Homework 3

13 de outubro de 2023 10:52

1. Consider the problem of learning a regression model from 4 bivariate observations $\left\{\begin{pmatrix}0.7\\-0.3\end{pmatrix}, \begin{pmatrix}0.4\\0.5\end{pmatrix}, \begin{pmatrix}-0.2\\0.8\end{pmatrix}, \begin{pmatrix}-0.4\\0.3\end{pmatrix}\right\}$ with targets $(0.8, 0.6, 0.3, 0.3)$.

a. [4v] Given the radial basis function, $\phi_j(x) = exp\left(-\frac{\|x-c_j\|^2}{2}\right)$, that transforms the original space onto a new space characterized by the similarity of the original observations to the following data points, $\left\{c_1 = \begin{pmatrix}0\\0\end{pmatrix}, c_2 = \begin{pmatrix}1\\-1\end{pmatrix}, c_3 = \begin{pmatrix}-1\\1\end{pmatrix}\right\}$.

Learn the Ridge regression (l_2 regularization) using the closed solution with $\lambda = 0.1$.

	y_1	y_2	z
x_1	0.7	-0.3	0.8
x_2	0.4	0.5	0.6
x_3	-0.2	0.8	0.3
x_4	-0.4	0.3	0.3

$\xrightarrow{\phi}$

	y_1'	y_2'	y_3'	z
x_1'	$e^{-0.29}$	$e^{-0.29}$	$e^{-2.29}$	0.8
x_2'	$e^{-0.205}$	$e^{-1.305}$	$e^{-1.105}$	0.6
x_3'	$e^{-0.34}$	$e^{-2.34}$	$e^{-0.34}$	0.3
x_4'	$e^{-0.125}$	$e^{-1.825}$	$e^{-0.425}$	0.3

Regressão de Ridge: $w = (X'^T X' + \lambda I)^{-1} X'^T z$, $\lambda = 0.1$

$$w = \left(\begin{bmatrix}1 & 1 & 1 & 1\\ e^{-0.29} & e^{-0.205} & e^{-0.34} & e^{-0.125}\\ e^{-0.29} & e^{-1.305} & e^{-2.34} & e^{-1.825}\\ e^{-2.29} & e^{-1.105} & e^{-0.34} & e^{-0.425}\end{bmatrix}\begin{bmatrix}1 & e^{-0.29} & e^{-0.29} & e^{-2.29}\\ 1 & e^{-0.205} & e^{-1.305} & e^{-1.105}\\ 1 & e^{-0.34} & e^{-2.34} & e^{-0.34}\\ 1 & e^{-0.125} & e^{-1.825} & e^{-0.425}\end{bmatrix} + \begin{bmatrix}0.1 & 0 & 0 & 0\\ 0 & 0.1 & 0 & 0\\ 0 & 0 & 0.1 & 0\\ 0 & 0 & 0 & 0.1\end{bmatrix}\right)^{-1} X'^T z$$

$$= \begin{bmatrix}4.100000 & 3.157178 & 1.276981 & 1.798017\\ 3.157178 & 2.608966 & 0.991646 & 1.429161\\ 1.276981 & 0.991646 & 0.768703 & 0.339552\\ 1.798017 & 1.429161 & 0.339552 & 1.153987\end{bmatrix}\begin{bmatrix}1 & 1 & 1 & 1\\ e^{-0.29} & e^{-0.205} & e^{-0.34} & e^{-0.125}\\ e^{-0.29} & e^{-1.305} & e^{-2.34} & e^{-1.825}\\ e^{-2.29} & e^{-1.65} & e^{-0.34} & e^{-0.425}\end{bmatrix}\begin{bmatrix}0.8\\0.6\\0.3\\0.3\end{bmatrix}$$

$$= \begin{bmatrix}0.33914\\0.19945\\0.40096\\-0.29600\end{bmatrix}\begin{matrix}w_0\\w_1\\w_2\\w_3\end{matrix}$$

$$y(x') = 0.33914 + 0.19945\,x_1' + 0.40096\,x_2' - 0.296\,x_3'$$

b. [2v] Compute the training RMSE for the learnt regression.

