

# Homework III

Francisco Silva : 110409

Jocina Cardoso : 109864

## 1. Pen and Paper

1. OLS (Ordinary Least Squares) procura os pesos  $w_i$  que minimizam a função de custo, que é a soma dos erros quadráticos (Sum of squared Error)  $SSE = \sum_{i=1}^N (\hat{z}_i - z_i)^2$

$$\textcircled{1} (y_1, y_2) = \begin{bmatrix} 4 \\ 1 \\ 6 \\ 18 \\ 8 \end{bmatrix}; Y_{\text{num}} = \begin{bmatrix} 3,5 \\ 1 \\ 3,8 \\ 10,1 \\ 8,5 \end{bmatrix} = z; X = \begin{bmatrix} 1 & 4 \\ 1 & 1 \\ 1 & 6 \\ 1 & 18 \\ 1 & 8 \end{bmatrix}; X^T = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 \\ 4 & 1 & 6 & 18 & 8 \end{bmatrix};$$

$$\underset{\text{OLS}}{W} = (X^T X)^{-1} X^T z = \left( \begin{bmatrix} 5 & 37 \\ 37 & 441 \end{bmatrix} \right)^{-1} \begin{bmatrix} 1 & 1 & 1 & 1 & 1 \\ 4 & 1 & 6 & 18 & 8 \end{bmatrix} \begin{bmatrix} 3,5 \\ 1 \\ 3,8 \\ 10,1 \\ 8,5 \end{bmatrix} =$$

Usando numpy.linalg.pinv

$$= \begin{bmatrix} 0,52751 & -0,044426 \\ -0,044426 & 0,00598 \end{bmatrix} \begin{bmatrix} 1 & 1 & 1 & 1 & 1 \\ 4 & 1 & 6 & 18 & 8 \end{bmatrix} \begin{bmatrix} 3,5 \\ 1 \\ 3,8 \\ 10,1 \\ 8,5 \end{bmatrix} = \begin{bmatrix} 1,46136 \\ 0,52955 \end{bmatrix} \rightarrow \text{Mo ab}$$

como preveremos novos outputs  
 $\hat{z}_i = x_i^T W_{\text{OLS}}$

2. Ridge visa prevenir o overfitting, tornando o modelo menos complexo e menos sensível a pequenas variações das dades de treino  $E(w) = \frac{1}{n} \sum_{i=1}^n (\hat{z}_i - z_i)^2 + \frac{\lambda}{2} \|w\|_2^2$

$$\lambda = 1; I \text{ é } 2 \times 2 \text{ pq } X^T X \text{ é } 2 \times 2$$

$$\underset{\text{Ridge}}{W} = (X^T X + \lambda I)^{-1} X^T z = \left( \begin{bmatrix} 5 & 37 \\ 37 & 441 \end{bmatrix} + \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \right)^{-1} X^T z = \begin{bmatrix} 0,34451 & -0,02834 \\ -0,02834 & 0,00463 \end{bmatrix} \begin{bmatrix} 26,9 \\ 287,6 \end{bmatrix} =$$

$$= \begin{bmatrix} 0,97319 \\ 0,56921 \end{bmatrix} \rightarrow \text{Mo ab}$$

$$\hat{z}_i = x_i^T W_{\text{Ridge}} \text{ como preveremos os novos outputs}$$

★ continua na última pag

3. Avaliar a performance dos 2 modelos que "aprendemos" usando Mean Absolute Error:  $MAE = \frac{1}{n} \sum_{i=1}^n |\hat{z}_i - z_i|$

$$\text{Para OLS, } W_{\text{OLS}} = \begin{bmatrix} 1,46136 \\ 0,52955 \end{bmatrix}, \hat{z}_i = x_i^T W_{\text{OLS}}, x_i^T = [1 \ 0]_i$$

$$\hat{z}_1 = 3,57956 \quad \hat{z}_2 = 1,99091 \quad \hat{z}_3 = 4,63866 \quad \hat{z}_4 = 10,94326 \quad \hat{z}_5 = 5,69776$$

$$MAE_{\text{train}} = \frac{|3,5 - \hat{z}_1| + |1 - \hat{z}_2| + |3,8 - \hat{z}_3| + |10,1 - \hat{z}_4| + |8,5 - \hat{z}_5|}{5} = \frac{5,60463}{5} = 1,12093 \quad \text{Train}$$

