Homework III – Group 029

(ist1103000, ist1103179)

I. Pen-and-paper

1

(a)

ridge regression $\Rightarrow$ $\underbrace{E(w) = 1/2 \sum_{n=1}^{N} (t_n - w^T \cdot x_n)^2 + \frac{\lambda}{2} \|w\|_2^2}_{\text{quadratic regularizer}}$

radial basis function : $\phi_j(x) = \exp\left(-\frac{\|x - c_j\|^2}{2}\right)$

closed-form solution : $\nabla E(w) = 0 \Leftrightarrow w = (X^T \cdot X + \lambda \cdot I)^{-1} \cdot X^T \cdot t$

First, we need to build the $n \times (d+1)$ design matrix to account for the bias parameter, where n is the number of examples and d is the original number of input features. Since $\phi_j(x) = \exp\left(-\frac{\|x - c_j\|^2}{2}\right)$:

$$X = \begin{bmatrix} 1 & \phi_1(x_1) & \phi_2(x_1) & \phi_3(x_1) \\ 1 & \phi_1(x_2) & \phi_2(x_2) & \phi_3(x_2) \\ 1 & \phi_1(x_3) & \phi_2(x_3) & \phi_3(x_3) \\ 1 & \phi_1(x_4) & \phi_2(x_4) & \phi_3(x_4) \end{bmatrix}$$

$$\phi_1(x_1) = \exp\left(-\frac{\sqrt{(0.7-0)^2 + (-0.3-0)^2}^2}{2}\right) \simeq 0.74826 ; \phi_1(x_2) = 0.81465$$

$\phi_1(x_3) \simeq 0.71177$; $\phi_1(x_4) \simeq 0.88250$ $\rightarrow c_j = c_1 = \begin{pmatrix} 0 \\ 0 \end{pmatrix}$

$$\phi_2(x_1) = \exp\left(-\frac{\sqrt{(0.7-1)^2 + (-0.3+1)^2}^2}{2}\right) \simeq 0.74826 ; \phi_2(x_2) \simeq 0.27117$$

$\phi_2(x_3) \simeq 0.09633$; $\phi_2(x_4) \simeq 0.16122$ $\rightarrow c_j = c_2 = \begin{pmatrix} 1 \\ -1 \end{pmatrix}$

$$\phi_3(x_1) = \exp\left(-\frac{\sqrt{(0.7+1)^2 + (-0.3-1)^2}^2}{2}\right) \simeq 0.10127 ; \phi_3(x_2) \simeq 0.33121$$

$\phi_3(x_3) \simeq 0.71177$; $\phi_3(x_4) \simeq 0.65377$ $\rightarrow c_j = c_3 = \begin{pmatrix} -1 \\ 1 \end{pmatrix}$

And thus:

$$X = \begin{bmatrix} 1 & 0.74826 & 0.74826 & 0.10127 \\ 1 & 0.81465 & 0.27117 & 0.33121 \\ 1 & 0.71177 & 0.09633 & 0.71177 \\ 1 & 0.88250 & 0.16122 & 0.65377 \end{bmatrix}$$

$$X^T = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 0.74826 & 0.81465 & 0.71177 & 0.88250 \\ 0.74826 & 0.27117 & 0.09633 & 0.16122 \\ 0.10127 & 0.33121 & 0.71177 & 0.65377 \end{bmatrix} \qquad t = \begin{bmatrix} 0.8 \\ 0.6 \\ 0.3 \\ 0.3 \end{bmatrix}$$

$$I = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \qquad \lambda = 0.1$$

$$w = \underbrace{(X^T \cdot X + \lambda \cdot I)^{-1}} \cdot X^T \cdot t$$

$$X^T \cdot X = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 0.74826 & 0.81465 & 0.71177 & 0.88250 \\ 0.74826 & 0.27117 & 0.09633 & 0.16122 \\ 0.10127 & 0.33121 & 0.71177 & 0.65377 \end{bmatrix} \times \begin{bmatrix} 1 & 0.74826 & 0.74826 & 0.10127 \\ 1 & 0.81465 & 0.27117 & 0.33121 \\ 1 & 0.71177 & 0.09633 & 0.71177 \\ 1 & 0.88250 & 0.16122 & 0.65377 \end{bmatrix}$$