$$\hat{z} = X'w = \begin{bmatrix}1 & e^{-0.29} & e^{-0.29} & e^{-2.29}\\ 1 & e^{-0.205} & e^{-1.305} & e^{-1.105}\\ 1 & e^{-0.34} & e^{-2.34} & e^{-0.34}\\ 1 & e^{-0.125} & e^{-1.825} & e^{-0.425}\end{bmatrix}\begin{bmatrix}0.33914\\0.19945\\0.40096\\-0.29600\end{bmatrix} = \begin{bmatrix}0.75844\\0.51232\\0.30905\\0.38629\end{bmatrix}$$

RMSE (root.min.squared.error) $= \sqrt{\frac{1}{n}\sum_{i=1}^{n}(z_i - \hat{z}_i)^2}$

$$= \sqrt{\frac{1}{4}\left((0.8-0.75844)^2 + (0.6-0.51232)^2 + (0.3-0.30905)^2 + (0.3-0.38629)^2\right)}$$

$$\simeq 0.06508$$

2. [6v] Consider a MLP classifier of three outcomes – A, B and C – characterized by the weights,

$$W^{[1]} = \begin{pmatrix}1 & 1 & 1 & 1\\ 1 & 1 & 2 & 1\\ 1 & 1 & 1 & 1\end{pmatrix}, b^{[1]} = \begin{pmatrix}1\\1\\1\end{pmatrix}, W^{[2]} = \begin{pmatrix}1 & 4 & 1\\ 1 & 1 & 1\end{pmatrix}, b^{[2]} = \begin{pmatrix}1\\1\end{pmatrix}, W^{[3]} = \begin{pmatrix}1 & 1\\ 3 & 1\\ 1 & 1\end{pmatrix}, b^{[3]} = \begin{pmatrix}1\\1\\1\end{pmatrix}$$

the activation $f(x) = \frac{e^{0.5x-2} - e^{-0.5x+2}}{e^{0.5x-2} + e^{-0.5x+2}} = tanh(0.5x - 2)$ for every unit, and squared error loss $\frac{1}{2}\|\mathbf{z} - \hat{\mathbf{z}}\|_2^2$. Perform one batch gradient descent update (with learning rate $\eta = 0.1$) for training observations $\mathbf{x}_1 = \begin{pmatrix}1\\1\\1\\1\end{pmatrix}$ and $\mathbf{x}_2 = \begin{pmatrix}1\\0\\0\\-1\end{pmatrix}$ with targets B and A, respectively.

$A = \begin{pmatrix}1\\-1\\-1\end{pmatrix}$ $B = \begin{pmatrix}-1\\1\\-1\end{pmatrix}$ $E(w) = \frac{1}{2}\sum_{i=1}^{n}(t_i - x_i)^2$, $x = \phi(z)$

C.A:

$$\frac{\partial E}{\partial w^{[3]}} = \frac{\partial E}{\partial x^{[3]}}\frac{\partial x^{[3]}}{\partial z^{[3]}}\frac{\partial z^{[3]}}{\partial w^{[3]}} = \delta^{[3]}\frac{\partial z^{[3]}}{\partial w^{[3]}} ; \frac{\partial E}{\partial b^{[3]}} = \delta^{[3]}$$

$$\delta^{[p]} = \left(\frac{\partial z^{[p+1]}}{\partial x^{[p]}}\right)^T \cdot \delta^{[p+1]} \circ \frac{\partial x^{[p]}}{\partial z^{[p]}}$$

$$\frac{\partial E}{\partial x^3} = -\sum_{i=1}^{n}(t_i - x_i)$$

$$\frac{\partial x^{[p]}}{\partial z^{[p]}} = \frac{\partial f(z^{[p]})}{\partial z^{[p]}} = 0.5\cdot sech^2(0.5\,z^{[p]} - 2)$$

$$\frac{\partial z^{[p]}}{\partial w^{[p]}} = x^{[p-1]}$$

$$\frac{\partial z^{[p]}}{\partial b^{[p]}} = 1$$

$$\frac{\partial z^{[p]}}{\partial x^{[p-1]}} = w^{[p]}$$

forward propagation

$$z^{[1][1]} = w^{[1]}x^{[0][1]} + b^{[1]} = \begin{bmatrix}1 & 1 & 1 & 1\\ 1 & 1 & 2 & 1\\ 1 & 1 & 1 & 1\end{bmatrix}\begin{bmatrix}1\\1\\1\\1\end{bmatrix} + \begin{bmatrix}1\\1\\1\end{bmatrix} = \begin{bmatrix}5\\6\\5\end{bmatrix}$$

