Aprendizagem 2022

Homework III – Group 126

Part I: Pen and paper

1. To learn the Ridge regression we need to use the following formula:

$$w = \left(X^T X + \lambda \cdot I\right)^{-1} \cdot X^T \cdot z$$

We are given:

$$X = \begin{bmatrix} 1 & 0.8 & 0.64 & 0.512 \\ 1 & 1 & 1 & 1 \\ 1 & 1.2 & 1.44 & 1.728 \\ 1 & 1.4 & 1.96 & 2.744 \\ 1 & 1.6 & 2.56 & 4.096 \end{bmatrix} \qquad z = \begin{bmatrix} 24 \\ 20 \\ 10 \\ 13 \\ 12 \end{bmatrix} \qquad \lambda \cdot I = \begin{bmatrix} 2 & 0 & 0 & 0 \\ 0 & 2 & 0 & 0 \\ 0 & 0 & 2 & 0 \\ 0 & 0 & 0 & 2 \end{bmatrix}$$

With this data we can proceed with the calculations:

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

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

$$X^T \cdot X + \lambda \cdot I = \begin{bmatrix} 5 & 6 & 7.6 & 10.08 \\ 6 & 7.6 & 10.08 & 13.8784 \\ 7.6 & 10.08 & 13.8784 & 19.68 \\ 10.08 & 13.8784 & 19.68 & 28.55488 \end{bmatrix} + \begin{bmatrix} 2 & 0 & 0 & 0 \\ 0 & 2 & 0 & 0 \\ 0 & 0 & 2 & 0 \\ 0 & 0 & 0 & 2 \end{bmatrix}$$

$$= \begin{bmatrix} 7 & 6 & 7.6 & 10.08 \\ 6 & 9.6 & 10.08 & 13.8784 \\ 7.6 & 10.08 & 15.8784 & 19.68 \\ 10.08 & 13.8784 & 19.68 & 30.55488 \end{bmatrix}$$

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

$$X^T \cdot z = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 \\ 0.8 & 1 & 1.2 & 1.4 & 1.6 \\ 0.64 & 1 & 1.44 & 1.96 & 2.56 \\ 0.512 & 1 & 1.728 & 2.744 & 4.096 \end{bmatrix} \cdot \begin{bmatrix} 24 \\ 20 \\ 10 \\ 13 \\ 12 \end{bmatrix} = \begin{bmatrix} 79 \\ 88.6 \\ 105.96 \\ 134.392 \end{bmatrix}$$

$$w = \begin{bmatrix} 0.34168753 & -0.1214259 & -0.07490231 & -0.00932537 \\ -0.1214259 & 0.3892078 & -0.09667718 & -0.07445624 \\ -0.07490231 & -0.09667718 & 0.37257788 & -0.17135047 \\ -0.00932537 & -0.07445624 & -0.17135047 & 0.17998796 \end{bmatrix} \cdot \begin{bmatrix} 79 \\ 88.6 \\ 105.96 \\ 134.392 \end{bmatrix}$$

$$\simeq \begin{bmatrix} 7.05 \\ 4.64 \\ 1.97 \\ -1.30 \end{bmatrix}$$

2. Given $w = \begin{bmatrix} 7.05 & 4.64 & 1.97 & -1.30 \end{bmatrix}^T$ we get the regression:

$$\hat{z}(x) = 7.05 + 4.64x + 1.97x^2 - 1.3x^3$$

Using this function we can calculate the predicted values.

x	z	$\hat{z}$	$(z_i - \hat{z}_i)$	$(z_i - \hat{z}_i)^2$
0.8	24	11.36	12.64	159.7696
1	20	12.36	7.64	58.3696
1.2	10	13.21	-3.21	10.3041
1.4	13	13.84	-0.84	0.7056
1.6	12	14.19	-2.19	4.7961

Then we can calculate the training RMSE for the learnt model:

$$RMSE = \sqrt{\frac{\sum_{i=1}^{n} (z_i - \hat{z}_i)^2}{n}} = \sqrt{\frac{233.945}{5}} \simeq 6.84$$