$$= \begin{bmatrix} 4 & 3.15718 & 1.27698 & 1.79802 \\ 3.15718 & 2.50897 & 0.99164 & 1.42917 \\ 1.27698 & 0.99164 & 0.66870 & 0.33956 \\ 1.79802 & 1.42917 & 0.33956 & 1.05399 \end{bmatrix}$$

$$\lambda \cdot I = \begin{bmatrix} 0.1 & 0 & 0 & 0 \\ 0 & 0.1 & 0 & 0 \\ 0 & 0 & 0.1 & 0 \\ 0 & 0 & 0 & 0.1 \end{bmatrix} \qquad X^T \cdot X + \lambda \cdot I = \begin{bmatrix} 4.1 & 3.15718 & 1.27698 & 1.79802 \\ 3.15718 & 2.60897 & 0.99164 & 1.42917 \\ 1.27698 & 0.99164 & 0.76870 & 0.33956 \\ 1.79802 & 1.42917 & 0.33956 & 1.15399 \end{bmatrix}$$

$$(X^T \cdot X + \lambda \cdot I)^{-1} = \frac{1}{\underbrace{|X^T \cdot X + \lambda \cdot I|}_{\hookrightarrow \text{ determinant}}} \underbrace{(\widetilde{X^T \cdot X + \lambda \cdot I})}_{\hookrightarrow \text{ adjugate matrix}} = \frac{1}{0.05961} (\widetilde{X^T X + \lambda I})$$

$$(\widetilde{X^T X + \lambda I})_{11} = (-1)^2 \begin{vmatrix} 2.60897 & 0.99164 & 1.42917 \\ 0.99164 & 0.76870 & 0.33956 \\ 1.42917 & 0.33956 & 1.15399 \end{vmatrix}$$

$$\vdots$$

$$(\widetilde{X^T X + \lambda I})_{44} = (-1)^8 \begin{vmatrix} 4.1 & 3.15718 & 1.27698 \\ 3.15718 & 2.60897 & 0.99164 \\ 1.27698 & 0.99164 & 0.76870 \end{vmatrix}$$

And thus, $(X^T \cdot X + \lambda \cdot I)^{-1} = \dfrac{1}{0.05961} \begin{bmatrix} 0.27112 & -0.22514 & -0.11094 & -0.11096 \\ -0.22514 & 0.35664 & -0.05277 & -0.07537 \\ -0.11094 & -0.05277 & 0.25827 & 0.16222 \\ -0.11096 & -0.07537 & 0.16222 & 0.27015 \end{bmatrix} =$

$$= \begin{bmatrix} 4.54819 & -3.77683 & -1.86110 & -1.86140 \\ -3.77683 & 5.98285 & -0.88534 & -1.26436 \\ -1.86110 & -0.88534 & 4.33259 & 2.72137 \\ -1.86140 & -1.26436 & 2.72137 & 4.53189 \end{bmatrix}$$

$$w = \begin{bmatrix} 4.54819 & -3.77683 & -1.86110 & -1.86140 \\ -3.77683 & 5.98285 & -0.88534 & -1.26436 \\ -1.86110 & -0.88534 & 4.33259 & 2.72137 \\ -1.86140 & -1.26436 & 2.72137 & 4.53189 \end{bmatrix} \begin{bmatrix} 1 & 1 & 1 & 1 \\ 0.74826 & 0.81465 & 0.71177 & 0.88250 \\ 0.74826 & 0.27117 & 0.09633 & 0.16122 \\ 0.10127 & 0.33121 & 0.71177 & 0.65377 \end{bmatrix} \begin{bmatrix} 0.8 \\ 0.6 \\ 0.3 \\ 0.3 \end{bmatrix} =$$

$$= \begin{bmatrix} 0.33915 \\ 0.19948 \\ 0.40097 \\ -0.29601 \end{bmatrix}$$

(b) $RMSE = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(t_i - \hat{t}_i)^2}$

$$\begin{array}{l} \;\; y_1 \qquad\quad y_2 \qquad\quad y_3 \\ \phi(x_1) = (0.74826, 0.74826, 0.10127) \\ \phi(x_2) = (0.81465, 0.27117, 0.33121) \\ \phi(x_3) = (0.71177, 0.09633, 0.71177) \\ \phi(x_4) = (0.88250, 0.16122, 0.65377) \end{array}$$

$$w = \begin{bmatrix} w_0 \\ w_1 \\ w_2 \\ w_3 \end{bmatrix} = \begin{bmatrix} 0.33915 \\ 0.19948 \\ 0.40097 \\ -0.29601 \end{bmatrix}$$