$$x^{[1][1]} = tanh\left(0.5\begin{bmatrix}5\\6\\5\end{bmatrix} - 2\right) = \begin{bmatrix}0.462117\\0.761594\\0.462117\end{bmatrix}$$

$$z^{[2][1]} = w^{[2]}x^{[1][1]} + b^{[2]} = \begin{bmatrix}1 & 4 & 1\\ 1 & 1 & 1\end{bmatrix}\begin{bmatrix}0.462117\\0.761594\\0.462117\end{bmatrix} + \begin{bmatrix}1\\1\end{bmatrix} = \begin{bmatrix}4.970611\\2.685828\end{bmatrix}$$

$$x^{[2][1]} = tanh\left(0.5\begin{bmatrix}4.970611\\2.685828\end{bmatrix} - 2\right) = \begin{bmatrix}0.450483\\-0.576421\end{bmatrix}$$

$$z^{[3][1]} = w^{[3]}x^{[2][1]} + b^{[3]} = \begin{bmatrix}1 & 1\\ 3 & 1\\ 1 & 1\end{bmatrix}\begin{bmatrix}0.450483\\-0.576421\end{bmatrix} + \begin{bmatrix}1\\1\\1\end{bmatrix} = \begin{bmatrix}0.874062\\1.775027\\0.874062\end{bmatrix}$$

$$x^{[3][1]} = tanh\left(0.5\begin{bmatrix}0.874062\\1.775027\\0.874062\end{bmatrix} - 2\right) = \begin{bmatrix}-0.915900\\-0.804940\\-0.915900\end{bmatrix}$$

$$z^{[1][2]} = w^{[1]}x^{[0][2]} + b^{[1]} = \begin{bmatrix}1 & 1 & 1 & 1\\ 1 & 1 & 2 & 1\\ 1 & 1 & 1 & 1\end{bmatrix}\begin{bmatrix}1\\0\\0\\-1\end{bmatrix} + \begin{bmatrix}1\\1\\1\end{bmatrix} = \begin{bmatrix}1\\1\\1\end{bmatrix}$$

$$x^{[1][2]} = tanh\left(0.5\begin{bmatrix}1\\1\\1\end{bmatrix} - 2\right) = \begin{bmatrix}-0.905148\\-0.905148\\-0.905148\end{bmatrix}$$

$$z^{[2][2]} = w^{[2]}x^{[1][2]} + b^{[2]} = \begin{bmatrix}1 & 4 & 1\\ 1 & 1 & 1\end{bmatrix}\begin{bmatrix}-0.905148\\-0.905148\\-0.905148\end{bmatrix} + \begin{bmatrix}1\\1\end{bmatrix} = \begin{bmatrix}-4.430890\\-1.715445\end{bmatrix}$$

$$x^{[2][2]} = tanh\left(0.5\begin{bmatrix}-4.430890\\-1.715445\end{bmatrix} - 2\right) = \begin{bmatrix}-0.999564\\-0.993432\end{bmatrix}$$

$$z^{[3][2]} = w^{[3]}x^{[2][2]} + b^{[3]} = \begin{bmatrix}1 & 1\\ 3 & 1\\ 1 & 1\end{bmatrix}\begin{bmatrix}-0.999564\\-0.993432\end{bmatrix} + \begin{bmatrix}1\\1\\1\end{bmatrix} = \begin{bmatrix}-0.992996\\-2.992124\\-0.992996\end{bmatrix}$$

$$x^{[3][2]} = tanh\left(0.5\begin{bmatrix}-0.992996\\-2.992124\\-0.992996\end{bmatrix} - 2\right) = \begin{bmatrix}-0.986521\\-0.998164\\-0.986521\end{bmatrix}$$

backwards propagation

$$\delta^{[3][1]} = \frac{\partial E}{\partial x^{[3][1]}}\frac{\partial x^{[3][1]}}{\partial z^{[3][1]}} = (x^{[3][1]} - t^{[1]}) \circ 0.5\cdot sech^2(0.5\,z^{[3][1]} - 2)$$

$$= \left(\begin{bmatrix}-0.915900\\-0.804940\\-0.915900\end{bmatrix} - \begin{bmatrix}-1\\1\\-1\end{bmatrix}\right) \circ \begin{bmatrix}0.080563\\0.176036\\0.080563\end{bmatrix} = \begin{bmatrix}0.006775\\-0.317735\\0.006775\end{bmatrix}$$

