309551101 郭育麟

圖形辨識 Pattern recognition HW1

Part 1

1 . (15%)

Implement the linear regression model and train it by using gradient descent with mean absolute error and mean square error as the objective function, respectively

First , initial weight by random distribution . (w_0 and w_1)

$$\mathrm{y}_{\mathrm{predict}} = w_0 + w_1 x_{train}$$

MSE :

$$\mathrm{L} = \frac{\left(\mathrm{y}_{\mathrm{predict}} - y_{train}\right)^2}{n}$$

$$\frac{\partial \mathrm{L}}{\partial w_0} = \frac{2 * (w_0 + w_1 * x_{train} \ - y_{train})}{n}$$

$$\frac{\partial \mathrm{L}}{\partial w_1} = 2 * \frac{w_0 * x_{train} + w_1 * x_{train}^2 - x_{train} * y_{train}}{n}$$

MAE :

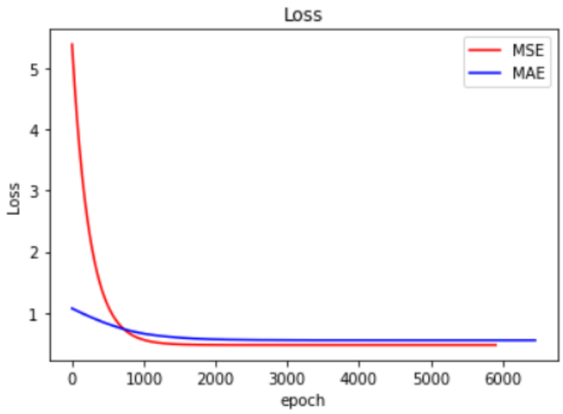
$$\mathrm{L} = \frac{\left|\mathrm{y}_{\mathrm{predict}} - y_{train}\right|}{n}$$

$$\begin{cases} \dfrac{\partial L}{\partial w_0} = 1 \,, \dfrac{\partial L}{\partial w_1} = x_{train} \ \ , if \ y_{predict} > y_{train} \\ \dfrac{\partial L}{\partial w_0} = -1 \,, \dfrac{\partial L}{\partial w_1} = -x_{train} \,, if \ y_{predict} < y_{train} \end{cases}$$

By calculating partial derivative , we could get correct direction , so that we could do gradient descent . After iterations , the model could converge .

2 . (15%)

Plot the learning curve of the training with both losses in the same figure, you should find that loss decreases and converges after a few iterations (x-axis=iteration, y-axis=loss, Matplotlib or other plot tools is available to use)



3 . (15%)

What’re the mean square error and mean absolute error between your predictions and the ground truths on the testing data (prediction=model(x_test), ground truth=y_test)

```
MSE loss in Testing is 0.4909033351910995
MAE loss in Testing is 0.5639134594465589
```

4 . (10%)

What’re the weights β_1 and intercepts β_0 of your linear model trained from both losses?

```
MSE beta_0 is -0.0012568875552996476    beta_1 is 0.4527479923944977
MAE beta_0 is -0.03806884295021232     beta_1 is 0.43502821072553516
```

5 . (10%)

Please explain the difference between gradient descent, mini-batch gradient descent, and stochastic gradient descent?

Gradient : Using all training set data to calculate of the loss function and update the parameters

Mini-batch gradient descent : Using one batch training data to calculate of the loss function and update the parameters

Stochastic gradient descent : Using one training set data to calculate of the loss function and update the parameters

6 . (5%)

All your codes should follow the PEP8 coding style and with clear comments

Part 2

1. (10%)

Suppose that we have three colored boxes R (red), B (blue), and G (green). Box R contains 3 apples, 4 oranges, and 3 guavas, box B contains 2 apples, 0 orange, and 2 guavas, and box G contains 12 apples, 4 oranges, and 4 guavas. If a box is chosen at random with probabilities p(R)=0.2, p(B)=0.4, p(G)=0.4, and a piece of fruit is removed from the box (with equal probability of selecting any of the items in the box), then what is the probability of selecting guava? If we observe that the selected fruit is in fact an apple, what is the probability that it came from the blue box?

