Aprendizagem 2022/23

Homework III – Group 015

I. Pen-and-paper

1)

Consider the problem of learning a regression model from 5 univariate observations $((0.8),(1),(1.2),(1.4),(1.6))$ with targets $(24,20,10,13,12)$.

1) [5v] Consider the basis function, $\phi_j(x) = x^j$, for performing a 3-order polynomial regression,

$$\hat{z}(x,\mathbf{w}) = \sum_{j=0}^{3} w_j \phi_j(x) = w_0 + w_1 x + w_2 x^2 + w_3 x^3 .$$

Learn the Ridge regression (l_2 regularization) on the transformed data space using the closed form solution with $\lambda = 2$.

Hint: use numpy matrix operations (e.g., `linalg.pinv` for inverse) to validate your calculus.

	y_1	out
x_1	0.8	24
x_2	1	20
x_3	1.2	10
x_4	1.4	13
x_5	1.6	12

$$\phi_j(x) = x^j \qquad \Phi_{ij} = \phi_j(x_i)$$

$$\hat{z} = \Phi w$$

$$\hat{z} = w^T \phi(x)$$

$$\hat{z}(x,w) = \sum_{j=0}^{3} w_j \phi_j(x) = w_0 + w_1 x + w_2 x^2 + w_3 x^3$$

$$X = \begin{bmatrix} 1 & 0.8 \\ 1 & 1 \\ 1 & 1.2 \\ 1 & 1.4 \\ 1 & 1.6 \end{bmatrix} \qquad \Phi = \begin{bmatrix} 1 & \phi_1(0.8) & \phi_2(0.8) & \phi_3(0.8) \\ 1 & \phi_1(1) & \phi_2(1) & \phi_3(1) \\ 1 & \phi_1(1.2) & \phi_2(1.2) & \phi_3(1.2) \\ 1 & \phi_1(1.4) & \phi_2(1.4) & \phi_3(1.4) \\ 1 & \phi_1(1.6) & \phi_2(1.6) & \phi_3(1.6) \end{bmatrix}$$

$$\Phi = \begin{bmatrix} 1 & 0.8 & 0.8^2 & 0.8^3 \\ 1 & 1 & 1^2 & 1^3 \\ 1 & 1.2 & 1.2^2 & 1.2^3 \\ 1 & 1.4 & 1.4^2 & 1.4^3 \\ 1 & 1.6 & 1.6^2 & 1.6^3 \end{bmatrix} = \begin{bmatrix} 1 & 0.8 & 0.64 & 0.512 \\ 1 & 1 & 1 & 1 \\ 1 & 1.2 & 1.44 & 1.728 \\ 1 & 1.4 & 1.96 & 2.744 \\ 1 & 1.6 & 2.56 & 4.096 \end{bmatrix}$$

$$\boxed{w = (\Phi^T \Phi + \lambda I)^{-1} \Phi^T z}$$

$$\lambda = 2$$

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

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

$$\lambda I = 2 \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 2 & 0 & 0 & 0 \\ 0 & 2 & 0 & 0 \\ 0 & 0 & 2 & 0 \\ 0 & 0 & 0 & 2 \end{bmatrix}$$

$$\Phi^T \Phi + \lambda I = \begin{bmatrix} 7 & 6 & 7.6 & 10.08 \\ 6 & 9.6 & 10.08 & 13.8784 \\ 7.6 & 10.08 & 15.8784 & 19.68 \\ 10.08 & 13.8784 & 19.68 & 30.55488 \end{bmatrix}$$

$$
\left(\Phi^T \Phi + \lambda I\right)^{-1} = \begin{bmatrix} 0.34168753 & -0.1214259 & -0.07490231 & -0.00932537 \\ -0.1214259 & 0.3892078 & -0.09667718 & -0.07445624 \\ -0.07490231 & -0.09667718 & 0.37257788 & -0.17135047 \\ -0.00932537 & -0.07445624 & -0.17135047 & 0.17998796 \end{bmatrix}
$$

