Aprendizagem 2024/25

Homework III – Group 028

(ist1106987, ist1107057)

I. Pen-and-paper

1) Considering the $\theta(y_1, y_2) = y_1 \times y_2$

D	$\theta(y_1,y_2)$	y_{num}
x_1	1	1.25
x_2	3	7.0
x_3	6	2.7
x_4	9	3.2
x_5	8	5.5

and the OLS closed form solution

$$\beta = (X^T \cdot X)^{-1} \cdot X^T \cdot Y$$

where X is the feature matrix and Y is the y_{num} outcome matrix, the np.lingalg.pinv function is applied which returns the following:

$$\beta_0 = 3.31593, \quad \beta_1 = 0.11372$$

2) Considering the table calculated in the previous exercise and the Ridge Regression

$$\beta = (X^T \cdot X + \lambda \cdot I)^{-1} \cdot X^T \cdot Y$$

where X is the feature matrix, Y is the y_{num} outcome matrix, I is the identity matrix and $\lambda = 1$, the np.lingalg.pinv function is applied which returns the following:

$$\beta_0 = 1.81809, \quad \beta_1 = 0.32376$$

The bias was reduced when switching from OLS to Ridge regularization, which suggests that the regularization is preventing the model from relying too much on a high bias to explain the target variable, aligning it better with the average behavior of the target.

On the other hand, the feature coefficient was increased, which means that the regularization has allowed for more weight to be placed on this feature, giving the model the ability to rely more on it. This implies that the OLS was previously underweighting the feature due to noise in the data.

In conclusion,

3) For each test point, $\theta(y_1, y_2)$ is computed, then the regression equation:

$$\hat{y} = \beta_0 + \beta_1 \times \theta(y_1, y_2)$$

Training samples:

For $x_1 = (1, 1, 1.25)$:

$\theta(y_1, y_2) = 1 \times 1 = 1$

$\hat{y}_{OLS} = 3.31593 + 0.11372 \times 1 = 3.31593 + 0.11372 = 3.42965$

$\hat{y}_{Ridge} = 1.81809 + 0.32376 \times 1 = 1.81809 + 0.32376 = 2.14185$

For $x_2 = (1, 3, 7)$:

$\theta(y_1, y_2) = 1 \times 3 = 3$

$\hat{y}_{OLS} = 3.31593 + 0.11372 \times 3 = 3.31593 + 0.34116 = 3.65709$

$\hat{y}_{Ridge} = 1.81809 + 0.32376 \times 3 = 1.81809 + 0.97128 = 2.78937$

For $x_3 = (3, 2, 2.7)$:

$\theta(y_1, y_2) = 3 \times 2 = 6$

$\hat{y}_{OLS} = 3.31593 + 0.11372 \times 6 = 3.31593 + 0.68232 = 3.99825$

$\hat{y}_{Ridge} = 1.81809 + 0.32376 \times 6 = 1.81809 + 1.94256 = 3.76065$

For $x_4 = (3, 3, 3.2)$:

$\theta(y_1, y_2) = 3 \times 3 = 9$

$\hat{y}_{OLS} = 3.31593 + 0.11372 \times 9 = 3.31593 + 1.02348 = 4.33941$

$\hat{y}_{Ridge} = 1.81809 + 0.32376 \times 9 = 1.81809 + 2.91384 = 4.73193$

For $x_5 = (2, 4, 5.5)$:

$\theta(y_1, y_2) = 2 \times 4 = 8$

$\hat{y}_{OLS} = 3.31593 + 0.11372 \times 8 = 3.31593 + 0.90976 = 4.22569$

$\hat{y}_{Ridge} = 1.81809 + 0.32376 \times 8 = 1.81809 + 2.59008 = 4.40817$

Test samples:

For $x_6 = (2, 2, 0.7)$:

$\theta(y_1, y_2) = 2 \times 2 = 4$

$\hat{y}_{OLS} = 3.31593 + 0.11372 \times 4 = 3.31593 + 0.45488 = 3.77081$

$\hat{y}_{Ridge} = 1.81809 + 0.32376 \times 4 = 1.81809 + 1.29504 = 3.11313$

For $x_7 = (1, 2, 1.1)$:

$\theta(y_1, y_2) = 1 \times 2 = 2$

$\hat{y}_{OLS} = 3.31593 + 0.11372 \times 2 = 3.31593 + 0.22744 = 3.54337$

$\hat{y}_{Ridge} = 1.81809 + 0.32376 \times 2 = 1.81809 + 0.64752 = 2.46561$

For $x_8 = (5, 1, 2.2)$:

