Aprendizagem 2022/23

Homework III

Deadline 28/10/2022 23:59 via Fenix as PDF

1)

Design Matrix :

$$X = \begin{array}{c} \text{Bias} \\ \downarrow \\ \end{array}$$

$$X = \begin{pmatrix} 1 & x_1^1 & x_1^2 & x_1^3 \\ 1 & x_2^1 & x_2^2 & x_2^3 \\ 1 & x_3^1 & x_3^2 & x_3^3 \\ 1 & x_4^1 & x_4^2 & x_4^3 \\ 1 & x_5^1 & x_5^2 & x_5^3 \end{pmatrix} = \begin{pmatrix} 1 & 0.8 & 0.64 & 0.512 \\ 1 & 1 & 1 & 1 \\ 1 & 1.2 & 1.44 & 1.728 \\ 1 & 1.4 & 1.96 & 2.744 \\ 1 & 1.6 & 2.56 & 4.096 \end{pmatrix}$$

$$X^T = \begin{pmatrix} 1 & 1 & 1 & 1 & 1 \\ 0.8 & 1 & 1.2 & 1.4 & 1.6 \\ 0.64 & 1 & 1.44 & 1.96 & 2.56 \\ 0.512 & 1 & 1.728 & 2.744 & 4.096 \end{pmatrix}$$

$$4 \times 5 \quad 5 \times 4$$

$$X^T X = \begin{pmatrix} 1 & 1 & 1 & 1 & 1 \\ 0.8 & 1 & 1.2 & 1.4 & 1.6 \\ 0.64 & 1 & 1.44 & 1.96 & 2.56 \\ 0.512 & 1 & 1.728 & 2.744 & 4.096 \end{pmatrix} \begin{pmatrix} 1 & 0.8 & 0.64 & 0.512 \\ 1 & 1 & 1 & 1 \\ 1 & 1.2 & 1.44 & 1.728 \\ 1 & 1.4 & 1.96 & 2.744 \\ 1 & 1.6 & 2.56 & 4.096 \end{pmatrix}$$

$$= \begin{pmatrix} 5 & 6 & 7.6 & 10.08 \\ 6 & 7.6 & 10.08 & 13.8784 \\ 7.6 & 10.08 & 13.8784 & 19.68 \\ 10.08 & 13.8784 & 19.68 & 28.55488 \end{pmatrix}$$

$$(X^T X + \lambda I)^{-1} = \left[\begin{pmatrix} 5 & 6 & 7.6 & 10.08 \\ 6 & 7.6 & 10.08 & 13.8784 \\ 7.6 & 10.08 & 13.8784 & 19.68 \\ 10.08 & 13.8784 & 19.68 & 28.55488 \end{pmatrix} + \begin{pmatrix} 2 & 0 & 0 & 0 \\ 0 & 2 & 0 & 0 \\ 0 & 0 & 2 & 0 \\ 0 & 0 & 0 & 2 \end{pmatrix}\right]^{-1}$$

$$= \begin{pmatrix} 7 & 6 & 7.6 & 10.08 \\ 6 & 9.6 & 10.08 & 13.8784 \\ 7.6 & 10.08 & 15.8784 & 19.68 \\ 10.08 & 13.8784 & 19.68 & 30.55488 \end{pmatrix}^{-1}$$

$$= \begin{pmatrix} 0.3417 & -0.1214 & -0.0749 & -0.0093 \\ -0.1214 & 0.3892 & -0.0967 & -0.0745 \\ -0.0749 & -0.0967 & 0.3726 & -0.1714 \\ -0.0093 & -0.0745 & -0.1714 & 0.18 \end{pmatrix}$$

$$(X^T X + \lambda I)^{-1} X^T = \begin{pmatrix} 0.3417 & -0.1214 & -0.0749 & -0.0093 \\ -0.1214 & 0.3892 & -0.0967 & -0.0745 \\ -0.0749 & -0.0967 & 0.3726 & -0.1714 \\ -0.0093 & -0.0745 & -0.1714 & 0.18 \end{pmatrix} \begin{pmatrix} 1 & 1 & 1 & 1 & 1 \\ 0.8 & 1 & 1.2 & 1.4 & 1.6 \\ 0.64 & 1 & 1.44 & 1.96 & 2.56 \\ 0.512 & 1 & 1.728 & 2.744 & 4.096 \end{pmatrix}$$

