Machine Learning

Lab 1

Instructions

This document lists the details of what to do and what to include in the written report. Sample code for both MATLAB and Python is provided and can give hints on how to proceed. Follow the code in your chosen language and the comments for tips. The parts in the sample code that you need to fill in is marked with:

```
# ===================== YOUR CODE HERE ===================== #
```

A written report should be handed in before the end of the course. The report should include the answers to all the asked questions in this document, all the plots that are created, as well as the code used.

Tips: Google Colab or Jupyter notebooks for python or Matlab LiveScript are useful tools for facilitating this.

The report is graded with either PASS or PASS WITH DISTINCTION. Complete all the exercises to receive the higher grade.

Preparation

Some useful resources:

- The Matrix cookbook
- Review Python code
- Matlab - Python code cheat sheet
- Linear Algebra review

Exercises for Pass (G)

1 Load and pre-process data

In this part we will load, visualize, and pre-process the data that will be used to train a linear regression model.

1. Load the dataset in *datasets/carbig.mat*. The input features X is the variable 'Weight' and the output y is the variable 'Horsepower'. Plot the data.

2. Pre-process the data by removing rows of both X and y that contains any NaN (not-a-number) or Inf (infinity) values.

3. Scale the features X down using mean normalization.

$$x_{\text{norm}} = \frac{x - \mu}{\sigma} \tag{1}$$

   where x is the feature vector, μ is the mean of x, and σ is the standard deviation of x.

For the report: Show a plot of the data with x and y-labels and title. Report what the mean and standard deviation is for x after it has been normalized with mean normalization. Also explain why it is important to normalize the features.

2 Linear Regression

In this part, we will implement the linear regression algorithm. Remember from the lectures the linear regression hypothesis, h, is given by:

$$h_\theta(x) = \theta^T x = \theta_0 + \theta_1 x_1 \tag{2}$$

And the cost function, J, is:

$$J(\theta) = \frac{1}{2m} \sum_{i=1}^{m} (h_\theta(x^{(i)}) - y^{(i)})^2 \tag{3}$$

and the partial derivative term becomes:

$$\frac{\partial}{\partial \theta_j} J(\theta) = \frac{1}{m} \sum_{i=1}^{m} (h_\theta(x^{(i)}) - y^{(i)}) x_j^{(i)} \tag{4}$$

1. Implement a function of linear regression that returns the cost J and gradient grad.

2. Compare the analytical and numerical gradient. The difference should be small if you have a correct implementation. *Tips: Make it work with only one training sample first.*

For the report: Report what value of diff and cost you got.

3 Gradient Descent

In this part, we will implement gradient descent and train our linear regression model.

The generic update rule for gradient descent is given by:

$$\theta_j := \theta_j - \alpha \frac{\partial}{\partial \theta_j} J(\theta) \tag{5}$$

where:

- θ_j represents the j^{th} parameter,
- α is the learning rate, and
- $J(\theta)$ is the cost function.

1. Implement gradient descent and train the linear regression model with gradient descent

2. Make a plot of the cost as a function of training iteration. (referred to as training cost/training curve/training loss/loss curve etc)

3. Make a plot of the data and the trained linear model fit.

For the report: Include the plot of training cost and the plot of the linear fit. Discuss the plots. Is the number of iterations in the gradient descent sufficient to find a good solution? Try different values for the learning rate and show how the training plot differs.

4 Linear regression with multiple variables

Now train a linear regression model that predicts a car's horsepower using two input values from */datasets/carbig.mat*, namely 'Weight' and 'MPG'.

1. Load the data with two variables. Visualize the data in a 3D-plot.
2. Train a linear regression model with gradient descent on the new data. Plot the loss curve.
3. Use the trained model to predict how much horsepower a car would have that weights 3000 kg and has a MPG of 30. *Hint: It should be around 95-100. Remember to normalize the values of 3000 and 30 with the mean and standard deviation used when you normalized the data.*

For the report: What prediction of the horsepower did you get? Show how you calculated that value.

5 Normal equation

In this part, we will utilize the normal equation to determine the values for θ without the necessity of using gradient descent. The normal equation is defined as:

$$\theta = (X^T X)^{-1} X^T y \tag{6}$$

where:

- θ is the parameter vector that minimizes the cost function.
- X is the matrix of input features, with each row representing a training example and each column a feature.
- y is the output vector, with each element representing an output for the corresponding training example.

1. Load your data and ensure it's cleaned and structured appropriately for applying the normal equation. *Hint: It is not necessary to normalize the input for the normal equation.*

2. Use the model parameters from the normal equation to predict how much horsepower a car would have that weights 3000 kg and has a MPG of 30.

For the report: Compare the predicted horsepower obtained using the normal equation and obtained using linear regression with gradient descent. Show your calculations for the predicted horsepower using the normal equation.

6 Logistic Regression

Logistic regression classification algorithm will be implemented in this part.

1. Complete the code for the logistic regression classifier and compare the analytical and numerical gradient.

2. Train the logistic regression on hospital data with input variable age and blood pressure to classify if the person is a smoker or not.

3. Calculate the probability that a person with age 32 and blood pressure 124 is a smoker.

For the report: Discuss the results obtained and include plots of the data and logistic regression decision boundary.

Exercises for Pass with distinction (VG)

7 Gradient Descent with Momentum

Implement momentum in the gradient descent algorithm. Remember:

$$v_t := \gamma v_{t-1} + \alpha \frac{\partial}{\partial \theta_j} J(\theta) \tag{7}$$

$$\theta_j := \theta_j - v_t \tag{8}$$

where:

- v_t is the velocity parameter that helps accelerate gradients vectors in the right directions,

- γ (often denoted as β in literature) is the momentum term that controls the blending of the gradient and previous velocity,
- α is the learning rate,
- θ_j represents the j^{th} parameter, and
- $J(\theta)$ is the cost function.

1. Change the gradient descent algorithm to include momentum
2. Train a linear regression model on the carbig data set with both weight and weight+MPG as inputs.

For the report: Implement momentum and show a plot of the training curve with and without momentum in the same plot with different colors. Include a legend that shows which line belongs to which method. Show the code in the notebook. Discuss the effects of using momentum.

8 Non-linear logistic regression with polynomial features

1. Use the variables age and blood pressure to construct polynomial features of degree 2 and use as input to train a logistic regression model.
2. Calculate again what the probability that a person with age 32 and blood pressure 124 is a smoker.

For the report: Discuss the reasons for the difference in the output from a linear and non-linear logistic regression classifier.

9 10-fold cross validation

1. Perform a 10-fold cross validation on the hospital data using a linear logistic regression model.
2. Calculate the mean average and standard deviation of the results from the 10-fold cross validation

For the report: Explain how you performed the 10-fold cross-validation and show the results and the code that was used.