$$
\left(\Phi^T \Phi + \lambda I\right)^{-1} \Phi^T = \begin{bmatrix} 0.19183474 & 0.13603395 & 0.07200288 & -0.00070608 & -0.08254055 \\ 0.08994535 & 0.09664848 & 0.07747793 & 0.02966982 & -0.05115977 \\ -0.00152564 & 0.02964793 & 0.04950363 & 0.04981662 & 0.02236208 \\ -0.08640083 & -0.07514413 & -0.03439835 & 0.04447593 & 0.17011812 \end{bmatrix}
$$

$$
\left(\Phi^T \Phi + \lambda I\right)^{-1} \Phi^T z = W = \begin{array}{cc} \begin{bmatrix} 7.0450759 \\ 4.64092765 \\ 1.96734046 \\ -1.30088142 \end{bmatrix} & \begin{array}{c} w_0 \\ w_1 \\ w_2 \\ w_3 \end{array} \end{array}
$$

This is the Ridge regression on the transformed data space using the closed form solution with $\lambda = 2$

$$
\hat{z}(x, w) = \sum_{j=0}^{3} w_j \phi_j(x) = w_0 + w_1 x + w_2 x^2 + w_3 x^3
$$

$$
\hat{z}(x, w) = 7.0450759 + 4.64092765 x + 1.96734046 x^2 - 1.30088142 x^3
$$

Aprendizagem 2022/23

Homework III – Group 015

2) Answer 2

2) [1v] Compute the training RMSE for the learnt regression model.

$$\boxed{RMSE = \sqrt{\frac{\sum_{i=1}^{m} (z_i - \hat{z}_i)^2}{m}}}$$

$\rightarrow \hat{z}_1 = 7.0450759 + 4.66092765 \times 0.8 + 1.96734046 \times 0.8^2 - 1.30088142 \times 0.8^3$

$\hat{z}_1 = 11{,}35086463$

$\rightarrow \hat{z}_2 = 7.0450759 + 4.66092765 \times 1 + 1.96734046 \times 1^2 - 1.30088142 \times 1^3$

$\hat{z}_2 = 12.35246259$

$\rightarrow \hat{z}_3 = 7.0450759 + 4.66092765 \times 1.2 + 1.96734046 \times 1.2^2 - 1.30088142 \times 1.2^3$

$\hat{z}_3 = 13{,}19923625$

$\rightarrow \hat{z}_4 = 7.0450759 + 4.66092765 \times 1.4 + 1.96734046 \times 1.4^2 - 1.30088142 \times 1.4^3$

$\hat{z}_4 = 13.8287433$

$\rightarrow \hat{z}_5 = 7.0450759 + 4.66092765 \times 1.6 + 1.96734046 \times 1.6^2 - 1.30088142 \times 1.6^3$

$\hat{z}_5 = 14.17854142$

$$RMSE = \sqrt{\frac{\sum_{i=1}^{5} (z_i - \hat{z}_i)^2}{m}}$$

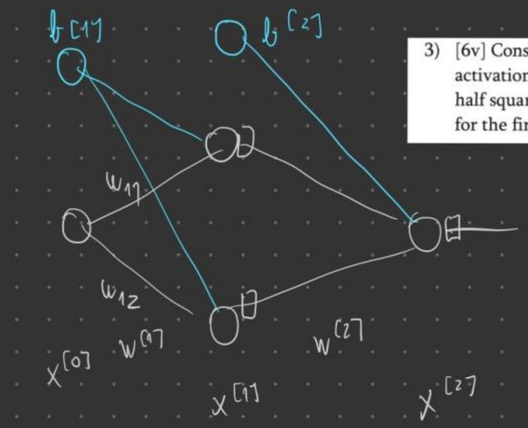
$$= \sqrt{\frac{(24-11{,}35086463)^2 + (20-12.35246259)^2 + (10-13{,}19923625)^2 + (13-13.8287433)^2 + (12-14.17854142)^2}{5}}$$