hyperplane: $0.33915 + 0.19948 y_1 + 0.40097 y_2 - 0.29601 y_3$

- $0.33915 + 0.19948 \times 0.74826 + 0.40097 \times 0.74826 - 0.29601 \times 0.10127 \simeq 0.75847$
- $0.33915 + 0.19948 \times 0.81465 + 0.40097 \times 0.27117 - 0.29601 \times 0.33121 \simeq 0.51235$
- $0.33915 + 0.19948 \times 0.71177 + 0.40097 \times 0.09633 - 0.29601 \times 0.71177 \simeq 0.30907$
- $0.33915 + 0.19948 \times 0.88250 + 0.40097 \times 0.16122 - 0.29601 \times 0.65377 \simeq 0.38631$

$\hat{t} = (0.75847, 0.51235, 0.30907, 0.38631)$

$t = (0.8, 0.6, 0.3, 0.3)$

$$RMSE = \sqrt{\frac{1}{4}\left((0.8-0.75847)^2 + (0.6-0.51235)^2 + (0.3-0.30907)^2 + (0.3-0.38631)^2\right)} \simeq$$

$\simeq 0.06507$

(2) Batch gradient descent update (2 observations):

(i) define weights and biases

(ii) Forward propagation (input → output): $x^{[p]}$

(iii) compute auxiliary derivatives → $\frac{\delta E}{\delta w}, \frac{\delta E}{\delta b}$

(iv) compute deltas ($\delta^{[p]}$)

(v) compute $\boxed{w^{[p]} = w^{[p]} - \eta \frac{\delta E}{\delta w^{[p]}}}$ and $\boxed{b^{[p]} = b^{[p]} - \eta \frac{\delta E}{\delta b^{[p]}}}$

(i)

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

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

(ii) $\phi^{[p]}(x) = \tanh(0.5x-2)$

Regarding x_1:

$$x^{[1](1)} = \phi^{[1]}(W^{[1]}x^{[0](1)}+b^{[1]}) = \phi^{[1]}\left(\begin{bmatrix}1&1&1&1\\1&1&2&1\\1&1&1&1\end{bmatrix}\begin{bmatrix}1\\1\\1\\1\end{bmatrix}+\begin{bmatrix}1\\1\\1\end{bmatrix}\right) = \phi^{[1]}\begin{pmatrix}5\\6\\5\end{pmatrix} =$$

$$= \begin{bmatrix}\tanh(0.5\times5-2)\\\tanh(0.5\times6-2)\\\tanh(0.5\times5-2)\end{bmatrix} = \begin{bmatrix}0.46212\\0.76159\\0.46212\end{bmatrix}$$

$$x^{[2](1)} = \phi^{[2]}(W^{[2]}x^{[1](1)}+b^{[2]}) = \phi^{[2]}\left(\begin{bmatrix}1&4&1\\1&1&1\end{bmatrix}\begin{bmatrix}0.46212\\0.76159\\0.46212\end{bmatrix}+\begin{bmatrix}1\\1\end{bmatrix}\right) = \phi^{[2]}\begin{pmatrix}4.97060\\2.68583\end{pmatrix} =$$

$$= \begin{bmatrix}0.45048\\-0.57642\end{bmatrix}$$

$$x^{[3](1)} = \phi^{[3]}(W^{[3]}x^{[2](1)}+b^{[3]}) = \phi^{[3]}\left(\begin{bmatrix}1&1\\3&1\\1&1\end{bmatrix}\begin{bmatrix}0.45048\\-0.57642\end{bmatrix}+\begin{bmatrix}1\\1\\1\end{bmatrix}\right) = \phi^{[3]}\begin{pmatrix}0.87406\\1.77502\\0.87406\end{pmatrix} =$$

$$= \begin{bmatrix}-0.91590\\-0.80494\\-0.91590\end{bmatrix}$$

Regarding x_2:

$$x^{[1](2)} = \phi^{[1]}\left(\begin{bmatrix}1&1&1&1\\1&1&2&1\\1&1&1&1\end{bmatrix}\begin{bmatrix}1\\0\\0\\-1\end{bmatrix}+\begin{bmatrix}1\\1\\1\end{bmatrix}\right) = \phi^{[1]}\begin{pmatrix}1\\1\\1\end{pmatrix} = \begin{bmatrix}-0.90515\\-0.90515\\-0.90515\end{bmatrix}$$