$$\delta^{[2][1]} = \left(\frac{\partial z^{[3][1]}}{\partial x^{[2][1]}}\right)^T \circ \delta^{[3][1]} \cdot \frac{\partial x^{[2][1]}}{\partial z^{[2][1]}} = w^{[3]T} \cdot \delta^{[3][1]} \circ \frac{\partial f(z^{[2][1]})}{\partial z^{[2][1]}}$$

$$= \begin{bmatrix}1 & 3 & 1\\ 1 & 1 & 1\end{bmatrix} \cdot \begin{bmatrix}0.006775\\-0.317735\\0.006775\end{bmatrix} \circ 0.5\cdot sech^2(0.5\,z^{[2][1]} - 2)$$

$$= \begin{bmatrix}-0.374482\\-0.101558\end{bmatrix}$$

$$\delta^{[1][1]} = \left(\frac{\partial z^{[2][1]}}{\partial x^{[1][1]}}\right)^T \circ \delta^{[2][1]} \cdot \frac{\partial x^{[1][1]}}{\partial z^{[1][1]}} = \begin{bmatrix}1 & 1\\ 4 & 1\\ 1 & 1\end{bmatrix}\begin{bmatrix}-0.374482\\-0.101558\end{bmatrix} \circ 0.5\cdot sech^2(0.5\,z^{[1][1]} - 2)$$

$$= \begin{bmatrix}-0.187190\\-0.335872\\-0.187190\end{bmatrix}$$

$$\delta^{[3][2]} = \frac{\partial E}{\partial x^{[3][2]}}\frac{\partial x^{[3][2]}}{\partial z^{[3][2]}} = (x^{[3][2]} - t^{[2]}) \circ 0.5\cdot sech^2(0.5\,z^{[3][2]} - 2)$$

$$= \left(\begin{bmatrix}-0.986521\\-0.998164\\-0.986521\end{bmatrix} - \begin{bmatrix}1\\-1\\-1\end{bmatrix}\right) \circ 0.5\cdot sech^2(0.5\,z^{[3][2]} - 2) = \begin{bmatrix}-2.659614\times10^{-2}\\3.369628\times10^{-6}\\1.804629\times10^{-4}\end{bmatrix}$$

$$\delta^{[2][2]} = \left(\frac{\partial z^{[3][2]}}{\partial x^{[2][2]}}\right)^T \circ \delta^{[3][2]} \cdot \frac{\partial x^{[2][2]}}{\partial z^{[2][2]}} = w^{[3]T} \cdot \delta^{[3][2]} \circ \frac{\partial f(z^{[2][2]})}{\partial z^{[2][2]}}$$

$$= \begin{bmatrix}1 & 3 & 1\\ 1 & 1 & 1\end{bmatrix} \cdot \begin{bmatrix}-2.659614\times10^{-2}\\3.369628\times10^{-6}\\1.804629\times10^{-4}\end{bmatrix} \circ 0.5\cdot sech^2(0.5\,z^{[2][2]} - 2)$$

$$= \begin{bmatrix}-1.150925\times10^{-6}\\-1.728993\times10^{-6}\end{bmatrix}$$

$$\delta^{[1][2]} = \left(\frac{\partial z^{[2][2]}}{\partial x^{[1][2]}}\right)^T \circ \delta^{[2][2]} \cdot \frac{\partial x^{[1][2]}}{\partial z^{[1][2]}} = \begin{bmatrix}1 & 1\\ 4 & 1\\ 1 & 1\end{bmatrix}\begin{bmatrix}-2.071069\times10^{-6}\\-2.071597\times10^{-6}\end{bmatrix} \circ 0.5\cdot sech^2(0.5\,z^{[1][2]} - 2)$$

$$= \begin{bmatrix}-1.666192\times10^{-5}\\-1.978162\times10^{-5}\\-1.666192\times10^{-5}\end{bmatrix}$$

gradient update

$$\frac{\partial E}{\partial w^{[1]}} = \delta^{[1][1]}\frac{\partial z^{[1][1]}}{\partial w^{[1]}} + \delta^{[1][2]}\frac{\partial z^{[1][2]}}{\partial w^{[1]}} = \begin{bmatrix}-0.187190\\-0.335872\\-0.187190\end{bmatrix}\begin{bmatrix}1 & 1 & 1 & 1\end{bmatrix} + \begin{bmatrix}-1.666192\times10^{-5}\\-1.978162\times10^{-5}\\-1.666192\times10^{-5}\end{bmatrix}\begin{bmatrix}1 & 0 & 0 & -1\end{bmatrix}$$