3. We are given the following data:

x	z
0.8	24
1	20
1.2	10

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

We can now do forward propagation:

- For x_1:

$$z_1^{[1]} = \begin{bmatrix} 1 \\ 1 \end{bmatrix} \cdot \begin{bmatrix} 0.8 \end{bmatrix} + \begin{bmatrix} 1 \\ 1 \end{bmatrix} = \begin{bmatrix} 1.8 \\ 1.8 \end{bmatrix}, \quad x_1^{[1]} = \begin{bmatrix} 1.197 \\ 1.197 \end{bmatrix}$$

$$z_1^{[2]} = \begin{bmatrix} 1 & 1 \end{bmatrix} \cdot \begin{bmatrix} 1.197 \\ 1.197 \end{bmatrix} + \begin{bmatrix} 1 \end{bmatrix} = \begin{bmatrix} 3.394 \end{bmatrix}, \quad x_1^{[2]} = \begin{bmatrix} 1.404 \end{bmatrix}$$

- For x_2:

$$z_2^{[1]} = \begin{bmatrix} 1 \\ 1 \end{bmatrix} \cdot \begin{bmatrix} 1 \end{bmatrix} + \begin{bmatrix} 1 \\ 1 \end{bmatrix} = \begin{bmatrix} 2.0 \\ 2.0 \end{bmatrix}, \quad x_2^{[1]} = \begin{bmatrix} 1.221 \\ 1.221 \end{bmatrix}$$

$$z_2^{[2]} = \begin{bmatrix} 1 & 1 \end{bmatrix} \cdot \begin{bmatrix} 1.221 \\ 1.221 \end{bmatrix} + \begin{bmatrix} 1 \end{bmatrix} = \begin{bmatrix} 3.442 \end{bmatrix}, \quad x_2^{[2]} = \begin{bmatrix} 1.411 \end{bmatrix}$$

- For x_3:

$$z_3^{[1]} = \begin{bmatrix} 1 \\ 1 \end{bmatrix} \cdot \begin{bmatrix} 1.2 \end{bmatrix} + \begin{bmatrix} 1 \\ 1 \end{bmatrix} = \begin{bmatrix} 2.2 \\ 2.2 \end{bmatrix}, \quad x_3^{[1]} = \begin{bmatrix} 1.246 \\ 1.246 \end{bmatrix}$$

$$z_3^{[2]} = \begin{bmatrix} 1 & 1 \end{bmatrix} \cdot \begin{bmatrix} 1.246 \\ 1.246 \end{bmatrix} + \begin{bmatrix} 1 \end{bmatrix} = \begin{bmatrix} 3.492 \end{bmatrix}, \quad x_3^{[2]} = \begin{bmatrix} 1.418 \end{bmatrix}$$

Now we need to derivate all functions in our network:

- $\frac{\partial E}{\partial x^{[2]}} = x^{[2]} - t$
- $\frac{\partial x^{[i]}}{\partial z^{[i]}} = e^{0.1z^{[i]}} \cdot 0.1$
- $\frac{\partial z^{[i]}}{\partial w^{[i]}} = x^{[i-1]}$
- $\frac{\partial z^{[i]}}{\partial x^{[i-1]}} = w^{[i]}$
- $\frac{\partial z^{[i]}}{\partial b^{[i]}} = 1$

We can now calculate the deltas for the layers:

- Last layer: $\delta^{[2]} = \frac{\partial E}{\partial x^{[2]}} \circ \frac{\partial x^{[2]}}{\partial z^{[2]}} = (x^{[2]} - t) \circ (e^{0.1z^{[2]}} \cdot 0.1)$

$$\delta_1^{[2]} = (1.404 - 24) \circ (0.140) = -3.163$$

$$\delta_2^{[2]} = (1.411 - 20) \circ (0.141) = -2.621$$

$$\delta_3^{[2]} = (1.418 - 10) \circ (0.142) = -1.219$$

- Hidden layer: $\delta^{[1]} = \left(\frac{\partial z^{[2]}}{\partial x^{[1]}}\right)^T \cdot \delta^{[2]} \circ \frac{\partial x^{[i]}}{\partial z^{[i]}} = w^{[2]^T} \cdot \delta^{[2]} \circ (e^{0.1z^{[1]}} \cdot 0.1)$