$\theta(y_1, y_2) = 5 \times 1 = 5$

$\hat{y}_{OLS} = 3.31593 + 0.11372 \times 5 = 3.31593 + 0.5686 = 3.88453$

$\hat{y}_{Ridge} = 1.81809 + 0.32376 \times 5 = 1.81809 + 1.6188 = 3.43689$

RMSE Formula:

$$RMSE(\hat{y}, y) = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(yi - \hat{y}i)^2}$$

Squared residuals: $(yi - \hat{y})^2$

Training samples:

$$(1.25 - 3.42965)^2 = (-2.17965)^2 = 4.75088$$

$$(7 - 3.65709)^2 = (3.34291)^2 = 11.17506$$

$$(2.7 - 3.99825)^2 = (-1.29825)^2 = 1.68545$$

$$(3.2 - 4.33941)^2 = (-1.13941)^2 = 1.29826$$

Aprendizagem 2024/25

Homework III – Group 028

(ist1106987, ist1107057)

$$(5.5 - 4.22569)^2 = (1.27431)^2 = 1.62388$$

$$\text{RMSE}_{OLS} = \sqrt{\frac{4.75088 + 11.17506 + 1.68545 + 1.29826 + 1.62388}{5}} = \sqrt{\frac{20..53353}{5}} = 2.02649$$

$$(1.25 - 2.14185)^2 = (-0.89185)^2 = 0.7954$$
$$(7 - 2.78937)^2 = (4.21063)^2 = 17.73745$$
$$(2.7 - 3.76065)^2 = (-1.06065)^2 = 1.12598$$
$$(3.2 - 4.73193)^2 = (-1.53193)^2 = 2.34681$$
$$(5.5 - 4.40817)^2 = (1.09183)^2 = 1.1921$$

$$\text{RMSE}_{Ridge} = \sqrt{\frac{0.7954 + 17.73745 + 1.12598 + 2.34681 + 1.1921 =}{5}} = \sqrt{\frac{4.63955}{5}} = 2.15454$$

Test samples:

$$(0.7 - 3.77081)^2 = (-3.07081)^2 = 9.42988$$
$$(1.1 - 3.54337)^2 = (-2.44337)^2 = 5.96807$$
$$(2.2 - 3.88453)^2 = (-1.68453)^2 = 2.83764$$

$$\text{RMSE}_{OLS} = \sqrt{\frac{9.42988 + 5.96807 + 2.837649}{3}} = \sqrt{\frac{18.235599}{3}} = 2.4656$$

$$(0.7 - 3.11313)^2 = (-2.41313)^2 = 5.82316$$
$$(1.1 - 2.46561)^2 = (-1.36561)^2 = 1.86489$$
$$(2.2 - 3.43689)^2 = (-1.23689)^2 = 1.53091$$

$$\text{RMSE}_{Ridge} = \sqrt{\frac{5.82316 + 1.86489 + 1.53091}{3}} = \sqrt{\frac{9.21896}{3}} = 1.7529$$

The RMSE results align with expectations for both OLS and Ridge regression models. OLS performs slightly better on the training samples than Ridge, which is expected since OLS minimizes the sum of squared residuals without any regularization. However, on the test set, OLS performs worse than Ridge, indicating that OLS overfits the training data and generalizes poorly to new data. Ridge regression, by minimizing the mean squared error while also reducing as much as possible the magnitude of the coefficients through regularization, sacrifices some training accuracy but achieves better generalization. Overall, these results demonstrate Ridge's ability to reduce overfitting and perform better than OLS on unseen data, as expected.

4) The first step would the pre-activation:

$$Z^{[1]} = W^{[1]} \times x_1 + b^{[1]}$$

$$Z^{[1]} = \begin{bmatrix} 0.1 & 0.1 \\ 0.1 & 0.2 \\ 0.2 & 0.1 \end{bmatrix} \times \begin{bmatrix} 1 \\ 1 \end{bmatrix} + \begin{bmatrix} 0.1 \\ 0 \\ 0.1 \end{bmatrix} = \begin{bmatrix} 0.3 \\ 0.3 \\ 0.4 \end{bmatrix}$$