$$x^{[2](2)} = \phi^{[2]}\left(\begin{bmatrix}1&4&1\\1&1&1\end{bmatrix}\begin{bmatrix}-0.90515\\-0.90515\\-0.90515\end{bmatrix}+\begin{bmatrix}1\\1\end{bmatrix}\right) = \phi^{[2]}\begin{pmatrix}-4.43090\\-1.71545\end{pmatrix} = \begin{bmatrix}-0.99956\\-0.99343\end{bmatrix}$$

$$x^{[3](2)} = \phi^{[3]}\left(\begin{bmatrix}1&1\\3&1\\1&1\end{bmatrix}\begin{bmatrix}-0.99956\\-0.99343\end{bmatrix}+\begin{bmatrix}1\\1\\1\end{bmatrix}\right) = \phi^{[3]}\begin{pmatrix}-0.99299\\-2.99211\\-0.99299\end{pmatrix} = \begin{bmatrix}-0.98652\\-0.99816\\-0.98652\end{bmatrix}$$

(iii)

$$x^{[p]} = \phi^{[p]}(\underbrace{W^{[p]}x^{[p-1]}+b^{[p]}}_{z^{[p]}})$$

$$E(t, x^{[3]}) = \frac{1}{2}\| x^{[3]} - t\|_2^2$$

$$\frac{\delta E}{\delta x^{[3]}}(x^{[3]}, t) = \frac{1}{2}2(x^{[3]}-t) = x^{[3]}-t$$

$$\frac{\delta x^{[p]}}{\delta z^{[p]}}(z^{[p]}) = \frac{\delta\phi(z^{[p]})}{\delta z^{[p]}} = \phi'(z^{[p]}) = 0.5(1-\tanh^2(0.5z^{[p]}-2))$$

$$\frac{\delta z^{[p]}}{\delta W^{[p]}}(W^{[p]}, x^{[p-1]}, b^{[p]}) = (x^{[p-1]})^T$$

$$\frac{\delta z^{[p]}}{\delta x^{[p-1]}}(W^{[p]}, x^{[p-1]}, b^{[p]}) = W^{[p]}$$

$$\frac{\delta z^{[p]}}{\delta b^{[p]}}(W^{[p]}, x^{[p-1]}, b^{[p]}) = 1$$

And thus, $$\delta^{[3]} = \frac{\delta E}{\delta x^{[3]}} \circ \frac{\delta x^{[3]}}{\delta z^{[3]}} = (x^{[3]} - t) \circ 0.5\left(1 - \tanh^2(0.5 z^{[3]} - 2)\right),$$

↳ error magnitude

↳ output layer

$$\delta^{[2]} = \left(\frac{\delta z^{[3]}}{\delta x^{[2]}}\right)^T \cdot \delta^{[3]} \circ \frac{\delta x^{[2]}}{\delta z^{[2]}} = (W^{[3]})^T \cdot \delta^{[3]} \circ 0.5\left(1 - \tanh^2(0.5 z^{[2]} - 2)\right)$$

and $$\delta^{[1]} = \left(\frac{\delta z^{[2]}}{\delta x^{[1]}}\right)^T \cdot \delta^{[2]} \circ \frac{\delta x^{[1]}}{\delta z^{[1]}} = (W^{[2]})^T \cdot \delta^{[2]} \circ 0.5\left(1 - \tanh^2(0.5 z^{[1]} - 2)\right)$$

↳ first layers (backpropagation)

(iv)

since there are three outcomes (A, B and C) and the target is B in regard to x_1:

$t = B = \begin{pmatrix} -1 \\ 1 \\ -1 \end{pmatrix}$ (due to the studied properties of tanh activation – codomain in $[-1,1]$ –, the encodings can't be $(0\ 1\ 0)^T$)

$$\delta^{[3](1)} = (x^{[3]} - t) \circ \phi'(z^{[3]}) = \left(\begin{bmatrix} -0.91590 \\ -0.80494 \\ -0.91590 \end{bmatrix} - \begin{bmatrix} -1 \\ 1 \\ -1 \end{bmatrix}\right) \circ \phi'\begin{pmatrix} 0.87406 \\ 1.77502 \\ 0.87406 \end{pmatrix} =$$