$$\delta_1^{[1]} = \begin{bmatrix} -3.163 \\ -3.163 \end{bmatrix} \circ \begin{bmatrix} 0.120 \\ 0.120 \end{bmatrix} = \begin{bmatrix} -0.380 \\ -0.380 \end{bmatrix}$$

$$\delta_2^{[1]} = \begin{bmatrix} -2.621 \\ -2.621 \end{bmatrix} \circ \begin{bmatrix} 0.122 \\ 0.122 \end{bmatrix} = \begin{bmatrix} -0.320 \\ -0.320 \end{bmatrix}$$

$$\delta_3^{[1]} = \begin{bmatrix} -1.219 \\ -1.219 \end{bmatrix} \circ \begin{bmatrix} 0.125 \\ 0.125 \end{bmatrix} = \begin{bmatrix} -0.152 \\ -0.152 \end{bmatrix}$$

We can now calculate the updated weights and biases:

- Last layer:

$$\begin{aligned}
\frac{\partial E}{\partial w^{[2]}} &= \delta_2^{[2]} \frac{\partial z_2^{[2]}}{\partial w^{[2]}} + \delta_2^{[2]} \frac{\partial z_2^{[2]}}{\partial w^{[2]}} + \delta_3^{[2]} \frac{\partial z_3^{[2]}}{\partial w^{[2]}} \\
&= \delta_1^{[2]} \left(x_1^{[1]}\right)^T + \delta_2^{[2]} \left(x_2^{[1]}\right)^T + \delta_3^{[2]} \left(x_3^{[1]}\right)^T \\
&= \begin{bmatrix}-3.163\end{bmatrix} \cdot \begin{bmatrix}1.197 & 1.197\end{bmatrix} + \begin{bmatrix}-2.621\end{bmatrix} \cdot \begin{bmatrix}1.221 & 1.221\end{bmatrix} + \\
&\quad \begin{bmatrix}-1.219\end{bmatrix} \cdot \begin{bmatrix}1.246 & 1.246\end{bmatrix} \\
&= \begin{bmatrix}-8.505 & -8.505\end{bmatrix}
\end{aligned}$$

$$\begin{aligned}
w'^{[2]} &= w^{[2]} - \eta \cdot \frac{\partial E}{\partial w^{[2]}} \\
&= \begin{bmatrix}1 & 1\end{bmatrix} - 0.1 \cdot \begin{bmatrix}-8.505 & -8.505\end{bmatrix} \\
&= \begin{bmatrix}0.1851 & 0.1851\end{bmatrix}
\end{aligned}$$

$$\begin{aligned}
\frac{\partial E}{\partial b^{[2]}} &= \delta_1^{[2]} \frac{\partial z_1^{[2]}}{\partial b^{[2]}} + \delta_2^{[2]} \frac{\partial z_2^{[2]}}{\partial b^{[2]}} + \delta_3^{[2]} \frac{\partial z_3^{[2]}}{\partial b^{[2]}} \\
&= \delta_1^{[2]} + \delta_2^{[2]} + \delta_3^{[2]} \\
&= \begin{bmatrix}-3.163\end{bmatrix} + \begin{bmatrix}-2.621\end{bmatrix} + \begin{bmatrix}-1.219\end{bmatrix} \\
&= \begin{bmatrix}-7.003\end{bmatrix}
\end{aligned}$$

$$\begin{aligned}
b'^{[2]} &= b^{[2]} - \eta \cdot \frac{\partial E}{\partial b^{[2]}} \\
&= \begin{bmatrix}1\end{bmatrix} - 0.1 \cdot \begin{bmatrix}-7.003\end{bmatrix} \\
&= \begin{bmatrix}1.7\end{bmatrix}
\end{aligned}$$