$= 6{,}843294891$

Aprendizagem 2022/23

Homework III – Group 015

3) Answer 3



3) [6v] Consider a multi-layer perceptron characterized by one hidden layer with 2 nodes. Using the activation function $f(x) = e^{0.1x}$ on all units, all weights initialized as 1 (including biases), and the half squared error loss, perform one batch gradient descent update (with learning rate $\eta = 0.1$) for the first three observations (0.8), (1) and (1.2).

$$\boxed{x^{[0]} = \begin{bmatrix} 0.8 \end{bmatrix}} \qquad b^{[1]} = \begin{bmatrix} 1 \\ 1 \end{bmatrix} \qquad W^{[1]} = \begin{bmatrix} 1 \\ 1 \end{bmatrix}$$

$$z^{[1]} = W^{[1]} x^{[0]} + b^{[1]}$$

$$z^{[1]} = \begin{bmatrix} 1 \\ 1 \end{bmatrix} \begin{bmatrix} 0.8 \end{bmatrix} + \begin{bmatrix} 1 \\ 1 \end{bmatrix}$$

$$= \begin{bmatrix} 0.8 \\ 0.8 \end{bmatrix} + \begin{bmatrix} 1 \\ 1 \end{bmatrix} = \begin{bmatrix} 1.8 \\ 1.8 \end{bmatrix}$$

$$x^{[1]} = \begin{bmatrix} e^{0.1 \times 1.8} \\ e^{0.1 \times 1.8} \end{bmatrix} = \begin{bmatrix} e^{0.18} \\ e^{0.18} \end{bmatrix}$$

$$z^{[2]} = W^{[2]} x^{[1]} + b^{[2]}$$

$$= \begin{bmatrix} 1 & 1 \end{bmatrix} \begin{bmatrix} e^{0.18} \\ e^{0.18} \end{bmatrix} + 1 = 2e^{0.18} + 1$$

$$x^{[2]} = e^{0.1 \times (2e^{0.18}+1)} \approx \boxed{1,4042}$$

$$\boxed{x^{[0]} = 1}$$

$$z^{[1]} = W^{[1]} x^{[0]} + b^{[1]}$$

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

$$= \begin{bmatrix} 1 \\ 1 \end{bmatrix} + \begin{bmatrix} 1 \\ 1 \end{bmatrix} = \begin{bmatrix} 2 \\ 2 \end{bmatrix}$$

$$x^{[1]} = \begin{bmatrix} e^{0.2} \\ e^{0.2} \end{bmatrix}$$

$$z^{[2]} = W^{[2]} x^{[1]} + b^{[2]}$$

$$= \begin{bmatrix} 1 & 1 \end{bmatrix} \begin{bmatrix} e^{0.2} \\ e^{0.2} \end{bmatrix} + 1$$

$$= 2e^{0.2} + 1$$

$$x^{[2]} = e^{0.1 (2e^{0.2}+1)} \approx \boxed{1.4110}$$

$$\boxed{x^{[0]} = 1.2}$$

$$z^{[1]} = W^{[1]} x^{[0]} + b^{[1]}$$

$$= \begin{bmatrix} 1 \\ 1 \end{bmatrix} \begin{bmatrix} 1.2 \end{bmatrix} + \begin{bmatrix} 1 \\ 1 \end{bmatrix} = \begin{bmatrix} 2.2 \\ 2.2 \end{bmatrix}$$

$$x^{[1]} = \begin{bmatrix} e^{0.1 \times 2.2} \\ e^{0.1 \times 2.2} \end{bmatrix} = \begin{bmatrix} e^{0.22} \\ e^{0.22} \end{bmatrix}$$

$$z^{[2]} = W^{[2]} x^{[1]} + b^{[2]}$$

$$= \begin{bmatrix} 1 & 1 \end{bmatrix} \begin{bmatrix} e^{0.22} \\ e^{0.22} \end{bmatrix} + 1$$