$$= \begin{bmatrix} 0.08410 \\ -1.80494 \\ 0.08410 \end{bmatrix} \circ \begin{bmatrix} 0.5(1 - \tanh^2(0.5 \cdot 0.87406 - 2)) \\ 0.5(1 - \tanh^2(0.5 \cdot 1.77502 - 2)) \\ 0.5(1 - \tanh^2(0.5 \cdot 0.87406 - 2)) \end{bmatrix} = \begin{bmatrix} 6.77540 \times 10^{-3} \\ -0.31773 \\ 6.77540 \times 10^{-3} \end{bmatrix}$$

$$\delta^{[2](1)} = (W^{[3]})^T \cdot \delta^{[3](1)} \circ \phi'(z^{[2]}) = \begin{bmatrix} 1 & 3 & 1 \\ 1 & 1 & 1 \end{bmatrix} \begin{bmatrix} 6.77540 \times 10^{-3} \\ -0.31773 \\ 6.77540 \times 10^{-3} \end{bmatrix} \circ \phi'\begin{pmatrix} 4.97060 \\ 2.68583 \end{pmatrix} =$$

$$= \begin{bmatrix} -0.93964 \\ -0.30418 \end{bmatrix} \circ \phi'\begin{pmatrix} 4.97060 \\ 2.68583 \end{pmatrix} = \begin{bmatrix} -0.37448 \\ -0.10156 \end{bmatrix}$$

$$\delta^{[1](1)} = (W^{[2]})^T \cdot \delta^{[2](1)} \circ \phi'(z^{[1]}) = \begin{bmatrix} 1 & 1 \\ 4 & 1 \\ 1 & 1 \end{bmatrix} \begin{bmatrix} -0.37448 \\ -0.10156 \end{bmatrix} \circ \phi'\begin{pmatrix} 5 \\ 6 \\ 5 \end{pmatrix} =$$

$$= \begin{bmatrix} -0.47904 \\ -1.60248 \\ -0.47904 \end{bmatrix} \circ \begin{bmatrix} 0.39322 \\ 0.20999 \\ 0.39322 \end{bmatrix} = \begin{bmatrix} -0.18837 \\ -0.33650 \\ -0.18837 \end{bmatrix}$$

since there are three outcomes (A, B and C) and the target is A in regard to x_2:

$t = A = \begin{pmatrix} 1 \\ -1 \\ -1 \end{pmatrix}$ (due to the studied properties of tanh activation – codomain in $[-1,1]$ –, the encodings can't be $(1\ 0\ 0)^T$)

$$\delta^{[3](2)} = \left(\begin{bmatrix} -0.98652 \\ -0.99816 \\ -0.98652 \end{bmatrix} - \begin{bmatrix} 1 \\ -1 \\ -1 \end{bmatrix}\right) \circ \phi'\begin{pmatrix} -0.99299 \\ -2.99211 \\ -0.99299 \end{pmatrix} = \begin{bmatrix} -1.98652 \\ 1.84000 \times 10^{-3} \\ 0.01348 \end{bmatrix} \circ \begin{bmatrix} 0.01339 \\ 1.83831 \times 10^{-3} \\ 0.01339 \end{bmatrix} =$$

$$= \begin{bmatrix} -0.02660 \\ 3.38249 \times 10^{-6} \\ 1.80497 \times 10^{-4} \end{bmatrix}$$

$$\delta^{[2](2)} = \begin{bmatrix} 1 & 3 & 1 \\ 1 & 1 & 1 \end{bmatrix} \begin{bmatrix} -0.02660 \\ 3.38249 \times 10^{-6} \\ 1.80497 \times 10^{-4} \end{bmatrix} \circ \phi'\begin{pmatrix} -4.43090 \\ -1.71545 \end{pmatrix} = \begin{bmatrix} -0.02641 \\ -0.02642 \end{bmatrix} \circ \begin{bmatrix} 4.39903 \times 10^{-4} \\ 6.54842 \times 10^{-3} \end{bmatrix} =$$

$$= \begin{bmatrix} -1.16178 \times 10^{-5} \\ -1.73009 \times 10^{-4} \end{bmatrix}$$

$$\delta^{[1](2)} = \begin{bmatrix} 1 & 1 \\ 1 & 1 \\ 1 & 1 \end{bmatrix} \begin{bmatrix} -1.16178 \times 10^{-5} \\ -1.73009 \times 10^{-4} \end{bmatrix} \circ \phi'\begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix} = \begin{bmatrix} -1.84627 \times 10^{-4} \\ -2.19480 \times 10^{-4} \\ -1.84627 \times 10^{-4} \end{bmatrix} \circ \begin{bmatrix} 0.09035 \\ 0.09035 \\ 0.09036 \end{bmatrix} = \begin{bmatrix} -1.66810 \times 10^{-5} \\ -1.98300 \times 10^{-5} \\ -1.66810 \times 10^{-5} \end{bmatrix}$$

