CS9542B – ARTIFICIAL INTELLIGENCE II – ASSIGNMENT #1

NICKY BAYAT – 251090689

Refreshing Mathematics [20 points]

Question 1:

Gradient is the derivation of a multi-variable function f with respect to x, therefore as a general rule the gradient of this function is:

$$\nabla f(w) = \nabla w^T x = \frac{d\mathrm{w}^{\mathrm{T}} x}{d\mathrm{w}} = \frac{dx^{\mathrm{T}}\mathrm{w}}{d\mathrm{w}} = x^{\mathrm{T}}$$

In linear regression gradient can be calculate based on Mean Squared Error as described below and then this derivative is used to update w.

$$\nabla f(w) = \nabla w^T x = \frac{\partial}{\partial w}\sum_{j=1}^{n}(y_j - w^T x_j)^2 = -2\sum_{j=1}^{n} x_{j,i}(y_j - w^T x_j)$$

Question 2:

We know that:

$$\nabla_A trace(ABA^T C) = CAB + C^T AB^T$$

If we consider C as identity matrix then (1):

$$\nabla_A trace(ABA^T) = AB + AB^T$$

Moreover, we know that:

$$trace(AB) = trace(BA)$$

Therefore, we can say (2):

$$trace(WW^T A) = trace(WAW^T)$$

Based on (1) and (2) we can conclude that:

$$\nabla_w trace(WW^T A) = WA + WA^T$$

reference: https://math.stackexchange.com/questions/1808083/gradient-of-traceabatc-w-r-t-a-matrix-a

---

Question 3:

Hessian matrix of f with respect to w is the first column of H matrix.

$$H(f(w)) = \begin{pmatrix} A + A^T & \nabla_w \nabla_A trace(WW^T A) \\ 2W & \nabla_A \nabla_A trace(WW^T A \end{pmatrix}$$

Question 4:

$$\boldsymbol{\alpha = 1 + e^{-x}}$$

$$\boldsymbol{\beta = \alpha^{-1}}$$

$$\frac{d\, log(\beta)}{d\, x} = \frac{d\, log(\beta)}{d\, \beta} \times \frac{d\, \beta}{d\, x} = \frac{d\, log(\beta)}{d\, \beta} \times \frac{d\, \alpha^{-1}}{d\, \alpha} \times \frac{d\, \alpha}{d\, x} = \frac{1}{\beta} \times \left(\frac{-1}{\alpha^2}\right) \times (-e^{-x}) = \frac{1}{1 + e^x}$$

Linear and Polynomial Regression [50 points]

Linear regression line is calculated using **analytical solution** for gradient of loss.

(a) The figures of this section will be saved in current directory.

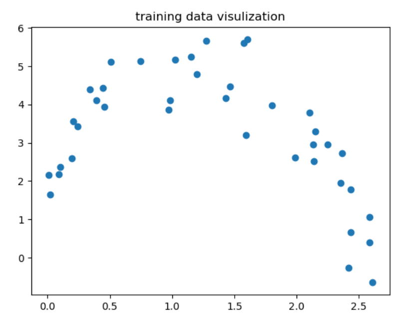

Figure 2 - Training Set

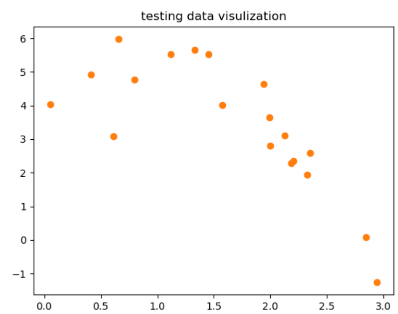

Figure 1 - Testing Set

(b) Average Training Error = **4.3232**

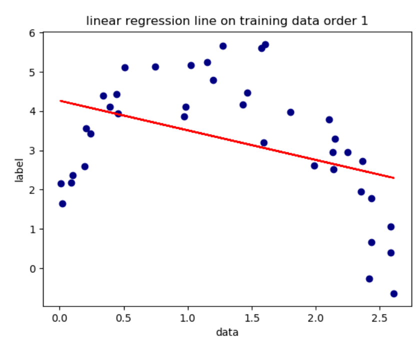

Figure 3 - Linear Regression Train Set

(c) Average Testing Error = **4.65330**

Figure 4 - Linear Regression Test Set

(d) 2nd-order polynomial regression; Since error on both train and test decreased, we can deduce that it is a better fit to our data. This can be observed in below figures as well.

Average Training Error = **0.942**

Average Testing Error = **1.573**

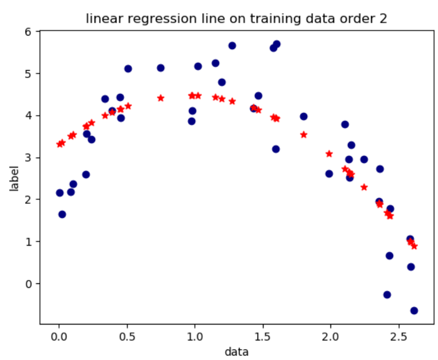

Figure 6 - 2nd-order regression train set

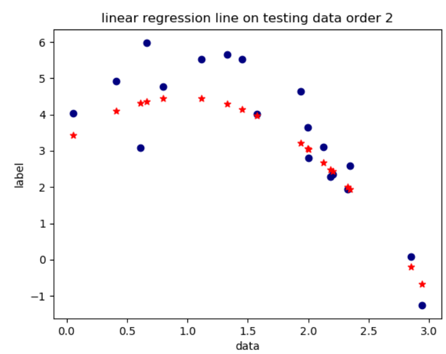

Figure 5 - 2nd-order regression test set

(e) 3rd order polynomial regression: It seems to be better fit than 2nd-order regression. Because both errors decreased.