And since there is no activation in the hidden layers:

$$Z^{[2]} = W^{[2]} \times Z^{[1]} + b^{[2]}$$

$$Z^{[2]} = \begin{bmatrix} 1 & 2 & 2 \\ 1 & 2 & 1 \\ 1 & 1 & 1 \end{bmatrix} \times \begin{bmatrix} 0.3 \\ 0.3 \\ 0.4 \end{bmatrix} + \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} = \begin{bmatrix} 2.7 \\ 2.3 \\ 2.0 \end{bmatrix}$$

Next, since there are only 2 layers in this neural network, the *softmax* function is applied to the output of layer 2 to obtain the class probabilities:

$$softmax\left(Z_c^{[out]}\right) = \frac{e^{Z_c^{[out]}}}{\sum_{l=1}^{|C|} e^{Z_l^{[out]}}}$$

$$softmax\left(Z_A^{[2]}\right) = \frac{e^{2.7}}{e^{2.7} + e^{2.3} + e^{2.0}} = \frac{14.8797}{14.8797 + 9.9742 + 7.3891} = \frac{14.8797}{32.2430} = 0.4614$$

$$softmax\left(Z_B^{[2]}\right) = \frac{9.9742}{32.2430} = 0.3094$$

$$softmax\left(Z_C^{[2]}\right) = \frac{7.3891}{32.2430} = 0.2292$$

The predicted probabilities for class A, B, and C are:

$$X^{[out]} = X^{[2]} = softmax\left(Z^{[2]}\right) = \begin{bmatrix} 0.4614 \\ 0.3094 \\ 0.2292 \end{bmatrix}$$

Since the true class is B, the target is $t = \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}$.

To calculate the cross-entropy loss,

$$E = -\sum_{i=1}^{n} \sum_{l=1}^{|C|} t_l^{(i)} \log\left(X_l^{[out](i)}\right)$$

since we are only using the training observation x_1,

$$E = -\sum_{l=1}^{3} t_l \log\left(X_l^{[out]}\right)$$

and only $t_2 = 1$:

$$E = -\log\left(X_2^{[2]}\right) = -log(0.3094) = 1.1737$$

Next, to perform the update on the weights and biases, first we need to calculate the gradients for the weights and biases of layer 2:

$$\frac{\partial E}{\partial W^{[l]}} = \delta^{[l]} \times (a^{[l-1]})^T$$

$$\frac{\partial E}{\partial b^{[l]}} = \delta^{[l]}$$

Aprendizagem 2024/25

Homework III – Group 028

(ist1106987, ist1107057)

$$\delta^{[2]} = \frac{\partial E}{\partial Z^{[2]}} = \frac{\partial E}{\partial X^{[2]}} \times \frac{\partial Y^{[2]}}{\partial Z^{[2]}} = \ldots some\ important\ steps \ldots = X^{[2]} - t = \begin{bmatrix} 0.4614 \\ 0.3094 \\ 0.2292 \end{bmatrix} - \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix} = \begin{bmatrix} 0.4614 \\ -0.6906 \\ 0.2292 \end{bmatrix}$$

$$\frac{\partial E}{\partial W^{[2]}} = \delta^{[2]} \times (Z^{[1]})^T = \begin{bmatrix} 0.4614 \\ -0.6906 \\ 0.2292 \end{bmatrix} \times [0.3 \quad 0.3 \quad 0.4] = \begin{bmatrix} 0.13842 & 0.13842 & 0.13842 \\ -0.20718 & -0.20718 & -0.20718 \\ 0.06876 & 0.06876 & 0.06876 \end{bmatrix}$$

$$\frac{\partial E}{\partial b^{[2]}} = \delta^{[2]} = \begin{bmatrix} 0.4614 \\ -0.6906 \\ 0.2292 \end{bmatrix}$$

$$\delta^{[1]} = (W^{[2]})^t \times \delta^{[2]} = \begin{bmatrix} 1 & 1 & 1 \\ 2 & 2 & 1 \\ 2 & 1 & 1 \end{bmatrix} \times \begin{bmatrix} 0.4614 \\ -0.6906 \\ 0.2292 \end{bmatrix} = \begin{bmatrix} 0 \\ -0.2292 \\ 0.4614 \end{bmatrix}$$

$$\frac{\partial E}{\partial W^{[1]}} = \delta^{[1]} \times (x_1)^T = \begin{bmatrix} 0 \\ -0.2292 \\ 0.4614 \end{bmatrix} \times [1 \quad 1] = \begin{bmatrix} 0 & 0 \\ -0.2292 & -0.2292 \\ 0.4614 & 0.4614 \end{bmatrix}$$

$$\frac{\partial E}{\partial b^{[1]}} = \delta^{[1]} = \begin{bmatrix} 0 \\ -0.2292 \\ 0.4614 \end{bmatrix}$$