(a) $0.2 * \frac{3}{3+4+3} + 0.4 * \frac{2}{2+0+2} + 0.4 * \frac{4}{12+4+4} = 0.2 * 0.3 + 0.4 * 0.5 + 0.4 * 0.2 =$

0.34 , So the probability of selecting guava is 0.34

(b) $0.2 * \frac{3}{3+4+3} + 0.4 * \frac{2}{2+0+2} + 0.4 * \frac{12}{12+4+4} = 0.2 * 0.3 + 0.4 * 0.5 + 0.4 * 0.6 =$

$0.06 + 0.2 + 0.24 = 0.5$

$\frac{0.2}{0.5} = 0.4$, so the probability that it came from the blue box is 0.4 .

2. (10%)

Using the definition $\mathrm{var}[f] = E[(f(x) - E[f(x)])^2]$ show that $\mathrm{var}[f(x)]$ satisfies $\mathrm{var}[f] = E[f(x)^2] - E[f(x)]^2$

$$\mathrm{E[f(x)]} = \frac{1}{n} * \sum_{i=1}^{n} f(x_i) = \frac{1}{\mathrm{n}} * n * \mu = \mu$$

$$\mathrm{var[f]} = E[(f(x) - E[f(x)])^2]$$

$$= \frac{1}{\mathrm{n}} * \sum_{i=1}^{n} [\mathrm{f(x_i)^2} - 2 * \mathrm{f(x_i)} * \mathrm{E[f(x)]} + \mathrm{E[f(x)]^2}]$$

$$= \frac{1}{\mathrm{n}} * \sum_{i=1}^{n} [\mathrm{f(x_i)^2} - 2 * f(x_i) * \mu + \mu^2]$$

$$= \frac{1}{\mathrm{n}} * \sum_{i=1}^{n} f(x_i)^2 - \frac{1}{\mathrm{n}} * 2 * \mu \sum_{i=1}^{n} f(x_i) + \frac{1}{n} * \sum_{i=1}^{n} \mu^2$$

$$= \frac{1}{\mathrm{n}} * \sum_{i=1}^{n} f(x_i)^2 - \frac{1}{n} * 2 * n * \mu^2 + \frac{1}{n} * n * \mu^2$$

$$= \frac{1}{\mathrm{n}} * \sum_{i=1}^{n} f(x_i)^2 - \mu^2$$

$= E[f(x)^2] - \mathrm{E}[\mathrm{f}(\mathrm{x})]^2$

3 . (10%)

Consider two variables x and y with joint distribution $\mathrm{p}(\mathrm{x}, \mathrm{y})$. Prove the following result

$$\mathrm{E}[\mathrm{x}] = E_y[E_x[x|y]]$$

Here $\mathrm{E_x}[x|y]$ denotes the expectation of x under the conditional p(x|y), with a similar notation for the conditional variance.

Hint: Please check the definitions of the expectation operator, the sum rule, and the product rule.

$$\mathrm{E_y}\left[E_x[X|Y]\right]$$

$$= \int_{-\infty}^{\infty} E_x[X|Y = y]\, \mathrm{f_Y}(y)dy$$

$$= \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} x\, f_{X|Y}(x|y)dx f_Y(y)dy$$

$$= \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} x f_{X|Y}(x|y) f_Y(y)dxdy$$

$$= \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} x * f_{X,Y}(x,y)dxdy$$

$$= \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} x * f_{X,Y}(x,y)dydx$$

$$= \int_{-\infty}^{\infty} x \int_{-\infty}^{\infty} f_{X,Y}(x,y)dydx$$

$$= \int_{-\infty}^{\infty} x\, f_X(x)dx$$

$$= \mathrm{E}(\mathrm{X})$$