$$= 2e^{0.22} + 1$$

$$x^{[2]} = e^{0.1\left(2e^{0.22}+1\right)} \approx \boxed{1.4180}$$

$$E(w) = \frac{1}{2} \sum \left(x^{[2]} - t \right)^2$$

$\boxed{\text{Para } W^{[2]}}$

$$\frac{\partial E}{\partial W^{[2]}} = \sum_i \Bigg(\underbrace{\frac{\partial E}{\partial x_i^{[2]}} \circ \frac{\partial x_i^{[2]}}{\partial z_i^{[2]}}}_{\delta_i^{[2]}} \cdot \Big(\underbrace{\frac{\partial z_i^{[2]}}{\partial W^{[2]}}}_{(x_i^{[1]})^T} \Big)^T \Bigg)$$

$$f(x) = e^{0.1x}$$
$$f'(x) = 0.1 e^{0.1x}$$

δ da última camada

$$\boxed{\delta^{[p]} = \left(v^{[p]} - z \right) \circ \phi'^{[p]}\left(z^{[p]} \right)}$$

δ das camadas anteriores

$$\boxed{\delta^{[p]} = \left(W^{[p+1]} \delta^{[p+1]} \right) \circ \phi'^{[p]}\left(z^{[p]} \right)}$$

$$\boxed{\delta_1^{[2]}}$$

$$\delta_1^{[2]} = (1.4042 - 24) \circ \left(0.1\, e^{0.1 \times (2e^{0.18}+1)}\right)$$

$$\delta_1^{[2]} = -3.1728$$

$$\boxed{\delta_2^{[2]}}$$

$$\delta_2^{[2]} = (1.4110 - 20) \circ \left(0.1\, e^{0.1 \times (2e^{0.2}+1)}\right)$$

$$\delta_2^{[2]} = -2,6229$$

$$\boxed{\delta_3^{[2]}}$$

$$\delta_3^{[2]} = (1.4180 - 10) \circ \left(0.1\, e^{0.1 \times (2e^{0.22}+1)}\right)$$

$$\delta_3^{[2]} = -1,2169$$

$$\boxed{W^{[2]} = W^{[2]} - \eta \nabla E}$$

$$\frac{\delta E}{\delta W^{[2]}} = \sum_{i=1}^{3} \left(\delta_i^{[2]} \cdot (X_i^{[1]})^T \right)$$

$$\frac{\partial E}{\partial w^{[2]}} \simeq -3.1728 \cdot \begin{bmatrix} e^{0.18} & e^{0.18} \end{bmatrix} - 2.6229 \cdot \begin{bmatrix} e^{0.2} & e^{0.2} \end{bmatrix} - 1.2169 \cdot \begin{bmatrix} e^{0.22} & e^{0.22} \end{bmatrix}$$

$$= \begin{bmatrix} -8{,}5185 & -8{,}5185 \end{bmatrix}$$

$$W^{[2]} = \begin{bmatrix} 1 & 1 \end{bmatrix} - 0.1 \begin{bmatrix} -8{,}5185 & -8{,}5185 \end{bmatrix}$$

$$\boxed{W^{[2]} = \begin{bmatrix} 1{,}85185 & 1{,}85185 \end{bmatrix}}$$

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

$$z^{[2]} = W^{[2]} x^{[1]} + b^{[2]}$$

$$\frac{\partial E}{\partial b^{[2]}} = \sum_i \left(\underbrace{\frac{\partial E}{\partial x_i^{[2]}} \circ \frac{\partial x_i^{[2]}}{\partial z_i^{[2]}}}_{\delta_i^{[2]}} \cdot \underbrace{\frac{\partial z^{[2]}}{\partial b^{[2]}}}_{1} \right)$$

$$\frac{\partial E}{\partial b^{[2]}} = \sum_{i=1}^{3} \left(\delta_i^{[2]} \right) = -3.1728 - 2.6229 - 1.2169$$

$$= -7.0126$$

$$b^{[2]} = 1 - 0.1 \times (-7.0126)$$

$$\boxed{b^{[2]} = 1.70126}$$

$$\boxed{W^{[1]} = W^{[1]} - \eta \nabla E}$$

$$\frac{\partial E}{\partial W^{[1]}} = \sum_i \left(\underbrace{\frac{\partial E}{\partial X_i^{[1]}} \circ \frac{\partial X_i^{[1]}}{\partial Z_i^{[1]}}}_{\delta_i^{[1]}} \cdot \underbrace{\left(\frac{\partial Z_i^{[1]}}{\partial W^{[1]}} \right)^T}_{(X_i^{[0]})^T} \right)$$