$$\text{Test: } x_6^T = [1 \ 0 \ 1 \ 2] = [1 \ 0] \quad x_7^T = [1 \ 3 \ 1 \ 4] = [1 \ 1 \ 2 \ 3] \quad x_8^T = [1 \ 5 \ 1 \ 1] = [1 \ 5]$$

$$\hat{z}_6 = 1,46136 \quad \hat{z}_7 = 7,816 \quad \hat{z}_8 = 4,10911$$

$$MAE_{\text{test}} = \frac{|1 - \hat{z}_6| + |6,2 - \hat{z}_7| + |3,6 - \hat{z}_8|}{3} = \frac{2,58647}{3} = 0,86216$$

$$\text{Para Ridge, } W_{\text{Ridge}} = \begin{bmatrix} 0,97319 \\ 0,56921 \end{bmatrix}, \hat{z}_i = x_i^T W_{\text{Ridge}}, x_i^T = [1 \ 0]_i$$

$$\hat{z}_1 = 3,25003 \quad \hat{z}_2 = 1,54240 \quad \hat{z}_3 = 4,38845 \quad \hat{z}_4 = 11,24 \quad \hat{z}_5 = 5,52687$$

$$MAE_{\text{train}} = \frac{|3,5 - \hat{z}_1| + |1 - \hat{z}_2| + |3,8 - \hat{z}_3| + |10,1 - \hat{z}_4| + |8,5 - \hat{z}_5|}{5} = \frac{5,47292}{5} = 1,09458$$

Test:

$$\hat{z}_6 = 0,97319 \quad \hat{z}_7 = 7,80371 \quad \hat{z}_8 = 3,81924$$

$$MAE_{\text{test}} = \frac{|1 - \hat{z}_6| + |6,2 - \hat{z}_7| + |3,6 - \hat{z}_8|}{3} = \frac{1,84976}{3} = 0,61659$$

Modelo	Train MAE	Test MAE
OLS	1,12093	0,86216
Ridge	1,09458	0,61659

1. Erro de treino: Em teoria, o OLS deveria sempre ter um erro de treino igual ou menor, pois o seu único objetivo é minimizar o erro. Podemos explicar esta pequena diferença de o Ridge ter um erro de treino menor a, neste exercício estarmos a analisar o MAE enquanto o OLS minimiza o SSE. Mas o importante é que os erros de treino são muito próximos, mostrando que o modelo Ridge não sacrificou muito o seu ajuste aos dados de treino.

2. O teste MAE do modelo Ridge é significativamente mais baixo. Isto é exatamente o que se espera de um modelo regularizado. Ao criar um modelo mais "simples" e menos "acurado" nos dados de treino, a Regressão Ridge consegue generalizar melhor para os novos dados de teste.

4. temos  $b^{[1]}, w^{[1]}, b^{[2]}, w^{[2]}$

$\begin{matrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \end{matrix}$   
 $\downarrow$   
 $L_0 \quad 1 \quad 2$   
layer de inputs  
inner layers de pesos

Na layer 1 a activation function é a sigmoide  
Na layer 2 a activation function é o softmax, layer de output

Stochastic Gradient descent  $c_1 x_1, x_1' = [2 \ 2], n = 0,5$

1° Forward Propagation

$$z^{[2]} \rightarrow x^{[1]} \rightarrow z^{[1]} \rightarrow x^{[2]}, \text{ regras: } z^{[i]} = w^{[i]} x^{[i-1]} + b^{[i]}, x^{[i]} = AF(z^{[i]})$$

$$z^{[1]} = w^{[1]} x^{[0]} + b^{[1]} = \begin{bmatrix} 0,1 & 0,1 \\ 0,1 & 0,2 \\ 0,2 & 0,1 \end{bmatrix} \begin{bmatrix} 2 \\ 2 \end{bmatrix} + \begin{bmatrix} 0,4 \\ 0,6 \\ 0,1 \end{bmatrix} = \begin{bmatrix} 0,5 \\ 0,6 \\ 0,7 \end{bmatrix}$$

$$x^{[1]} = G(z^{[1]})$$

$$\therefore G(0,5) = \frac{1}{1+e^{-0,5}} \approx 0,62246 \quad \therefore G(0,6) = \frac{1}{1+e^{-0,6}} \quad \therefore G(0,7) = \frac{1}{1+e^{-0,7}}$$