- Hidden layer:

$$\begin{aligned}\frac{\partial E}{\partial w^{[1]}} &= \delta_1^{[1]}\frac{\partial z_1^{[1]}}{\partial w^{[1]}} + \delta_2^{[1]}\frac{\partial z_2^{[1]}}{\partial w^{[1]}} + \delta_3^{[1]}\frac{\partial z_3^{[1]}}{\partial w^{[1]}} \\ &= \delta_1^{[1]}\left(x_1^{[0]}\right)^T + \delta_2^{[1]}\left(x_2^{[0]}\right)^T + \delta_3^{[1]}\left(x_3^{[0]}\right)^T \\ &= \begin{bmatrix}-0.380\\-0.380\end{bmatrix}\cdot[0.8] + \begin{bmatrix}-0.320\\-0.320\end{bmatrix}\cdot[1] + \begin{bmatrix}-0.152\\-0.152\end{bmatrix}\cdot[1.2] \\ &= \begin{bmatrix}-0.806\\-0.806\end{bmatrix}\end{aligned}$$

$$\begin{aligned}w'^{[1]} &= w^{[1]} - \eta\cdot\frac{\partial E}{\partial w^{[1]}} \\ &= \begin{bmatrix}1\\1\end{bmatrix} - 0.1\cdot\begin{bmatrix}-0.806\\-0.806\end{bmatrix} \\ &= \begin{bmatrix}1.081\\1.081\end{bmatrix}\end{aligned}$$

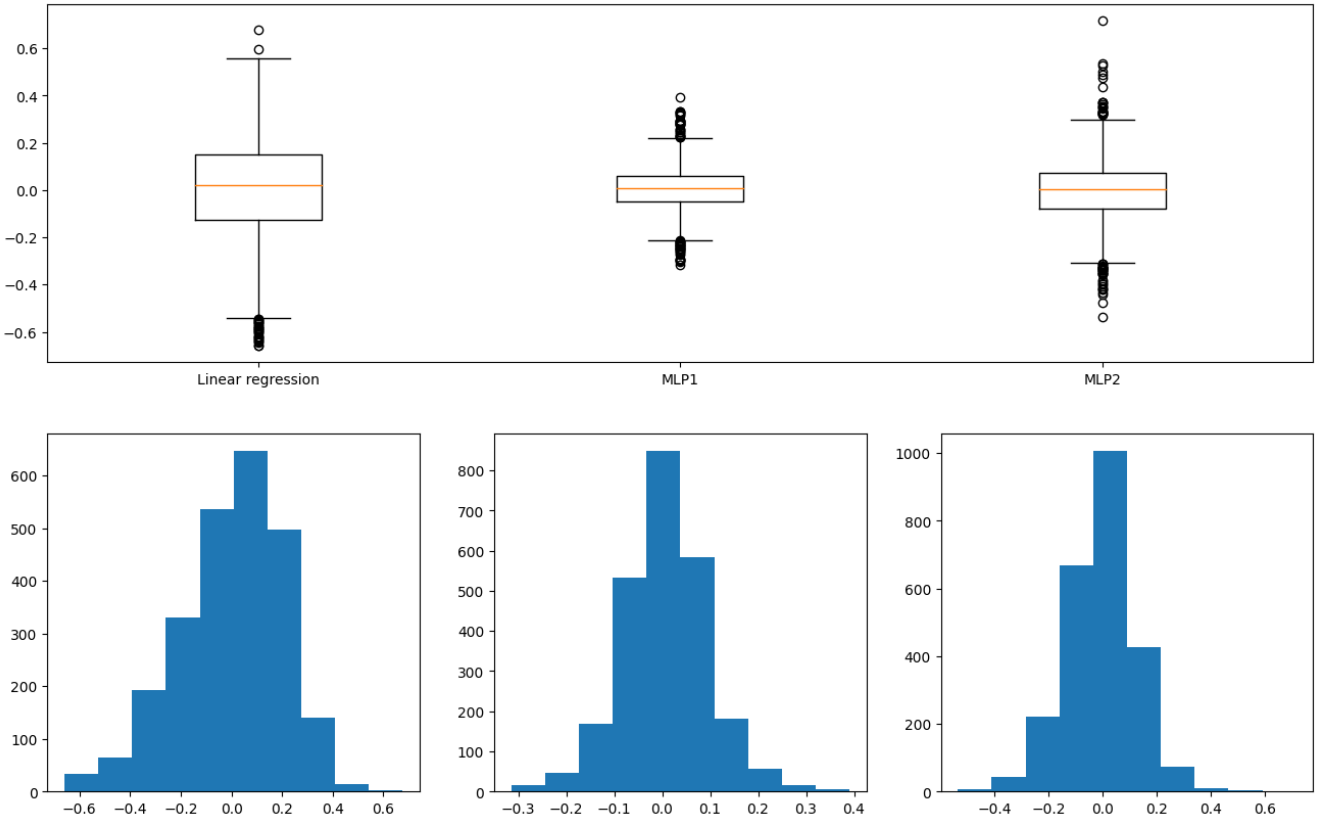
$$\begin{aligned}\frac{\partial E}{\partial b^{[1]}} &= \delta_1^{[1]}\frac{\partial z_1^{[1]}}{\partial b^{[1]}} + \delta_2^{[1]}\frac{\partial z_2^{[1]}}{\partial b^{[1]}} + \delta_3^{[1]}\frac{\partial z_3^{[1]}}{\partial b^{[1]}} \\ &= \delta_1^{[1]} + \delta_2^{[1]} + \delta_3^{[1]} \\ &= \begin{bmatrix}-0.380\\-0.380\end{bmatrix} + \begin{bmatrix}-0.320\\-0.320\end{bmatrix} + \begin{bmatrix}-0.152\\-0.152\end{bmatrix} \\ &= \begin{bmatrix}-0.852\\-0.852\end{bmatrix}\end{aligned}$$