$$\frac{\delta E}{\delta W^{[1]}} = \delta^{[1](1)} \left(x^{[0](1)} \right)^T + \delta^{[1](2)} \left(x^{[0](2)} \right)^T =$$

$$= \begin{bmatrix} -0.18837 \\ -0.33650 \\ -0.18837 \end{bmatrix} \begin{bmatrix} 1 & 1 & 1 & 1 \end{bmatrix} + \begin{bmatrix} -1.66810 \times 10^{-5} \\ -1.98300 \times 10^{-5} \\ -1.66810 \times 10^{-5} \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & -1 \end{bmatrix} =$$

$$= \begin{bmatrix} -0.18839 & -0.18837 & -0.18837 & -0.18835 \\ -0.33652 & -0.33650 & -0.33650 & -0.33648 \\ -0.18839 & -0.18837 & -0.18837 & -0.18835 \end{bmatrix}$$

$$\frac{\delta E}{\delta W^{[2]}} = \delta^{[2](1)} \left(x^{[1](1)} \right)^T + \delta^{[2](2)} \left(x^{[1](2)} \right)^T =$$

$$= \begin{bmatrix} -0.37448 \\ -0.10156 \end{bmatrix} \begin{bmatrix} 0.46212 & 0.76159 & 0.46212 \end{bmatrix} +$$

$$+ \begin{bmatrix} -1.16178 \times 10^{-5} \\ -1.73009 \times 10^{-4} \end{bmatrix} \begin{bmatrix} -0.90515 & -0.90515 & -0.90515 \end{bmatrix} =$$

$$= \begin{bmatrix} -0.17304 & -0.28519 & -0.17304 \\ -0.04678 & -0.07719 & -0.04678 \end{bmatrix}$$

$$\frac{\delta E}{\delta W^{[3]}} = \delta^{[3](1)} \left(x^{[2](1)} \right)^T + \delta^{[3](2)} \left(x^{[2](2)} \right)^T =$$

$$= \begin{bmatrix} 6.77540 \times 10^{-3} \\ -0.31773 \\ 6.77540 \times 10^{-3} \end{bmatrix} \begin{bmatrix} 0.45048 & -0.57642 \end{bmatrix} +$$

$$+ \begin{bmatrix} -0.02660 \\ 3.38249 \times 10^{-6} \\ 1.80497 \times 10^{-4} \end{bmatrix} \begin{bmatrix} -0.99956 & -0.99343 \end{bmatrix} =$$

$$= \begin{bmatrix} 0.02964 & 0.02252 \\ -0.14313 & 0.18314 \\ 2.87176 \times 10^{-3} & -4.08479 \times 10^{-3} \end{bmatrix}$$

$$\frac{\delta E}{\delta b^{[1]}} = \delta^{[1](1)} + \delta^{[1](2)} = \begin{bmatrix} -0.18839 \\ -0.33652 \\ -0.18839 \end{bmatrix}$$

$$\frac{\delta E}{\delta b^{[2]}} = \delta^{[2](1)} + \delta^{[2](2)} = \begin{bmatrix} -0.37449 \\ -0.10173 \end{bmatrix}$$

$$\frac{\delta E}{\delta b^{[3]}} = \delta^{[3](1)} + \delta^{[3](2)} = \begin{bmatrix} -0.01982 \\ -0.31773 \\ 6.95590 \times 10^{-3} \end{bmatrix}$$

(v) we can now update both the weights and the biases:

hidden layer 1

$$W^{[1]} = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & 2 & 1 \\ 1 & 1 & 1 & 1 \end{bmatrix} - 0.1 \frac{\delta E}{\delta W^{[1]}} = \begin{bmatrix} 1.01884 & 1.01884 & 1.01884 & 1.01884 \\ 1.03365 & 1.03365 & 2.03365 & 1.03365 \\ 1.01884 & 1.01884 & 1.01884 & 1.01884 \end{bmatrix}$$

$$b^{[1]} = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} - 0.1 \frac{\delta E}{\delta b^{[1]}} = \begin{bmatrix} 1.01884 \\ 1.03365 \\ 1.01884 \end{bmatrix}$$

