to display images and any intermediate results.
- You may use a generative AI tool or coding assistance. We recommend understanding the concepts and coding as much as possible, as you may be performing hand calculations and write code by yourself during the final exam. If you use generative AI tools, please indicate the prompts you used and explain in detail any changes you made to modify the code to complete the assignment.

Late submission policy and grading guidelines

Late submissions

Late submission policy does not apply for this assessment.

You can apply for a short extension via the lab quiz. The lab submission itself is not marked directly.

Lab Grading

This lab is worth 5/100 (5 out of 100) marks of the entire unit.

Lab attendance is not compulsory. The lab sessions are there for you to get help.

For the .ipynb submission, there are a number of tasks. We do not require you to complete every single section in the notebook. However, for the parts that we ask for a screenshot of your code and outputs, we require those parts to be completed. And the code completed should be functional, ie, if we asked for a screenshot for part 2.2, and the code is dependent on part 2.1, then both parts in the notebook should be completed.

You can see the marks for each sub-task in the Moodle quiz. The final mark of the quiz will be scaled to out of 5% of the unit.

Lab structure and learning outcomes

This lab is about integrating your Python knowledge and applying it to the learning material obtained from the lectures. By the end of this lab, you will be able to train a simple model, understand the concept of training/testing of models, and understand how the hyperparameters would effect the model performance. The following are the three tasks that you should complete.

- Task 1: Simple linear regression. This section focuses on the simple linear regression. You will write a function that calculates the best regression line for two datasets.
- Task 2: Logistic Regression and Gradient Descent. You will be training a logistic regression model with gradient descent. You will also write some helper functions to use for task 3.
- Task 3: Analysing convergence and accuracy. You will learn to analyse convergence of the model and calculate the accuracy of your model in an classification setting.

The learning outcomes for this lab are:

- Training and testing machine learning models.
- Visualizing and understanding the data before applying a machine learning model.
- Applying gradient descent to the logistic regression model.
- Evaluating model performance based on a separate test set.
- Furthering your understanding of Numpy and its core functionality.

Introduction

The concept of machine learning and deep learning has been around for decades. The fundamentals of model training and evaluating model performance correctly is a key requirement to train deep learning models in future labs. In this lab, we will still use Python notebook to lay out the template. After becoming familiar with the lecture and lab material corresponding to this lab, you can begin lab 2 by completing the provided template. The submission deadline is **16th of August (Friday) 4:30 PM AEST**.

It is recommended that you go through the following videos/documents prior to attending your second lab:

1. Begin by reading through this document. This document contains all the relevant information for lab 2.
2. Watch the lab 2 introduction and notebook videos.
3. Watch the lab 2 key concepts video (this will go through core knowledge for this lab) and/or watch the related lecture/workshop content.

In the lab sessions and in your own time, you will be completing the lab 2 notebook and the lab 2 quiz by going through this document and the provided notebook. Note that all the concepts should have been covered within the pre-recorded lectures and the workshops. If you feel unfamiliar to some of them, it would a good idea to review the relevant resources first.

Task 1: Simple linear regression

This section is a recap of linear regression. We will write a simple linear regression function and test it on different types of samples to visually analyze its performance. The sub-tasks are on the topics of:

- 1.1 Simple linear regression formulation
- 1.2 Test and visualize the linear regression

Consider a univariate linear regression problem where there is only one input variable x and one output variable y. The goal of this regression problem is to find the best line fit of the given samples. The simple linear regression equation we will use is written below.

$$y = wx + b \tag{1}$$

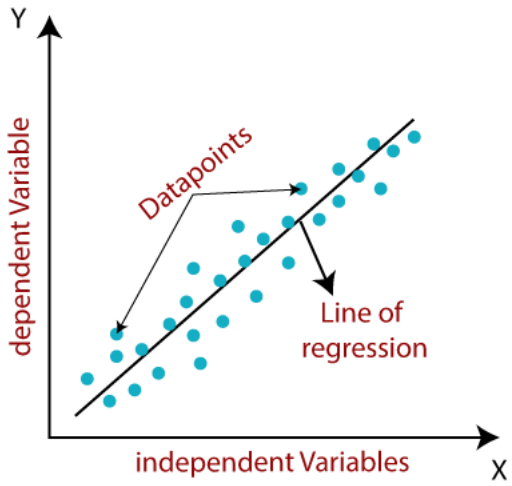


Figure 1: The simple linear regression.