Average Training Error = **0.940**

Average Testing Error = **1.495**

Figure 8 - 3rd order regression train set

Figure 7- 3rd-order regression test set

(f) 4^{th}-order regression: In this case training error decreases while testing error increases significantly which means we have **over-fitting**; therefore, **we can conduct that 3^{nd} order polynomial regression is the best regression choice for our data**.

Average Training Error = **0.866**

Average Testing Error = **3.162**

Figure 10 - 4th order regression train set

Figure 9 - 4th order regression test set

Regularization and Cross Validation [30 points]

(a) Here is the formula for l_2 regularization and the update function based on that:

$$L(w) = \frac{1}{2}\sum_{i=1}^{m}(\sum_{j=1}^{n} w_0 + w_j.x_{i,j} - y_i)^2 + \frac{\lambda}{2}\sum_{j=1}^{n} {w_j}^2$$

$$w = (X^T X + \lambda I^{\wedge})^{-1} X^T y$$

Using this formula, I have calculated the loss for $\lambda = [0.01, 0.1, 1, 10, 100, 1000, 10000]$. The result for training and testing loss is indicated in the table below:

λ	0.01	0.1	1	10	100	1000	10000
Training Loss	0.876	0.941	1.001	1.775	3.808	5.024	10.730
Testing Loss	2.493	1.688	1.724	4.125	8.949	7.358	11.490

Based on this table and figure 11 we can conclude that **lambda 0.1** works best for our data.

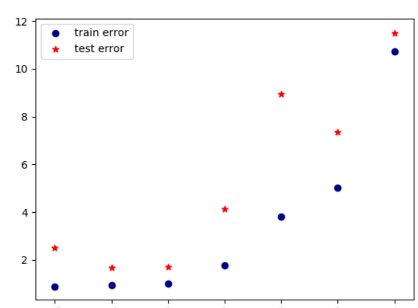

Figure 11 - Error based on different values of lambda

(b) In this section, the value of each weight parameter as a function of $\log_{10}\lambda$ is plotted.

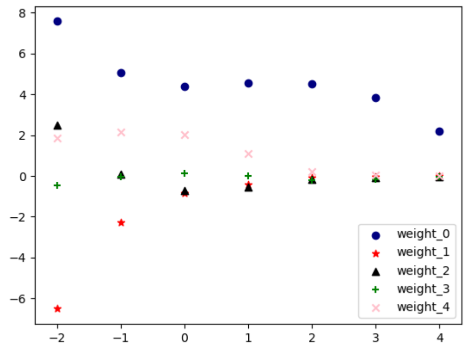

Figure 12 - weight parameter based on lambda

(c) Five-fold cross validation:

The training and validation average error over 5 folds for each lambda is summarized in table below; The best lambda still seems to be **0.1** which is same as (a).

λ	0.01	0.1	1	10	100	1000	10000
Training Loss	0.849	0.924	0.998	1.955	3.909	5.275	11.324
Validation Loss	1.150	1.144	1.168	2.141	4.090	5.462	11.795

l_2 regularized 4th_order polynomial regression curve with lambda=0.1 and test data are represented in the figure below.

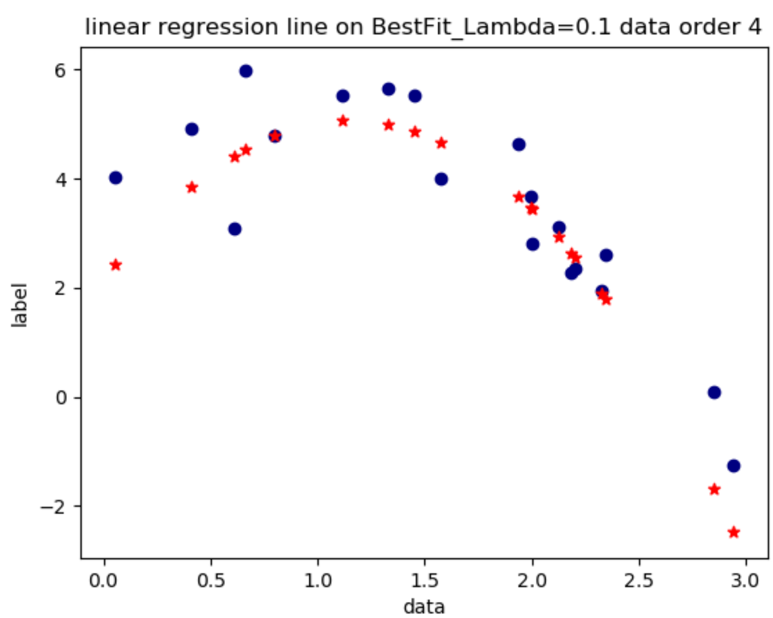

Figure 13 - Best Fit on test data