$$\begin{aligned}b'^{[1]} &= b^{[1]} - \eta\cdot\frac{\partial E}{\partial b^{[1]}} \\ &= \begin{bmatrix}1\\1\end{bmatrix} - 0.1\cdot\begin{bmatrix}-0.852\\-0.852\end{bmatrix} \\ &= \begin{bmatrix}1.085\\1.085\end{bmatrix}\end{aligned}$$

Part II: Programming

4. LR: 0.162829976437694
   MLP_1: 0.0680414073796843
   MLP_2: 0.0978071820387748

5. Plots:



6. MLP_1: 452
   MLP_2: 77

7. The only difference between the two MLPs is that MLP_1 considers early stopping, this causes the model to stop training as soon as the validation error reaches a minimum, avoiding overfitting the training data, ence the better performance. Still, given the fact that the loss function doesn't necessarily decrease at each iteration, it makes the monitoring of the convergence on the loss function challenging, thus increasing the number of iterations.

APPENDIX

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

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

X = df.iloc[:, :-1]
y = df.iloc[:, -1]

# Training-testing split
X_train, X_test, y_train, y_test = train_test_split(X, y,
                                                    test_size=0.3, random_state=0)

# Classifiers
classifiers = {'Ridge': Ridge(alpha=0.1),
               '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)}

# MAE
residuals = {}
for name in classifiers:
    classifiers[name].fit(X_train.values, y_train)
    y_pred = classifiers[name].predict(X_test.values)
    print(name, 'MAE:', mean_absolute_error(y_test, y_pred))
    residuals[name] = y_test - y_pred

# Plot
fig, axes = plt.subplot_mosaic("AAA;BCD")
fig.set_size_inches(16, 10)
axes['A'].boxplot([residuals['Ridge'], residuals['MLP1'], residuals['MLP2']],
                  labels=['Linear regression', 'MLP1', 'MLP2'])
axes['B'].hist(residuals['Ridge'])
axes['C'].hist(residuals['MLP1'])
axes['D'].hist(residuals['MLP2'])
plt.show()

# MLP Iterations
print('#Iterations MLP1:', classifiers['MLP1'].n_iter_)
print('#Iterations MLP2:', classifiers['MLP2'].n_iter_)
```