Aprendizagem 2024/2025

Homework III – Group 034

(ist199417, ist1106481)

I. Pen-and-paper

1)

D	y_1	y_2	$\phi(y_1, y_2)$	y_{num}
x_1	1	1	1	1.25
x_2	1	3	3	7
x_3	3	2	6	2.7
x_4	3	3	9	3.2
x_5	2	4	8	5.5

Definimos a matriz X e vetor z:

$$X = \begin{bmatrix} 1 & 1 \\ 1 & 3 \\ 1 & 6 \\ 1 & 9 \\ 1 & 8 \end{bmatrix} \quad z = \begin{bmatrix} 1.25 \\ 7 \\ 2.7 \\ 3.2 \\ 5.5 \end{bmatrix}$$

Usando numpy para os cálculos, temos:

$$w = (X^T \cdot X)^{-1} \cdot X^T \cdot z = \begin{bmatrix} 3.31593 \\ 0.11372 \end{bmatrix}$$

2)

Utilizando a mesma matriz X e o mesmo vetor z do exercício 1 e numpy para os cálculos, temos:

$$w = (X^T \cdot X + \lambda I)^{-1} \cdot X^T \cdot z = \begin{bmatrix} 1.81809 \\ 0.32376 \end{bmatrix}$$

O fator de regularização ajuda a combater o overfitting, ao diminuir a magnitude de coeficientes sobrestimados ou subestimados. É precisamente isso que vemos a acontecer. Os coeficientes da pergunta anterior são (3.31593, 0.11372) enquanto que os coeficientes com regularização são (1.81809, 0.32376). O coeficiente 3.31593 foi diminuído e o coeficiente 0.11372 foi aumentado.

3)

$$RMSE(\hat{z}, z) = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(z_i - \hat{z}_i)}$$

D	y_1	y_2	$\phi(y_1, y_2)$	y_{num}
x_6	2	2	4	0.7
x_7	1	2	2	1.1
x_8	5	1	5	2.2

	D	OLS	Ridge
Treino	x_1	3.42965	2.14184
	x_2	3.65708	2.78936
	x_3	3.99823	3.76064
	x_4	4.33938	4.73191
	x_5	4.22566	4.40816
Teste	x_6	3.77080	3.11312
	x_7	3.54336	2.46560
	x_8	3.88451	3.43688

Com os dados do enunciado, obtemos z e, com os dados desta tabela, obtemos $\hat{z}$.

(Treino, OLS)

$z = (1.25;\ 7;\ 2.7;\ 3.2;\ 5.5)$ $\hat{z} = (3.42965;\ 3.65708;\ 3.99823;\ 4.33938;\ 4.22566)$

$RMSE = 2.02650$

(Treino, Ridge)

$z = (1.25;\ 7;\ 2.7;\ 3.2;\ 5.5)$ $\hat{z} = (2.14184;\ 2.78936;\ 3.76064;\ 4.73191;\ 4.40816)$

$RMSE = 2.15354$

(Teste, OLS)

$z = (0.7;\ 1.1;\ 2.2)$ $\hat{z} = (3.77080;\ 3.54336;\ 3.88451)$

$RMSE = 2.46559$

(Treino, Ridge)

$z = (0.7;\ 1.1;\ 2.2)$ $\hat{z} = (3.11312;\ 2.46560;\ 3.43688)$

$RMSE = 1.75289$

O resultado obtido foi de encontro com o esperado. Ao adicionarmos o fator de regularização, estamos a tentar diminuir o overfitting, podendo o modelo ser pior no conjunto de treino, para depois vir a ser melhor nos dados de teste. Como vemos, no grupo de treino, o RMSE sem

regularização foi de 2.02650, enquanto que o RMSE com regularização foi de 2.15354. Portanto, de facto, no grupo de treino, a regularização levou a um aumento do erro da previsão em relação ao valor verdadeiro de ynum. Agora no grupo de teste, o RMSE sem regularização foi de 2.46559, enquanto que o RMSE com regularização foi de 1.75281. Ou seja, o modelo com regularização teve um resultado muito mais satisfatório no grupo de teste do que o modelo sem regularização. Foi até o menor erro entre todos os erros calculados!