Update weights and biases:

$$W^l = W^l - \eta \frac{\partial E}{\partial W^{[l]}}$$

$$b^l = b^l - \eta \frac{\partial E}{\partial b^{[l]}}$$

$$W^2 = W^2 - 0.1 \frac{\partial E}{\partial W^{[2]}} = \begin{bmatrix} 1 & 2 & 2 \\ 1 & 2 & 1 \\ 1 & 1 & 1 \end{bmatrix} - 0.1 \begin{bmatrix} 0.13842 & 0.13842 & 0.13842 \\ -0.20718 & -0.20718 & -0.20718 \\ 0.06876 & 0.06876 & 0.06876 \end{bmatrix}$$
$$= \begin{bmatrix} 0.98616 & 1.98616 & 1.981616 \\ 1.02072 & 2.02072 & 1.02072 \\ 0.99312 & 0.99312 & 0.99312 \end{bmatrix}$$

$$b^2 = b^2 - 0.1 \frac{\partial E}{\partial b^{[2]}} = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} - 0.1 \begin{bmatrix} 0.4614 \\ -0.6906 \\ 0.2292 \end{bmatrix} = \begin{bmatrix} 0.95386 \\ 1.06906 \\ 0.97708 \end{bmatrix}$$

$$W^1 = W^1 - 0.1 \frac{\partial E}{\partial W^{[1]}} = \begin{bmatrix} 0.1 & 0.1 \\ 0.1 & 0.2 \\ 0.2 & 0.1 \end{bmatrix} - 0.1 \begin{bmatrix} 0 & 0 \\ -0.2292 & -0.2292 \\ 0.4614 & 0.4614 \end{bmatrix} = \begin{bmatrix} 0.1 & 0.1 \\ 0.12292 & 0.22292 \\ 0.1538 & 0.0538 \end{bmatrix}$$

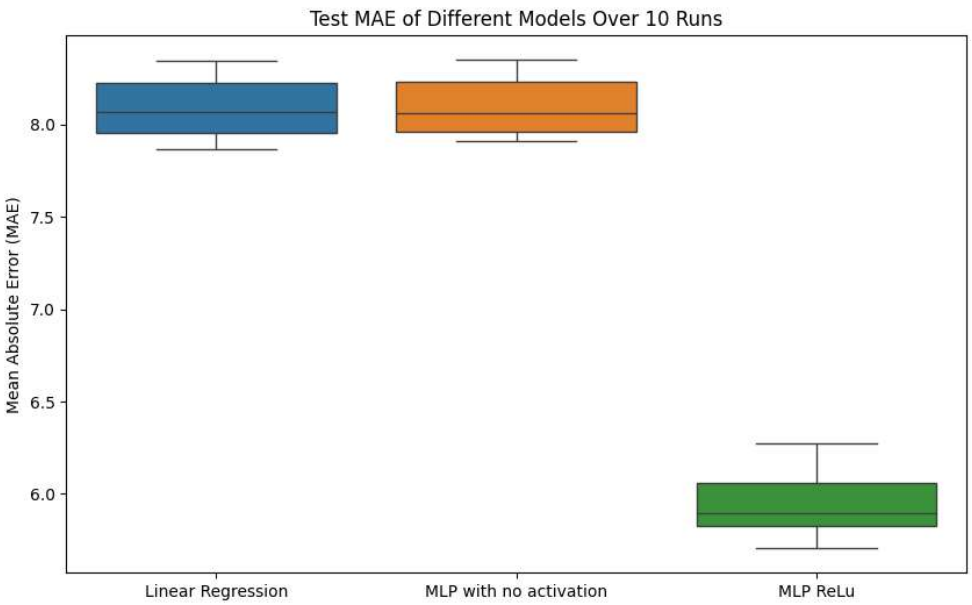
$$b^1 = b^1 - 0.1 \frac{\partial E}{\partial b^{[1]}} = \begin{bmatrix} 0.1 \\ 0 \\ 0.1 \end{bmatrix} - 0.1 \begin{bmatrix} 0 \\ -0.2292 \\ 0.4614 \end{bmatrix} = \begin{bmatrix} 0.1 \\ 0.02292 \\ 0.05386 \end{bmatrix}$$

Homework III – Group 028

(ist1106987, ist1107057)

II. Programming and critical analysis

5)



6)

From the observation of the boxplot above, we conclude that the test MAE from the Linear Regression and from the MLP with no activations are almost identical, with a median around 8. This similarity is expected, considering that an MLP with no activation means no nonlinearity is introduced in the model, which results in behavior like the linear model.