δ das camadas anteriores

$$\boxed{\delta^{[p]} = \left(W^{[p+1]} \delta^{[p+1]} \right) \circ \phi'^{[p]}\left(z^{[p]} \right)}$$

$$\delta_i^{[1]} = \left(\left(W^{[2]}\right)^T \delta_i^{[2]} \right) \circ f'\left(z_i^{[1]} \right)$$

$$\boxed{\delta_1^{[1]}}$$

$$\delta_1^{[1]} = \begin{bmatrix} 1.85185 \\ 1.85185 \end{bmatrix} \cdot \begin{bmatrix} -3.1728 \end{bmatrix} \circ \begin{bmatrix} 0.1 e^{0.1 \times 1.8} \\ 0.1 e^{0.1 \times 1.8} \end{bmatrix}$$

$$\delta_1^{[1]} = \begin{bmatrix} -5.8755 \\ -5.8755 \end{bmatrix} \circ \begin{bmatrix} 0.1\, e^{0.1 \times 1.8} \\ 0.1\, e^{0.1 \times 1.8} \end{bmatrix}$$

$$\delta_1^{[1]} = \begin{bmatrix} -0.7034 \\ -0.7034 \end{bmatrix}$$

$$\boxed{\delta_2^{[1]}}$$

$$\delta_2^{[1]} = \left(\left(W^{[2]}\right)^T \delta_2^{[2]} \right) \circ f'\left(z_2^{[1]}\right)$$

$$\delta_2^{[1]} = \begin{bmatrix} 1.85185 \\ 1.85185 \end{bmatrix} \cdot \begin{bmatrix} -2.6229 \end{bmatrix} \circ \begin{bmatrix} 0.1\, e^{0.1 \times 2} \\ 0.1\, e^{0.1 \times 2} \end{bmatrix}$$

$$\delta_2^{[1]} = \begin{bmatrix} -4.8572 \\ -4.8572 \end{bmatrix} \circ \begin{bmatrix} 0.1\, e^{0.2} \\ 0.1\, e^{0.2} \end{bmatrix}$$

$$\delta_2^{[1]} = \begin{bmatrix} -0.5933 \\ -0.5933 \end{bmatrix}$$

$$\boxed{\delta_3^{[1]}}$$

$$\delta_3^{[1]} = \left((W^{[2]})^T \ \delta_3^{[2]} \right) \circ f'\left(z_3^{[1]} \right)$$

$$\delta_3^{[1]} = \begin{bmatrix} 1.85185 \\ 1.85185 \end{bmatrix} \begin{bmatrix} -1.2169 \end{bmatrix} \circ \begin{bmatrix} 0.1 e^{0.1 \times 2.2} \\ 0.1 e^{0.1 \times 2.2} \end{bmatrix}$$

$$\delta_3^{[1]} = \begin{bmatrix} -2.2535 \\ -2.2535 \end{bmatrix} \circ \begin{bmatrix} 0.1 e^{0.22} \\ 0.1 e^{0.22} \end{bmatrix}$$

$$\delta_3^{[1]} = \begin{bmatrix} -0.2808 \\ -0.2808 \end{bmatrix}$$

$$\boxed{W^{[1]}}$$

$$\frac{\partial E}{\partial W^{[1]}} = \sum_{i=1}^{3} \left(\delta_i^{[1]} \cdot \left(x_i^{[0]} \right)^T \right)$$

$$\frac{\partial E}{\partial W^{[1]}} = \begin{bmatrix} -0.7034 \\ -0.7034 \end{bmatrix} \cdot \begin{bmatrix} 0.8 \end{bmatrix} + \begin{bmatrix} -0.5933 \\ -0.5933 \end{bmatrix} \cdot 1 \ +$$