$$= \begin{pmatrix} 0.1918 & 0.136 & 0.072 & -0.0007 & -0.082 \\ 0.9 & 0.0966 & 0.0777 & 0.0296 & -0.051 \\ -0.001 & 0.0296 & 0.0495 & 0.05 & 0.0223 \\ -0.086 & -0.075 & -0.034 & 0.0444 & 0.1701 \end{pmatrix}$$

$$w = (X^T X + \lambda I)^{-1} X^T Z = \begin{pmatrix} 0.1918 & 0.136 & 0.072 & -0.0007 & -0.082 \\ 0.9 & 0.0966 & 0.0777 & 0.0296 & -0.051 \\ -0.001 & 0.0296 & 0.0495 & 0.05 & 0.0223 \\ -0.086 & -0.075 & -0.034 & 0.0444 & 0.1701 \end{pmatrix} \begin{pmatrix} 24 \\ 20 \\ 10 \\ 13 \\ 12 \end{pmatrix}$$

$$= \begin{pmatrix} 7.0451 \\ 4.6409 \\ 1.9673 \\ -1.301 \end{pmatrix}$$

2) $\hat{z}(x,w) = w_0 + w_1 x + w_2 x^2 + w_3 x^3$

$$= 7.0451 + 4.6409\,x + 1.9673\,x^2 - 1.301\,x^3$$

$$\hat{z}_0 = \hat{z}(x_0, w) = 7.0451 + 4.6409 \times 0.8 + 1.9673 \times 0.8^2 - 1.301 \times 0.8^3$$
$$= 11.35078$$

$$\hat{z}_1 = \hat{z}(x_1, w) = 7.0451 + 4.6409 \times 1 + 1.9673 \times 1^2 - 1.301 \times 1^3$$
$$= 12.3523$$

$$\hat{z}_2 = \hat{z}(x_2, w) = 7.0451 + 4.6409 \times 1.2 + 1.9673 \times 1.2^2 - 1.301 \times 1.2^3$$
$$= 13.199$$

$$\hat{z}_3 = \hat{z}(x_3, w) = 7.0451 + 4.6409 \times 1.4 + 1.9673 \times 1.4^2 - 1.301 \times 1.4^3$$
$$= 13.8283$$

$$\hat{z}_4 = \hat{z}(x_4, w) = 7.0451 + 4.6409 \times 1.6 + 1.9673 \times 1.6^2 - 1.301 \times 1.6^3$$
$$= 14.178$$

$$RMSE = \sqrt{\sum_{i=0}^{4} \frac{(\hat{z}_i - z_i)^2}{5}}$$

$$= \sqrt{\frac{1}{5}\left((\hat{z}_0 - z_0)^2 + (\hat{z}_1 - z_1)^2 + (\hat{z}_2 - z_2)^2 + (\hat{z}_3 - z_3)^2 + (\hat{z}_4 - z_4)^2\right)}$$

$$= 6.8433$$

3) 1° • Forward Propagation

$$\rightarrow W^{[1]} = \begin{bmatrix} 1 \\ 1 \end{bmatrix} \quad \rightarrow b^{[1]} = \begin{bmatrix} 1 \\ 1 \end{bmatrix}$$

$$\rightarrow w^{[2]} = \begin{bmatrix} 1 & 1 \end{bmatrix} \rightarrow b^{[2]} = 1$$

$$\rightarrow z^{[i]} = w^{[i]} x^{[i-1]} + b^{[1]}$$

$$\rightarrow x^{[i]} = f(z^{[i]}) = e^{0.1 z^{[i]}}$$

$$x_0^{[0]} = 0.8$$

$$z_0^{[1]} = w_0^{[1]} x_0^{[0]} + b_0^{[1]}$$

$$= \begin{bmatrix} 1 \\ 1 \end{bmatrix} 0.8 + \begin{bmatrix} 1 \\ 1 \end{bmatrix}$$