hidden layer 2

$$W^{[2]} = \begin{bmatrix} 1 & 4 & 1 \\ 1 & 1 & 1 \end{bmatrix} - 0.1 \frac{\delta E}{\delta W^{[2]}} = \begin{bmatrix} 1.01730 & 4.02852 & 1.01730 \\ 1.00468 & 1.00772 & 1.00468 \end{bmatrix}$$

$$b^{[2]} = \begin{bmatrix} 1 \\ 1 \end{bmatrix} - 0.1 \frac{\delta E}{\delta b^{[2]}} = \begin{bmatrix} 1.03745 \\ 1.01017 \end{bmatrix}$$

output layer

$$W^{[3]} = \begin{bmatrix} 1 & 1 \\ 3 & 1 \\ 1 & 1 \end{bmatrix} - 0.1 \frac{\delta E}{\delta W^{[3]}} = \begin{bmatrix} 0.99704 & 0.99775 \\ 3.01431 & 0.98169 \\ 0.99971 & 1.00041 \end{bmatrix}$$

$$b^{[3]} = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} - 0.1 \frac{\delta E}{\delta b^{[3]}} = \begin{bmatrix} 1.00198 \\ 1.03177 \\ 0.99930 \end{bmatrix}$$

Aprendizagem 2023/24

Homework III – Group 029

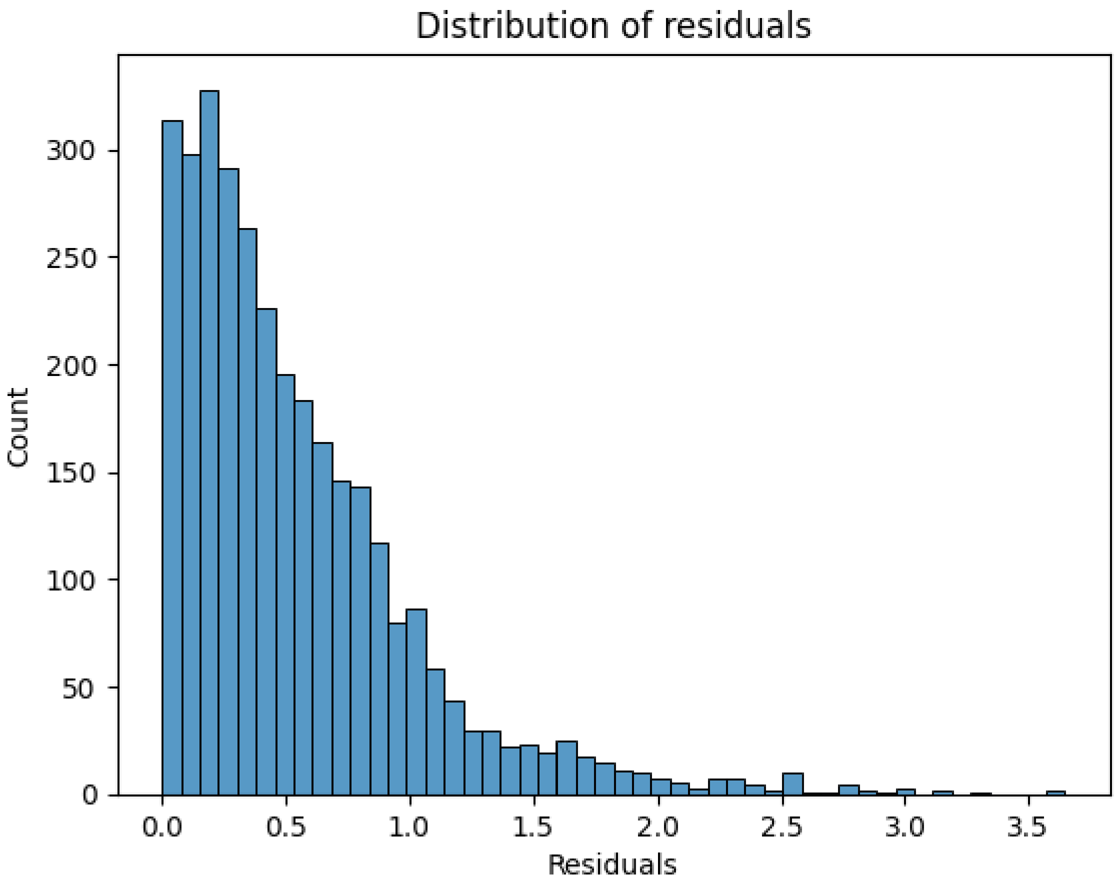
(ist1103000, ist1103179)