$$+ \begin{bmatrix} -0.2808 \\ -0.2808 \end{bmatrix} \cdot 1.2$$

$$\frac{\partial E}{\partial w^{[1]}} = \begin{bmatrix} -1.49298 \\ -1.49298 \end{bmatrix}$$

$$w^{[1]} = w^{[1]} - \eta \nabla E$$

$$w^{[1]} = \begin{bmatrix} 1 \\ 1 \end{bmatrix} - 0.1 \begin{bmatrix} -1.49298 \\ -1.49298 \end{bmatrix}$$

$$\boxed{w^{[1]} = \begin{bmatrix} 1.149298 \\ 1.149298 \end{bmatrix}}$$

$$\boxed{b^{[1]}}$$

$$\frac{\partial E}{\partial b^{[1]}} = \sum_{i=1}^{3} \delta_i^{[1]}$$

$$= \begin{bmatrix} -0.7034 \\ -0.7034 \end{bmatrix} + \begin{bmatrix} -0.5933 \\ -0.5933 \end{bmatrix} + \begin{bmatrix} -0.2808 \\ -0.2808 \end{bmatrix}$$

$$= \begin{bmatrix} -1.5775 \\ -1.5775 \end{bmatrix}$$

$$b^{[1]} = b^{[1]} - \eta \nabla E$$

$$b^{[1]} = \begin{bmatrix} 1 \\ 1 \end{bmatrix} - 0.1 \begin{bmatrix} -1.5775 \\ -1.5775 \end{bmatrix}$$

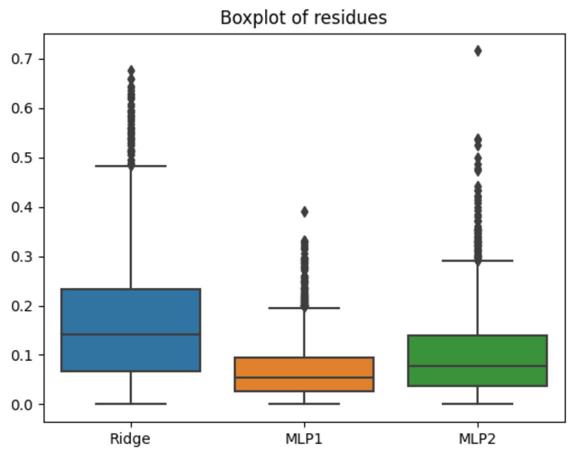
$$\boxed{b^{[1]} = \begin{bmatrix} 1.15775 \\ 1.15775 \end{bmatrix}}$$

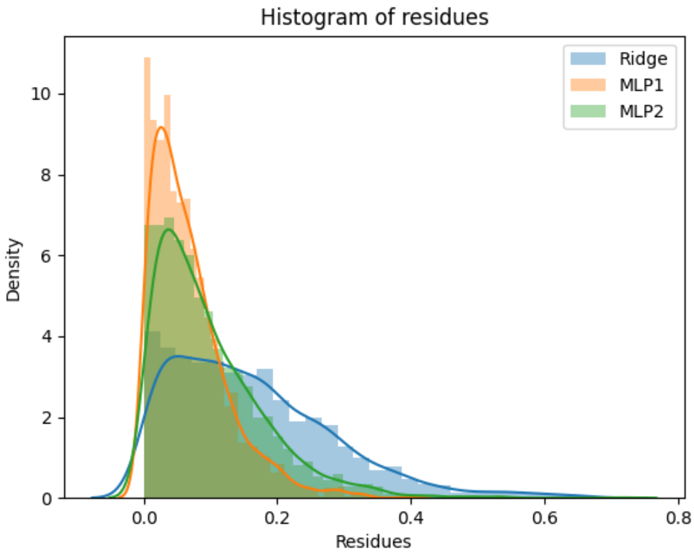
II. Programming and critical analysis

4) Answer 4

```
Mean Absolute Error Ridge Linear Regression: 0.162829976437694
Mean Absolute Error MLP1: 0.0680414073796843
Mean Absolute Error MLP2: 0.0978071820387748
```