$$= \begin{bmatrix} 1.8 \\ 1.8 \end{bmatrix}$$

$$x_0^{[1]} = f(z_0^{[1]})$$

$$= \begin{bmatrix} 1.197 \\ 1.197 \end{bmatrix}$$

$$z_0^{[2]} = w_0^{[2]} x_0^{[1]} + b_0^{[2]}$$

$$= \begin{bmatrix} 1 & 1 \end{bmatrix} \begin{bmatrix} 1.197 \\ 1.197 \end{bmatrix} + 1$$

$$= 3.394$$

$$x_0^{[2]} = f(z_0^{[2]})$$

$$= 1.4041$$

$$x_1^{[0]} = 1$$

$$z_1^{[1]} = w_1^{[1]} x_1^{[0]} + b_1^{[1]}$$

$$= \begin{bmatrix} 1 \\ 1 \end{bmatrix} 1 + \begin{bmatrix} 1 \\ 1 \end{bmatrix}$$

$$= \begin{bmatrix} 2 \\ 2 \end{bmatrix}$$

$$x_1^{[1]} = f(z_1^{[1]})$$

$$= \begin{bmatrix} 1.221 \\ 1.221 \end{bmatrix}$$

$$z_1^{[2]} = w_1^{[2]} x_1^{[1]} + b_1^{[2]}$$

$$= \begin{bmatrix} 1 & 1 \end{bmatrix} \begin{bmatrix} 1.221 \\ 1.221 \end{bmatrix} + 1$$

$$= 3.442$$

$$x_1^{[2]} = f(z_1^{[2]})$$

$$= 1.411$$

$$x_2^{[0]} = 1.2$$

$$z_2^{[1]} = w_2^{[1]} x_2^{[0]} + b_2^{[1]}$$

$$= \begin{bmatrix} 1 \\ 1 \end{bmatrix} 1.2 + \begin{bmatrix} 1 \\ 1 \end{bmatrix}$$

$$= \begin{bmatrix} 2.2 \\ 2.2 \end{bmatrix}$$

$$x_2^{[1]} = f(z_2^{[1]})$$

$$= \begin{bmatrix} 1.246 \\ 1.246 \end{bmatrix}$$

$$z_2^{[2]} = w_2^{[2]} x_2^{[1]} + b_2^{[2]}$$

$$= \begin{bmatrix} 1 & 1 \end{bmatrix} \begin{bmatrix} 1.246 \\ 1.246 \end{bmatrix} + 1$$

$$= 3.492$$

$$x_2^{[2]} = f(z_2^{[2]})$$

$$= 1.418$$

2° - Backward Propagation

$$\rightarrow x_0^{[2]} = 1.4041 \quad \rightarrow x_1^{[2]} = 1.411 \quad \rightarrow x_2^{[2]} = 1.418$$

$$\rightarrow T_0 = 24 \quad \rightarrow t_1 = 20 \quad \rightarrow t_2 = 10$$

$$\rightarrow E_j(w^{[i]}) = \frac{1}{2}\left(x_j^{[i]} - t_j\right)^2$$

$$\rightarrow f'(u) = 0.1 e^{0.1u}$$

$$\rightarrow x^{[i]} = f(z^{[i]})$$

$$\Rightarrow \frac{\partial x^{[i]}}{\partial z^{[i]}} = f'(z^{[i]}) \cdot z^{[i]'}$$

$$= 0.1 e^{0.1 z^{[i]}}$$

$$\rightarrow \delta_j^{[i]} = \frac{\partial E}{\partial x^{[i]}} \circ \frac{\partial x^{[i]}}{\partial z^{[i]}}$$

$$= \left(x_j^{[i]} - t_j\right) \circ 0.1 e^{0.1 z_j^{[i]}}$$

$$\rightarrow \delta_j^{[i]} = w^{[i+1]^T} \cdot \delta_j^{[i+1]} \circ 0.1 e^{0.1 z_j^{[i]}}$$