$$= \begin{bmatrix}-0.187207 & -0.187190 & -0.187190 & -0.187174\\ -0.335891 & -0.335872 & -0.335872 & -0.335852\\ -0.187207 & -0.187190 & -0.187190 & -0.187174\end{bmatrix}$$

$$\frac{\partial E}{\partial w^{[2]}} = \delta^{[2][1]}\frac{\partial z^{[2][1]}}{\partial w^{[2]}} + \delta^{[2][2]}\frac{\partial z^{[2][2]}}{\partial w^{[2]}} = \begin{bmatrix}-0.374482\\-0.101558\end{bmatrix}\begin{bmatrix}0.462117\\0.761594\\0.462117\end{bmatrix}^T + \begin{bmatrix}-1.150925\times10^{-6}\\-1.728993\times10^{-6}\end{bmatrix}\begin{bmatrix}-0.905148\\-0.905148\\-0.905148\end{bmatrix}^T$$

$$= \begin{bmatrix}-0.17304 & -0.285193 & -0.173044\\ -0.046775 & -0.077189 & -0.046775\end{bmatrix}$$

$$\frac{\partial E}{\partial w^{[3]}} = \delta^{[3][1]}\frac{\partial z^{[3][1]}}{\partial w^{[3]}} + \delta^{[3][2]}\frac{\partial z^{[3][2]}}{\partial w^{[3]}} = \begin{bmatrix}0.006775\\-0.317735\\0.006775\end{bmatrix}\begin{bmatrix}0.450483\\-0.576421\end{bmatrix}^T + \begin{bmatrix}-2.659614\times10^{-2}\\3.369628\times10^{-6}\\1.804629\times10^{-4}\end{bmatrix}\begin{bmatrix}-0.999564\\-0.993432\end{bmatrix}^T$$

$$= \begin{bmatrix}0.029637 & 0.022516\\ -0.143138 & 0.183146\\ -0.002872 & -0.004085\end{bmatrix}$$

$$w^{[1]} = w^{[1]} - 0.1\frac{\partial E}{\partial w^{[1]}} = \begin{bmatrix}1 & 1 & 1 & 1\\ 1 & 1 & 2 & 1\\ 1 & 1 & 1 & 1\end{bmatrix} - 0.1\begin{bmatrix}-0.187207 & -0.187190 & -0.187190 & -0.187174\\ -0.335891 & -0.335872 & -0.335872 & -0.335852\\ -0.187207 & -0.187190 & -0.187190 & -0.187174\end{bmatrix}$$

$$= \begin{bmatrix}1.018721 & 1.018719 & 1.018719 & 1.018717\\ 1.033589 & 1.033587 & 2.033587 & 1.033585\\ 1.018721 & 1.018719 & 1.018719 & 1.018717\end{bmatrix}$$

$$w^{[2]} = w^{[2]} - 0.1\frac{\partial E}{\partial w^{[2]}} = \begin{bmatrix}1 & 4 & 1\\ 1 & 1 & 1\end{bmatrix} - 0.1\begin{bmatrix}-0.17304 & -0.285193 & -0.173044\\ -0.046775 & -0.077189 & -0.046775\end{bmatrix}$$

$$= \begin{bmatrix}1.017304 & 4.028519 & 1.017304\\ 1.004678 & 1.007719 & 1.004678\end{bmatrix}$$

$$w^{[3]} = w^{[3]} - 0.1\frac{\partial E}{\partial w^{[3]}} = \begin{bmatrix}1 & 1\\ 3 & 1\\ 1 & 1\end{bmatrix} - 0.1\begin{bmatrix}0.029637 & 0.022516\\ -0.143138 & 0.183146\\ -0.002872 & -0.004085\end{bmatrix} = \begin{bmatrix}0.997036 & 0.997748\\ 3.014314 & 0.981685\\ 0.999713 & 1.000408\end{bmatrix}$$