5) Answer 5





6) Answer 6

```
MLP1 iterations to converge: 452
MLP2 iterations to converge: 77
```

7) Answer 7

What is motivating the difference between the number of iterations of both MLPs is the early stopping. The fact that the first MLP is parameterized with early stopping helps fighting overfitting. By doing this, we prevent the algorithm from getting too accustomed to the training data and, therefore, it needs more iterations to converge, which makes sense as it is consistently being interrupted by the early stopping. The second MLP converges much faster.

Regarding the observed performance differences between the MLPs, the one with early stopping demonstrates lower average residues and a lower Mean Absolute Error (MAE), which could be due to the fact that, as we fight overfitting, the trained data is better “prepared” when it comes to predicting the outcome of the testing data. On the other hand, the second MLP shows higher average residues and a higher MAE which could be a direct consequence of overfitting.

III. APPENDIX

```python
from sklearn.linear_model import LinearRegression, Ridge, Lasso
import pandas as pd
from scipy.io.arff import loadarff
from sklearn.model_selection import train_test_split
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.neural_network import MLPRegressor
```

```
from sklearn import metrics, datasets


data = loadarff('kin8nm.arff')
df = pd.DataFrame(data[0])

X = df.drop('y', axis=1)
y = df['y']


X_train, X_test, y_train, y_test = train_test_split(X, y, test_size = 0.30, random_state = 0)

#Ridge Regression

ridge = Ridge(alpha = 0.1)
ridge.fit(X_train, y_train)
y_pred_Ridge = ridge.predict(X_test)
print('Mean Absolute Error Ridge Linear Regression:', metrics.mean_absolute_error(y_test,
y_pred_Ridge))


#MLP1

mlp1 = MLPRegressor(hidden_layer_sizes = (10, 10), activation = 'tanh', max_iter = 500, random_state
= 0, early_stopping = True)
mlp1.fit(X_train.values, y_train)
y_pred_mlp1 = mlp1.predict(X_test.values)
print('Mean Absolute Error MLP1:', metrics.mean_absolute_error(y_test, y_pred_mlp1))

#MLP2

mlp2 = MLPRegressor(hidden_layer_sizes = (10, 10), activation = 'tanh', max_iter=500, random_state =
0, early_stopping = False)
mlp2.fit(X_train.values, y_train)
y_pred_mlp2 = mlp2.predict(X_test.values)
print('Mean Absolute Error MLP2:', metrics.mean_absolute_error(y_test, y_pred_mlp2))

#Boxplots

ridgeResidues = abs(y_test - y_pred_Ridge)
MLP1Residues = abs(y_test - y_pred_mlp1)
MLP2Residues = abs(y_test - y_pred_mlp2)

residues = pd.DataFrame({"Ridge": ridgeResidues, "MLP1": MLP1Residues, "MLP2": MLP2Residues})

sns.boxplot(data = residues)
plt.title('Boxplot of residues')
plt.savefig('boxplots.png')
plt.show()
```

```python
#Histograms

sns.distplot(residues['Ridge'], hist = True, label = 'Ridge')
sns.distplot(residues['MLP1'], hist = True, label = 'MLP1')
sns.distplot(residues['MLP2'], hist = True, label = 'MLP2')
plt.title('Histogram of residues')
plt.legend()
plt.xlabel('Residues')
plt.savefig('histograms.png')
plt.show()


print('MLP1 iterations to converge:', mlp1.n_iter_)
print('MLP2 iterations to converge:', mlp2.n_iter_)
if mlp1.n_iter_ < mlp1.max_iter:
    print("MLP1 converged")
else:
    print("MLP1 did not converge")

if mlp2.n_iter_ < mlp2.max_iter:
    print("MLP2 converged")
else:
    print("MLP2 did not converge")
```

END