$$\rightarrow \delta^{[i]} = \sum_{j=0}^{2} \delta_j^{[i]}$$

$$\rightarrow \frac{\partial E}{\partial w^{[i]}} = \sum_{j=0}^{2} \delta_j^{[i]} \cdot \left(\frac{\partial z_j^{[i]}}{\partial w^{[i]}}\right)^T$$

$$= \sum_{j=0}^{2} \delta_j^{[i]} \cdot \left(x_j^{[i-1]}\right)^T$$

$$\rightarrow w^{[k]} = w^{[k]} - n \frac{\partial E}{\partial w^{[i]}}$$

$$\rightarrow b^{[k]} = b^{[k]} - n\, \delta^{[k]}$$

Calculations

Level 2

$$\delta_0^{[2]} = (1.4041 - 24) \times 0.1 e^{0.1 \times 3.394}$$

$$= -3.1727$$

$$\delta_1^{[2]} = (1.411 - 20) \times 0.1 e^{0.1 \times 3.442}$$

$$= -2.623$$

$$\delta_2^{[2]} = (1.418 - 10) \times 0.1 e^{0.1 \times 3.492}$$

$$= -1.217$$

$$w^{[2]} = w^{[2]} - \eta \boxed{\frac{\partial E}{\partial w^{[2]}}}$$

$$= \sum \delta_i^{[2]} \cdot x_i^{[1]^T}$$

$$= -3.1727 \begin{bmatrix} 1.197 \\ 1.197 \end{bmatrix}^T - 2.623 \begin{bmatrix} 1.221 \\ 1.221 \end{bmatrix}^T - 1.217 \begin{bmatrix} 1.246 \\ 1.246 \end{bmatrix}^T$$

$$= \begin{bmatrix} -8.517 \\ -8.517 \end{bmatrix}^T$$

$$= \begin{bmatrix} 1 & 1 \end{bmatrix} - 0.1 \begin{bmatrix} -8.517 & -8.517 \end{bmatrix}$$

$$= \begin{bmatrix} 1.8517 & 1.8517 \end{bmatrix}$$

$$b^{[2]} = b^{[2]} - \eta \, \delta^{[2]}$$

$$= 1 - 0.1 \left(-3.1727 - 2.623 - 1.217 \right)$$

$$= 1.7013$$

Level 1

$$\delta_0^{[1]} = \begin{bmatrix} 1 \\ 1 \end{bmatrix} \times -3.1727 \circ \begin{bmatrix} 0.1 e^{0.1 \times 1.8} \\ 0.1 e^{0.1 \times 1.8} \end{bmatrix}$$

$$= \begin{bmatrix} -0.38 \\ -0.38 \end{bmatrix}$$

$$\delta_1^{[1]} = \begin{bmatrix} 1 \\ 1 \end{bmatrix} \times -2.623 \circ \begin{bmatrix} 0.1 e^{0.1 \times 2} \\ 0.1 e^{0.1 \times 2} \end{bmatrix}$$

$$= \begin{bmatrix} -0.32 \\ -0.32 \end{bmatrix}$$

$$\delta_2^{[1]} = \begin{bmatrix} 1 \\ 1 \end{bmatrix} \times -1.217 \circ \begin{bmatrix} 0.1 e^{0.1 \times 2.2} \\ 0.1 e^{0.1 \times 2.2} \end{bmatrix}$$

$$= \begin{bmatrix} -0.15 \\ -0.15 \end{bmatrix}$$

$$w^{[1]} = w^{[1]} - \eta \boxed{\frac{\partial E}{\partial w^{[1]}}}$$

$$\frac{\partial E}{\partial w^{[1]}} = \sum \delta^{[1]} \cdot x^{[0]^T}$$

$$= \begin{bmatrix} -0.38 \\ -0.38 \end{bmatrix} \times 0.8 + \begin{bmatrix} -0.32 \\ -0.32 \end{bmatrix} \times 1 + \begin{bmatrix} -0.15 \\ -0.15 \end{bmatrix} \times 1.2$$

