Lab assignment 2 – Logistic regression

1. Requirement

- Install logistic regression to predict whether a factory microchip is eligible to be sold on market.
- Raw data has 3 columns: first and second column are features and third column are labels.
- Raw features mapped to new feature domain consists of 28 dimensions. The map_feature function provided in file **map_feature.py** is responsible for this.
- Implement following functions to execute training and prediction:
  - compute_cost: calculate the cost of model of data set (the formula for calculating cost function is provided in "3. The formulas").
  - compute_gradient: calculate the gradient vector of the cost function (the formula for calculating the gradient vector is provided in "3. The formulas").
  - gradient_descent: calculate the gradient descent.
  - predict: predict whether a set of microchips are eligible to be sold on market (pass an array of 1 element for prediction of 1 microchip).
- Main program:
  - Read the training configuration from file **config.json**.
  - Training data from file **training_data.txt**.
  - Save model to file **model.json**.
  - Make prediction and calculate accuracy of training data set, save result to file **accuracy.json**.

2. Submission rules

- Put all source code and related files to folder named **[student_id]**.
- Compress folder to file **[student_id].zip**.

3. The formulas

- The cost function is calculated by this formula:

$$J(\theta) = \frac{1}{m}\sum_{i=1}^{m}\left[-y^{(i)}\log(h_\theta(x^{(i)})) - (1-y^{(i)})\log(1-h_\theta(x^{(i)}))\right] + \frac{\lambda}{2m}\sum_{j=1}^{n}\theta_j^2.$$

- The formula for calculating gradient vector:

$$\frac{\partial J(\theta)}{\partial \theta_0} = \frac{1}{m}\sum_{i=1}^{m}(h_\theta(x^{(i)}) - y^{(i)})x_j^{(i)} \qquad \text{for } j = 0$$

$$\frac{\partial J(\theta)}{\partial \theta_j} = \left(\frac{1}{m}\sum_{i=1}^{m}(h_\theta(x^{(i)}) - y^{(i)})x_j^{(i)}\right) + \frac{\lambda}{m}\theta_j \quad \text{for } j \geq 1$$