On the other hand, the MLP with ReLu activation resulted in a significant decrease of the test MAE, with a median around 5.9. This is due to the nonlinearity that the activation function introduces in the model, which allows the MLP to model complex relationships in the data and better capture its underlying structure, boosting the performance.

7)

The best combination of parameters is:

- L2 penalty = 0.01
- Batch size = 32
- Learning rate = 0.001

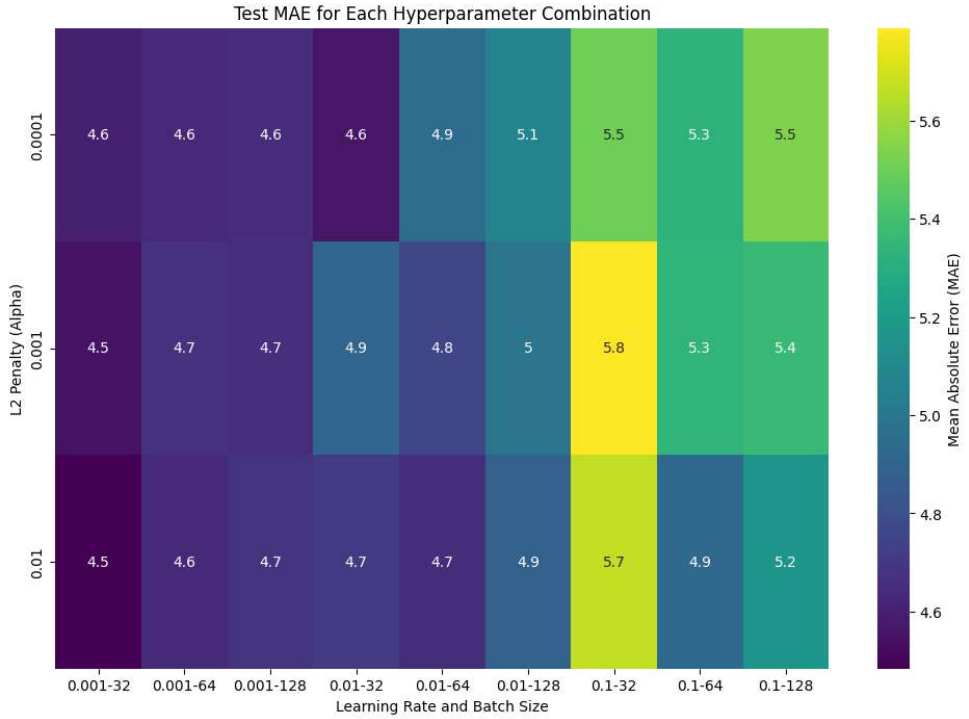
which results in a test MAE of 4.48387.

To better understand the trade-offs between the combinations, we plotted a heatmap that illustrates them.

Aprendizagem 2024/25

Homework III – Group 028

(ist1106987, ist1107057)



Concerning the L2 Penalty, we conclude that higher regularization (0.01) generally performs better, since it reduces overfitting in the model. Although, when combined with higher learning rates (especially 0.1) and a small batch size (32) it degrades performance, which is explained later in this answer.

Concerning the Learning Rate, we conclude that 0.001 provides better MAE across all combinations, considering it allows the model to take smaller steps during gradient descent, finding a more optimal solution, but leading to slower training.

Concerning the Batch Size, we conclude that a smaller batch size (32) generally leads to lower MAE in most combinations, particularly for low learning rates. This might be because a small batch size introduces more noise into the gradient estimates which offers a regularizing effect and helps the generalization ability of the model. But, when the learning rate is higher (the step size is bigger), a larger batch size (e.g. 64) results in more accurate estimates of the error gradient, which smooths out the updates and leads to a better MAE.

Taking this into account, when the Learning Rate is high (0.1) and the Batch Size is small (32), the higher L2 Penalties (0.001 and, especially, 0.01) degrade the MAE even further by applying too big of a regularization and, possibly, underfitting the model, which leads to a high MAE.

END