$$b^{[1]} = b^{[1]} - 0.1\frac{\partial E}{\partial b^{[1]}} = \begin{bmatrix}1\\1\\1\end{bmatrix} - 0.1(\delta^{[1][1]} + \delta^{[1][2]}) = \begin{bmatrix}1\\1\\1\end{bmatrix} - 0.1\left(\begin{bmatrix}-0.187190\\-0.335872\\-0.187190\end{bmatrix} + \begin{bmatrix}-1.666192\times10^{-5}\\-1.978162\times10^{-5}\\-1.666192\times10^{-5}\end{bmatrix}\right)$$

$$= \begin{bmatrix}1.018721\\1.033589\\1.018721\end{bmatrix}$$

$$b^{[2]} = b^{[2]} - 0.1\frac{\partial E}{\partial b^{[2]}} = \begin{bmatrix}1\\1\end{bmatrix} - 0.1(\delta^{[2][1]} + \delta^{[2][2]}) = \begin{bmatrix}1\\1\end{bmatrix} - 0.1\left(\begin{bmatrix}-0.374482\\-0.101558\end{bmatrix} + \begin{bmatrix}-1.150925\times10^{-6}\\-1.728993\times10^{-6}\end{bmatrix}\right)$$

$$= \begin{bmatrix}1.037449\\1.010173\end{bmatrix}$$

$$b^{[3]} = b^{[3]} - 0.1\frac{\partial E}{\partial b^{[3]}} = \begin{bmatrix}1\\1\\1\end{bmatrix} - 0.1(\delta^{[3][1]} + \delta^{[3][2]}) = \begin{bmatrix}1\\1\\1\end{bmatrix} - 0.1\left(\begin{bmatrix}0.006775\\-0.317735\\0.006775\end{bmatrix} + \begin{bmatrix}-2.659614\times10^{-2}\\3.369628\times10^{-6}\\1.804629\times10^{-4}\end{bmatrix}\right)$$

$$= \begin{bmatrix}1.001982\\1.031773\\0.999304\end{bmatrix}$$

I. Pen-and-paper

I.

Cálculo da Regressão de Ridge

In [ ]:
```
import numpy as np
import math

e = math.e

X = np.array([[1, e ** -0.29, e ** -0.29, e ** -2.29],
              [1, e ** -0.205, e ** -1.305, e ** -1.105],
              [1, e ** -0.34, e ** -2.34, e ** -0.34],
              [1, e ** -0.125, e ** -1.825, e ** -0.425]])

Z = np.array([0.8, 0.6, 0.3, 0.3]).T

W = np.linalg.pinv(X.T @ X + 0.1 * np.identity(4)) @ X.T @ Z

Z_pred = X @ W
```

II.

Cálculo do Delta 3

In [ ]:
```
X = np.array([-0.98652085, -0.9981635,  -0.98652085])
T = np.array([1, -1, -1])
Z = np.array([-0.99299631, -2.99212439, -0.99299631])

delta3 = (X - T) * (0.5 * (1 - np.tanh((0.5 * Z) - 2) ** 2))
```

Cálculo dos Deltas 1 e 2

In [ ]:
```
W_T = np.array([[1, 1],
                [4, 1],
                [1, 1]])
delta_prev = np.array([[-1.15092543e-05],
                       [-1.72899256e-04]])
Z = np.array([[1],
              [1],
              [1]])

delta = np.multiply(np.dot(W_T, delta_prev), 0.5 * (1 - np.tanh(0.5 * Z - 2)
** 2))
```

Cálculo dos dE / dW

In [ ]:
```
delta_X1 = np.array([[0.006775],
                     [-0.317735],
                     [0.006775]])
X1 = np.array([[0.450483, -0.576421]])
delta_X2 = np.array([[-2.659614e-02],
```

```
                    [3.369628e-06],
                    [1.804629e-04]])
X2 = np.array([[-0.999564, -0.993432]])

dEdW = delta_X1 @ X1 + delta_X2 @ X2
```

Cálculo dos pesos atualizados

In [ ]:
```
W = np.array([[1, 1, 1, 1],
              [1, 1, 2, 1],
              [1, 1, 1, 1]])
dEdW = np.array([[-0.18720683, -0.18719017, -0.18719017, -0.18717351],
                 [-0.33589135, -0.33587157, -0.33587157, -0.33585179],
                 [-0.18720683, -0.18719017, -0.18719017, -0.18717351]])

W_new = W - 0.1 * dEdW
```