$$= \begin{bmatrix} -0.804 \\ -0.804 \end{bmatrix}$$

$$= \begin{bmatrix} 1 \\ 1 \end{bmatrix} - 0.1 \begin{bmatrix} -0.804 \\ -0.804 \end{bmatrix}$$

$$= \begin{bmatrix} 1.0804 \\ 1.0804 \end{bmatrix}$$

$$b^{[1]} = b^{[1]} - \eta\, \delta^{[1]}$$

$$= \begin{bmatrix} 1 \\ 1 \end{bmatrix} - 0.1 \left(\begin{bmatrix} -0.38 \\ -0.38 \end{bmatrix} + \begin{bmatrix} -0.32 \\ -0.32 \end{bmatrix} + \begin{bmatrix} -0.15 \\ -0.15 \end{bmatrix} \right)$$

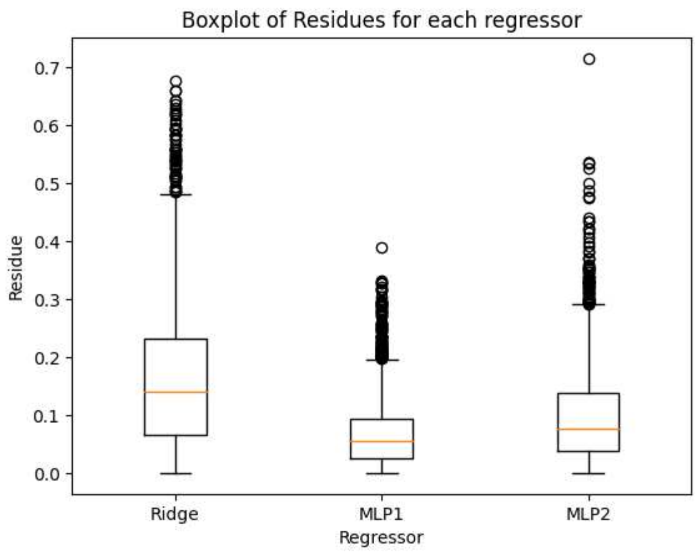
$$= \begin{bmatrix} 1.0805 \\ 1.0805 \end{bmatrix}$$

II. Programming and critical analysis

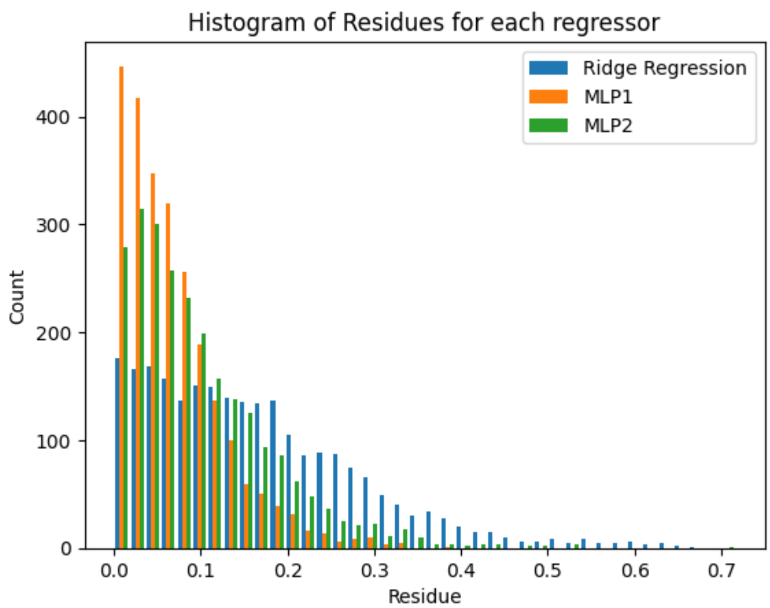
The code for the questions is in the Appendix of this document.