$$x^{[1]} = \begin{bmatrix} G(0,5) \\ G(0,6) \\ G(0,7) \end{bmatrix}^T = \begin{bmatrix} 0,62246 \\ 0,64566 \\ 0,66819 \end{bmatrix}^T$$

$$z^{[2]} = w^{[2]} x^{[1]} + b^{[2]} = \begin{bmatrix} 1 & 2 & 2 \\ 1 & 2 & 1 \\ 1 & 1 & 1 \end{bmatrix} \begin{bmatrix} 0,62246 \\ 0,64566 \\ 0,66819 \end{bmatrix} + \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} = \begin{bmatrix} 4,25016 \\ 3,58197 \\ 2,93631 \end{bmatrix}$$

$$x^{[2]} = \text{softmax}(z^{[2]}) = \frac{e^{4,25016}}{e^{4,25016} + e^{3,58197} + e^{2,93631}} = 124,90485$$

$$x^{[2]} = \begin{bmatrix} \text{softmax}(4,25016) \\ \text{softmax}(3,58197) \\ \text{softmax}(2,93631) \end{bmatrix}^T = \begin{bmatrix} e^{4,25016}/124,90485 \\ e^{3,58197}/124,90485 \\ e^{2,93631}/124,90485 \end{bmatrix}^T = \begin{bmatrix} 0,56135 \\ 0,28977 \\ 0,15088 \end{bmatrix} = \hat{z}; z = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}$$

como a classe verdadeira é a A, então este é o nosso vetor do valor real

representa a prob de  $x_1$  pertencer a cada uma das classes A,B,C, respectivamente.

2° Back Propagation

importante: Para a combinação de softmax e cross entropy loss, o gradiente é simplesmente a previsão menos o valor real: resíduos:  $\hat{z} - z$

$$\delta^{[2]} = \frac{\partial E}{\partial z^{[2]}} = \hat{z} - z = \begin{bmatrix} -0,43865 \\ 0,28977 \\ 0,15088 \end{bmatrix}$$

$$\delta^{[1]} = \left( \frac{\partial z^{[2]}}{\partial x^{[1]}} \right)^T \delta^{[2]} = (w^{[2]})^T \delta^{[2]} G(z^{[1]}) = (w^{[2]})^T \delta^{[2]} (G(z^{[1]})(1-G(z^{[1]})) = \\ = \begin{bmatrix} 1 & 1 & 1 \\ 2 & 2 & 1 \\ 2 & 1 & 1 \end{bmatrix} \begin{bmatrix} -0,43865 \\ 0,28977 \\ 0,15088 \end{bmatrix} (x^{[1]}(1-x^{[1]})) = \begin{bmatrix} 0 \\ -0,19088 \\ -0,43865 \end{bmatrix} \begin{bmatrix} 0,62246 \times (1-0,62246) \\ 0,64566 \times (1-0,64566) \\ 0,66819 \times (1-0,66819) \end{bmatrix} = \begin{bmatrix} 0 \\ -0,03452 \\ -0,09725 \end{bmatrix}$$

3° update the weights and biases

$$\text{Weights: } w^{[1]\text{new}} = w^{[1]\text{old}} - \eta \frac{\partial E}{\partial w^{[1]}} = w^{[1]\text{old}} - \eta \delta^{[1]} \frac{\partial z^{[1]}}{\partial w^{[1]}} = w^{[1]\text{old}} - \eta \delta^{[1]} x^{[0]} \quad \begin{bmatrix} 0,1 & 0,1 \\ 0,1 & 0,2 \\ 0,2 & 0,1 \end{bmatrix} - 0,5 \begin{bmatrix} 0 \\ -0,03452 \\ -0,09725 \end{bmatrix} = \begin{bmatrix} 0,1 & 0,1 \\ 0,1 & 0,2 \\ 0,2 & 0,1 \end{bmatrix} - 0,5 \begin{bmatrix} 0 \\ -0,06904 \\ -0,19458 \end{bmatrix} = \begin{bmatrix} 0,1 & 0,1 \\ 0,13452 & 0,23452 \\ 0,29725 & 0,19725 \end{bmatrix}$$