To solve the linear regression problem, we can apply a simple method called Least Squares Regression, which tries to minimise the square distances from each sample to the regression line. The benefit about this method is that, with a given set of samples (X, Y), where $X = \{x_1, x_2, ...x_n\}$ and $Y = \{y_1, y_2, ...y_n\}$, we can apply a closed-form expression for the estimated slope $\hat{w}$ and intercept $\hat{b}$ of the regression line.

$$\hat{w} = \frac{\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})}{\sum_{i=1}^{n}(x_i - \bar{x})^2} \tag{2}$$

$$\hat{b} = \bar{y} - \hat{w}\bar{x} \tag{3}$$

$\bar{x}$ and $\bar{y}$ are the mean values of the data points in X and Y.

Task 2: Training a model via Gradient Descent

In this section, we introduce the Cross Entropy Loss which is useful in training models for binary classification problems. We will also update the model parameters via a simple Gradient Descent method to minimise the loss function. The sub-tasks are on the topics of:

- 2.1 The sigmoid function
- 2.2 Logistic regression
- 2.3 Training with Gradient Descent
- 2.4 Evaluating the trained model

The sigmoid function is formulated as follows:

$$\sigma(z) = \frac{1}{1 + \exp(-z)}. \tag{4}$$

The sigmoid function makes an S shape (figure below) which bounds the values from $(-\infty, +\infty)$ to $(0, 1)$. Hence this is useful for binary classification.

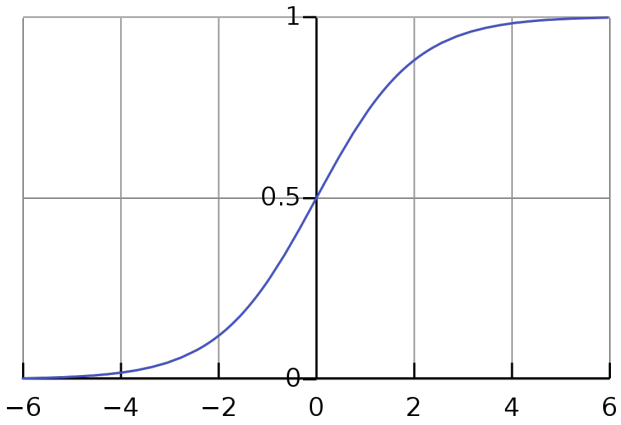


Figure 2: The sigmoid function.

The cross-entropy loss is defined as:

$$\mathcal{L}_{\text{CE}}(\boldsymbol{w}) = -\frac{1}{m}\sum_{i=1}^{m}\left\{y_i \log\Big(\underbrace{\sigma\left(\boldsymbol{w}^\top \boldsymbol{x}_i\right)}_{\hat{y}_i}\Big) + (1-y_i)\log\Big(1 - \underbrace{\sigma\left(\boldsymbol{w}^\top \boldsymbol{x}_i\right)}_{\hat{y}_i}\Big)\right\} \tag{5}$$

And the gradient of the cross entropy loss with respect to the weights $\boldsymbol{w}$ can be written as:

$$\nabla_{\boldsymbol{w}}\mathcal{L}_{\text{CE}} = \frac{1}{m}\sum_{i=1}^{m}\Big(\underbrace{\sigma\left(\boldsymbol{w}^\top \boldsymbol{x}_i\right)}_{\hat{y}_i} - y_i\Big)\boldsymbol{x}_i \tag{6}$$

The gradient can be calculated via the partial derivative of Eq. (5). This gradient function allows us to minimise the loss by updating the parameters of the model ($\boldsymbol{w}$) by deducting this gradient function from $\boldsymbol{w}$. The amount that is deducted is determined by a hyperparameter (usually written as η) called learning rate. The gradient descent updates via the formula:

$$\boldsymbol{w}_{\text{new}} = \boldsymbol{w}_{\text{old}} - \eta \cdot \nabla_{\boldsymbol{w}} \mathcal{L}_{\text{CE}} \tag{7}$$

Training and testing data have been split for you, so you will be using **only** the train data to update the model parameters and using the test data to evaluate the model's performance.

Task 3: Analysing convergence and accuracy

In the following task, you will analyse the accuracy and convergence of the model.

- Accuracy
- Convergence

In any machine learning problem, the base code that generates our model and predictions does not usually yield optimal results. The model requires some fine-tuning of hyperparameters so that the test accuracy is improved.

In this task, we will only change the learning rate. We have provided some learning rate values for you to try out, but in the future, you will have to test the values yourself.

Normally, learning rate is picked between values of $(0, 1]$ and you should test logarithmically, (*i.e.*, 0.01, 0.1, 1). Some of the values we provide are not usually used in real training process, they are more for you to see the effects of different scales of learning rates.

Hopefully this lab has given you insight in training your (potentially) first machine learning model and provided an opportunity to learn about structure of training and testing a model.