4) Ridge Regression MAE: 0.162829976437694
   MAE for MLP1: 0.0680414073796843
   MAE for MLP2: 0.0978071820387748

5) Resulting Boxplot:



Resulting Histogram:



6) MLP1 iterations: 452

   MLP2 iterations: 77

7) The number of iterations required for MLP1 and MLP2 to converge is very different (452 vs 77).

   This is because MLP1 uses early stopping and MLP2 does not. Early stopping is a method that stops the training when the validation score is not improving by at least e^{-4} {1} for 10 {2} consecutive epochs and it is done to avoid overfitting.

   MLP2 doesn't use early stopping, so it will continue to train until it reaches the maximum number of iterations (500) and, therefore, it is more likely to overfit the data.

   This is why the number of iterations required for MLP2 to converge is much lower than the number of iterations required for MLP1 to converge and, since MLP2 has a higher MAE and is more overfitted than MLP1, we can conclude that MLP1 has a better performance than MLP2.

{1} - default **`tol`** (tolerance) in MLPRegressor function.

{2} - default maximum number of epochs to not meet **`tol`** improvement in MLPRegressor function (n_iter_no_change).

III. APPENDIX

Code used in question 4) of **II. Programming and critical analysis**:

Loads the Parkinson Disease's Data:

```
import pandas as pd
import numpy as np
from sklearn.linear_model import LinearRegression, Ridge
from sklearn.neural_network import MLPRegressor
from sklearn.metrics import mean_absolute_error
from scipy.io.arff import loadarff
from sklearn import model_selection
from sklearn import metrics

#load data
data = loadarff('kin8nm.arff')
df = pd.DataFrame(data[0])
df.head()
```

Calculates the mean absolute error of the Ridge Regression:

```
#split data
X = df.drop('y', axis=1)
y = df['y']
X_train, X_test, y_train, y_test = model_selection.train_test_split(X.values,
      y.values, train_size=0.7, random_state=0)

#train models and get MAE for Ridge
rr = Ridge(alpha=0.1)
rr.fit(X_train, y_train)
pint('Ridge Regression MAE: ', mean_absolute_error(y_test, rr.predict(X_test)))
```

Calculates the mean absolute errors of the Multi-Layer Perceptrons:

```
#train models and get MAE for MLPs
mlp1 = MLPRegressor(hidden_layer_sizes=(10, 10), activation='tanh',
      max_iter=500, random_state=0, early_stopping=True)
mlp2 = MLPRegressor(hidden_layer_sizes=(10, 10), activation='tanh',
      max_iter=500, random_state=0)

mlp1.fit(X_train, y_train)
print("MAE for MLP1: ", mean_absolute_error(y_test, mlp1.predict(X_test)))
mlp2.fit(X_train, y_train)
print("MAE for MLP2: ", mean_absolute_error(y_test, mlp2.predict(X_test)))
```

Code used in question 5) of **II. Programming and critical analysis**:

```
#plots
import matplotlib.pyplot as plt

#Plot the residues (in absolute value) using two visualizations: boxplots and histograms.

#boxplot
bp = plt.boxplot([abs(y_test - rr.predict(X_test)), abs(y_test -
      mlp1.predict(X_test)), abs(y_test - mlp2.predict(X_test))])
plt.xticks([1, 2, 3], ['Ridge', 'MLP1', 'MLP2'])
plt.ylabel('Residue')
plt.xlabel('Regressor')
plt.title('Boxplot of Residues for each regressor')
plt.show()

#histogram
plt.hist([abs(y_test - rr.predict(X_test)), abs(y_test - mlp1.predict(X_test)), abs(y_test - mlp2.predict(X_test))], bins=40,
      label=['Ridge Regression', 'MLP1', 'MLP2'])
plt.legend()
plt.title('Histogram of Residues for each regressor')
plt.ylabel('Count')
plt.xlabel('Residue')
plt.show()
```

Code used in question 6) of **II. Programming and critical analysis**:

```
#iterations required for MLP1 and MLP2 to converge
print('MLP1 iterations: ', mlp1.n_iter_)
print('MLP2 iterations: ', mlp2.n_iter_)
```