→ Para  $i=1$ , layer 1

$$w^{[1]} = \begin{bmatrix} 0,1 & 0,1 \\ 0,1 & 0,2 \\ 0,2 & 0,1 \end{bmatrix} - 0,5 \begin{bmatrix} 0 \\ -0,03452 \\ -0,09725 \end{bmatrix} = \begin{bmatrix} 0,1 & 0,1 \\ 0,1 & 0,2 \\ 0,2 & 0,1 \end{bmatrix} - 0,5 \begin{bmatrix} 0 \\ -0,06904 \\ -0,19458 \end{bmatrix} = \begin{bmatrix} 0,1 & 0,1 \\ 0,13452 & 0,23452 \\ 0,29725 & 0,19725 \end{bmatrix}$$

→ Para  $i=2$ , layer 2

$$w^{[2]} = \begin{bmatrix} 1 & 2 & 2 \\ 1 & 2 & 1 \\ 1 & 1 & 1 \end{bmatrix} - 0,5 \begin{bmatrix} -0,43865 \\ 0,28977 \\ 0,15088 \end{bmatrix} \begin{bmatrix} 0,62246 & 0,64566 & 0,66819 \end{bmatrix} = \begin{bmatrix} 1 & 2 & 2 \\ 1 & 2 & 1 \\ 1 & 1 & 1 \end{bmatrix} - 0,5 \begin{bmatrix} -0,27304 & -0,28325 & -0,29309 \\ 0,17918 & 0,18583 & 0,19227 \\ 0,09392 & 0,09742 & 0,10081 \end{bmatrix} = \begin{bmatrix} 1,13652 & 2,14163 & 2,14655 \\ 0,91041 & 1,90708 & 0,90386 \\ 0,95364 & 0,95129 & 0,94959 \end{bmatrix}$$

$$\text{bias: } b^{[i]_{\text{new}}} = b^{[i]_{\text{old}}} - \eta \frac{\partial E}{\partial b^{[i]}} = b^{[i]_{\text{old}}} - \eta \delta^{[i]} \frac{\partial \epsilon^{[i]}}{\partial b^{[i]}} = b^{[i]_{\text{old}}} - \eta \delta^{[i]}$$

→ Para  $i=1$ ; layer 1

$$b^{[1]} = \begin{bmatrix} 0,1 \\ 0 \\ 0,1 \end{bmatrix} - 0,5 \begin{bmatrix} 0 \\ -0,03452 \\ -0,09725 \end{bmatrix} = \begin{bmatrix} 0,1 \\ 0,01726 \\ 0,14863 \end{bmatrix} = \begin{bmatrix} 0,1 \\ 0,01726 \\ 0,14863 \end{bmatrix}$$

→ Para  $i=2$ ; layer 2

$$b^{[2]} = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} - 0,5 \begin{bmatrix} -0,43865 \\ 0,28777 \\ 0,15088 \end{bmatrix} = \begin{bmatrix} 1,21933 \\ 0,85611 \\ 0,92456 \end{bmatrix}$$

A diferença fundamental entre usar uma função ativação Sigmoid e não usar define a capacidade do modelo de aprender para além de relações lineares

→ Um MLP sem funções de ativação não-lineares, independentemente do número de camadas, é matematicamente equivalente a um único modelo linear.

Permanece incapaz de resolver problemas que não sejam linearmente separáveis

→ A função Sigmoid introduz a não-linearidade, que é o que dá o poder às redes neurais (multilayer). Assim, a rede torna-se capaz de aprender fronteiras de decisão complexas e não-lineares.

## 2. ⚡ Continuação

	OLS	Ridge ( $\lambda=1$ )
$u_0(\text{bias})$	1,46136	0,97319
$u_1(0)$	0,52955	0,56921

Ao aplicar a Regressão Ridge c/  $\lambda=1$  fomos o modelo a encontrar um equilíbrio entre minimizar o erro de previsão e manter a magnitude dos seus coeficientes baixa. A descida acentuada do bias é o resultado mais direto da regularização.

A penalidade Ridge foi aplicada em ambos os coeficientes, "puxando-os" em direção a 0 para reduzir a complexidade geral do modelo.

A pequena subida de  $u_1$  demonstra que Ridge optimiza coeficientes em conjunto, ou seja, surge como forma de compensar a forte redução do bias.

$$\|W_{\text{OLS}}\|_2 = 1,55435 \quad \|W_{\text{Ridge}}\|_2 = 1,2742$$

$\|W_{\text{Ridge}}\|_2 < \|W_{\text{OLS}}\|_2$ , confirma que a regularização cumpre o seu objetivo de reduzir a magnitude total do modelo.