Cálculo dos biases atualizados

In [ ]:
```
B = np.array([[1],
              [1],
              [1]])
delta_X1 = np.array([[-0.187190],
                     [-0.335872],
                     [-0.187190]])
delta_X2 = np.array([[-1.6661921e-05],
                     [-1.9781619e-05],
                     [-1.6661921e-05]])

B_new = B - 0.1 * (delta_X1 + delta_X2)
```

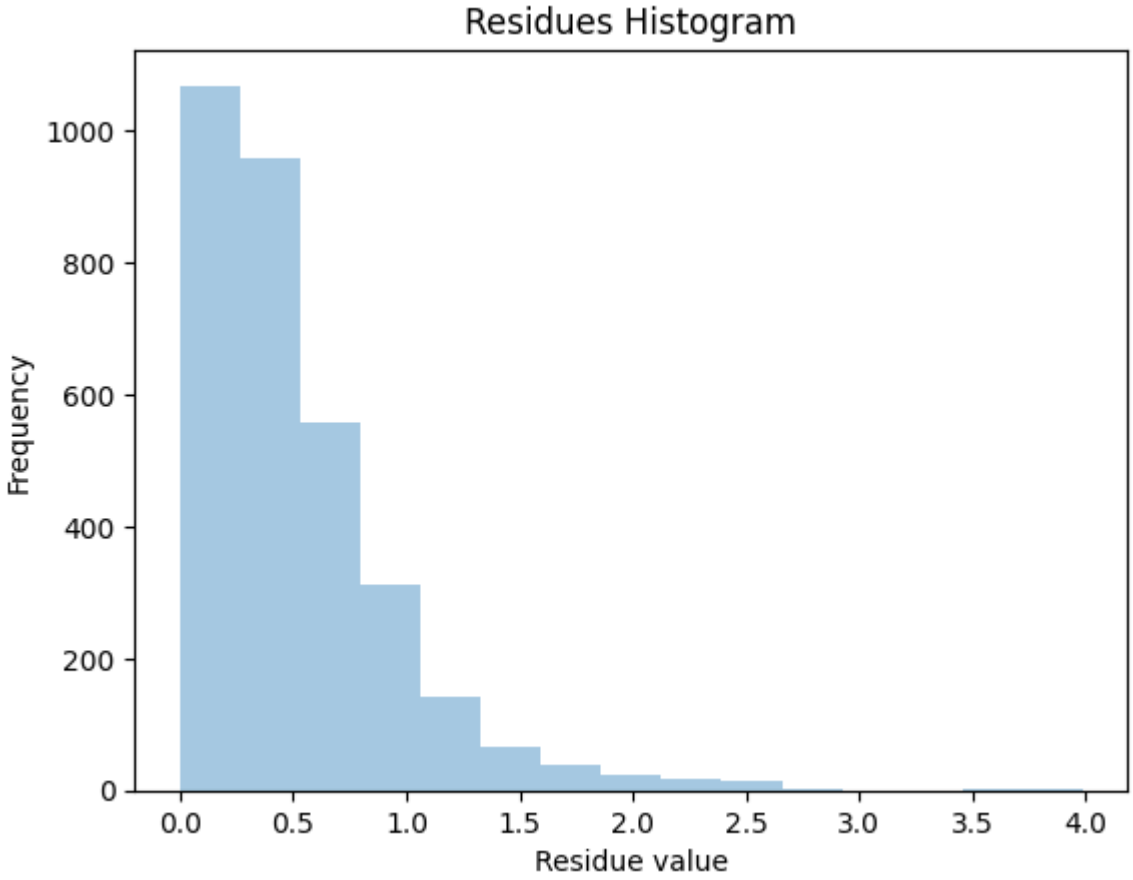
II. Programming and critical analysis

Consider the winequality-red.csv dataset (available at the webpage) where the goal is to estimate the quality (sensory appreciation) of a wine based on physicochemical inputs. Using a 80-20 training-test split with a fixed seed (random_state=0), you are asked to learn MLP regressors to answer the following questions. Given their stochastic behavior, average the performance of each MLP from 10 runs (for reproducibility consider seeding the MLPs with random_state $\in\{1..10\}$).