II. Programming and critical analysis

1)

Code and graphs:

```
import pandas as pd

import warnings
warnings.filterwarnings("ignore")

wine = pd.read_csv("winequality-red.csv", sep=";")

wine.head()

input_vars = wine.drop("quality", axis=1)
output_vars = wine["quality"]

from sklearn.model_selection import train_test_split

X_train, X_test, y_train, y_test = train_test_split(input_vars, output_vars,
stratify=output_vars, train_size=0.8, random_state=0)
print("#training obs =",len(X_train),"\n#testing obs =",len(X_test))

from sklearn.neural_network import MLPRegressor


all_residuals = []

for it in range(10):
    MLP_regressor = MLPRegressor(random_state=(it+1),
hidden_layer_sizes=(10,10), activation="relu", early_stopping=True,
validation_fraction=0.2)
    MLP_regressor.fit(X_train, y_train)

    y_pred_test = MLP_regressor.predict(X_test)

    residual = abs(y_pred_test - y_test)
    all_residuals.extend(residual)

import seaborn as sns

plot = sns.histplot(data= all_residuals).set(title="Distribution of
residuals", xlabel="Residuals")
```



2)

Code and graphs:

```
from sklearn.metrics import mean_absolute_error, mean_squared_error

MAE_no_round = 0
MAE_round = 0
RMSE_no_round = 0       # used in 3)

for it in range(10):
    MLP_regressor = MLPRegressor(random_state=(it+1),
hidden_layer_sizes=(10,10), activation="relu", early_stopping=True,
validation_fraction=0.2)
    MLP_regressor = MLP_regressor.fit(X_train, y_train)
    y_pred_test = MLP_regressor.predict(X_test)
    y_pred_test_round = [10 if x>10 else 1 if x<1 else round(x) for x in
y_pred_test]

    MAE_no_round += mean_absolute_error(y_test, y_pred_test)
    MAE_round += mean_absolute_error(y_test, y_pred_test_round)
    RMSE_no_round += mean_squared_error(y_test, y_pred_test, squared=False)

MAE_no_round /= 10
MAE_round /= 10
RMSE_no_round /= 10
```

```
print("MAE without bounding and rounding: ", MAE_no_round)
print("MAE with bounding and rounding: ", MAE_round)
```

Output:
```
MAE without bounding and rounding:  0.5437511706983347
MAE with bounding and rounding:  0.49281250000000004
```

Comment:

As is possible to verify above, we can assess that rounding our estimates reduces the MAE when compared to the MLP learnt in the previous question.

3)

Code and graphs:

```
rmse_early_stop = RMSE_no_round

n_iterations = [20, 50, 100, 200]

rmse_it = []
y_pred_test2 = [0]*y_test.size

for it in n_iterations:
    rmse = 0
    for it2 in range(10):
        MLP_regressor2 = MLPRegressor(random_state=(it2+1),
hidden_layer_sizes=(10,10), activation="relu", early_stopping=False,
max_iter=it)
        MLP_regressor2.fit(X_train, y_train)
        y_pred_test2 = MLP_regressor2.predict(X_test)

        rmse += mean_squared_error(y_test, y_pred_test2, squared=False)

    y_pred_test2 /= 10
    rmse /= 10
    rmse_it.append(rmse)

print("Early stop RMSE: ", rmse_early_stop)
for i in range(4):
 print(str(n_iterations[i]) + " iterations RMSE: " + str(rmse_it[i]))
```

Output:
```
Early stop RMSE:  0.7285645002031444
20 iterations RMSE: 1.5741537078556829
50 iterations RMSE: 0.9296215581878041
100 iterations RMSE: 0.7495674512113977
200 iterations RMSE: 0.697002682560151
```

4)

By analyzing the results above, we are able to infer that early stopping has a lower RMSE, and an overall better performance, when compared to having a well-defined number of iterations (especially when that number doesn't exceed 100 iterations). This may be the case mainly because of overfitting, since the regressor with early stopping stops training when the validation score is not improving, leading to a better generalization capacity when compared to a fixed number of iterations, which relies too much on the training set. It is also possible to note that, as the number of iterations increases, the RMSE does not decrease significantly, meaning that more iterations doesn't necessarily mean better performance.

END