1. **[3.5v] Learn a MLP regressor with 2 hidden layers of size 10, rectifier linear unit activation on all nodes, and early stopping with 20% of training data set aside for validation. All remaining parameters (e.g., loss, batch size, regularization term, solver) should be set as default. Plot the distribution of the residues (in absolute value) using a histogram.**

In [ ]:

```
import pandas as pd
from sklearn.neural_network import MLPRegressor
from sklearn.model_selection import train_test_split
from sklearn.metrics import mean_absolute_error
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns

df = pd.read_csv('winequality-red.csv', delimiter=';')
X = df.drop('quality', axis=1)
y = df['quality']

X_train, X_test, y_train, y_test = train_test_split(X, y,
train_size=0.8, random_state=0)
residues, errors_original, errors_rounded = [], [], []

for i in range(1, 11):
    mlp = MLPRegressor(hidden_layer_sizes=(10, 10), activation='relu',
early_stopping=True, random_state=i, validation_fraction=0.2)
    mlp.fit(X_train, y_train)
    y_pred = mlp.predict(X_test)
    residues.append(np.abs(np.array(y_pred) - np.array(y_test)))

    # for next question
    rounded_predictions = np.round(y_pred)
    rounded_predictions = np.clip(rounded_predictions, 1, 10)
    errors_original.append(mean_absolute_error(y_test, y_pred))
    errors_rounded.append(mean_absolute_error(y_test, rounded_predictions))

sns.distplot(residues, kde=False, bins=15)
plt.xlabel('Residue value')
plt.ylabel('Frequency')
plt.title('Residues Histogram')
plt.show()
```



2. **[1.5v] Since we are in the presence of a integer regression task, a recommended trick is to round and bound estimates. Assess the impact of these operations on the MAE of the MLP learnt in previous question.**

In [ ]:
```
mae_original = np.mean(errors_original)
mae_rounded = np.mean(errors_rounded)

print(f"Original MAE: {round(mae_original, 5)}")
print(f"MAE after rounding and bounding: {round(mae_rounded, 5)}")
```

```
Original MAE: 0.50972
MAE after rounding and bounding: 0.43875
```

Podemos observar que o *MAE* após *rounding and bounding* é inferior ao original, o que nos indica que estas *predictions* estão mais próximas dos valores reais. Apesar de apresentar esta vantagem e tornar as *predictions* mais interpretáveis (por terem uma natureza mais próxima da *integer regression*), o *rounding and bounding* pode levar a uma perda significativa de precisão.

3. **[1.5v] Similarly assess the impact on RMSE from replacing early stopping by a well-defined number of iterations in {20,50,100,200} (where one iteration corresponds to a batch).**

In [ ]:
```
from sklearn.metrics import mean_squared_error

for iter_num in (20, 50, 100, 200):
    errors = []
```

```
    for i in range(1, 11):
        mlp = MLPRegressor(hidden_layer_sizes=(10, 10), activation='relu',
max_iter=iter_num, random_state=i)
        mlp.fit(X_train, y_train)
        y_pred = mlp.predict(X_test)

        errors.append(mean_squared_error(y_test, y_pred, squared=False))

    rmse = np.mean(errors)
    print(f"RMSE for MLP with max {iter_num} iterations: {round(rmse, 5)}")
```

```
RMSE for MLP with max 20 iterations: 1.40398
RMSE for MLP with max 50 iterations: 0.79961
RMSE for MLP with max 100 iterations: 0.69404
RMSE for MLP with max 200 iterations: 0.65545
```

4. **[1.5v] Critically comment the results obtained in previous question, hypothesizing at least one reason why early stopping favors and/or worsens performance.**

R: Se não utilizarmos *early stopping*, o fim do treino é baseado apenas nos dados de treino, o que pode levar tanto a *underfitting* (caso pare num máximo local) como a *overfitting* (caso se ajuste demasiado aos dados de treino). O *early stopping* consiste em terminar o treino quando o *validation score* não é melhorado durante um certo número consecutivo de épocas, o que previne ambos os problemas mencionados anteriormente.

Analisando os resultados obtidos na pergunta anterior, podemos observar que com o aumento do número de iterações o erro diminui. Assim, não temos quaisquer evidências de *overfitting* (neste caso, o erro aumentaria com o número de iterações), pelo que não podemos concluir desta forma o *threshold* do *early stopping*. Posteriormente, calculámos o *RMSE* do exercício 2 (mesmo treino mas com *early stopping*), obtendo o valor 0.67065, o que nos indica que o treino original terminaria entre as 100 e as 200 iterações. O facto de o erro continuar a diminuir para um número de iterações além do *threshold* do *early stopping* ([100,200]) é algo inesperado, levando-nos a crer que os dados de validação utilizados podem não ser os mais adequados.