4)

(Usámos numpy para fazer os cálculos)

$$Input = x_1 = \begin{bmatrix}1\\1\end{bmatrix} = i, \qquad Resultados\ esperado = \begin{bmatrix}0\\1\\0\end{bmatrix} = y$$

Pesos e Bias atualizados:

$$W^{[x]\prime} = W^{[x]} - \eta \frac{\partial E\left(W^{[x]}\right)}{\partial W^{[x]}}\ , x = 1, 2$$

$$b^{[x]\prime} = b^{[x]} - \eta \frac{\partial E\left(b^{[x]}\right)}{\partial b^{[x]}}\ , x = 1, 2$$

Forward pass:

$$Hidden\ nodes = \begin{bmatrix}0.1 & 0.1\\0.1 & 0.2\\0.2 & 0.1\end{bmatrix}\begin{bmatrix}1\\1\end{bmatrix} + \begin{bmatrix}0.1\\0\\0.1\end{bmatrix} = \begin{bmatrix}0.3\\0.3\\0.4\end{bmatrix} = h$$

$$Outputs = \begin{bmatrix}1 & 2 & 2\\1 & 2 & 1\\1 & 1 & 1\end{bmatrix}\begin{bmatrix}0.3\\0.3\\0.4\end{bmatrix} + \begin{bmatrix}1\\1\\1\end{bmatrix} = \begin{bmatrix}2.7\\2.3\\2\end{bmatrix} = z$$

$$Outputs\ com\ ativação\ (Softmax) = \begin{bmatrix}\frac{e^{2.7}}{e^{2.7}+e^{2.3}+e^{2}}\\ \frac{e^{2.3}}{e^{2.7}+e^{2.3}+e^{2}}\\ \frac{e^{2}}{e^{2.7}+e^{2.3}+e^{2}}\end{bmatrix} = \begin{bmatrix}0.46149\\0.30934\\0.22917\end{bmatrix} = p$$

$$Cross\ Entropy\ loss = -(0 \cdot \log(0.46149) + 1 \cdot \log(0.30934) + 0 \cdot \log(0.22917)) = 1.17331$$

Backpropagation:

$$Derivada\ Softmax = \frac{\partial p_i}{\partial z_i} = p_i\left(\delta_{ij} - p_j\right), \qquad \delta_{ij} = \begin{cases}1 & , se\ i = j\\0 & , cc\end{cases} (Kronecker\ delta)$$

$$Derivada\ Cross\ Entropy = \frac{\partial Loss}{\partial z_i} = p_i - y_i$$

$$\frac{\partial Loss}{\partial z_i} = \begin{bmatrix} p_1 - y_1 \\ p_2 - y_2 \\ p_3 - y_3 \end{bmatrix} = \begin{bmatrix} 0.46149 - 0 \\ 0.30934 - 1 \\ 0.22917 - 0 \end{bmatrix} = \begin{bmatrix} 0.46149 \\ -0.69066 \\ 0.22917 \end{bmatrix}$$

$$\frac{\partial E\left(W^{[2]}\right)}{\partial W^{[2]}} = \frac{\partial Loss}{\partial z_i} \times h = \begin{bmatrix} 0.46149 \\ -0.69066 \\ 0.22917 \end{bmatrix} \times \begin{bmatrix} 0.3 \\ 0.3 \\ 0.4 \end{bmatrix} = \begin{bmatrix} 0.13845 & 0.13845 & 0.18460 \\ -0.20720 & -0.20720 & -0.27626 \\ 0.06875 & 0.06875 & 0.09167 \end{bmatrix}$$

$$\frac{\partial E\left(b^{[2]}\right)}{\partial b^{[2]}} = \frac{\partial Loss}{\partial z_i} = \begin{bmatrix} 0.46149 \\ -0.69066 \\ 0.22917 \end{bmatrix}$$

$$\frac{\partial E\left(W^{[1]}\right)}{\partial W^{[1]}} = W^{[2]T} \cdot \frac{\partial Loss}{\partial z_i} \times i = \begin{bmatrix} 0.00000 & 0.00000 \\ -0.22917 & -0.22917 \\ 0.46149 & 0.46149 \end{bmatrix}$$

$$\frac{\partial E\left(b^{[1]}\right)}{\partial b^{[1]}} = W^{[2]T} \cdot \frac{\partial Loss}{\partial z_i} = \begin{bmatrix} 0.00000 \\ -0.22917 \\ 0.46149 \end{bmatrix}$$

Resposta:

$$W^{[2]\prime} = W^{[2]} - \eta \frac{\partial E\left(W^{[2]}\right)}{\partial W^{[2]}} = \begin{bmatrix} 1 & 2 & 2 \\ 1 & 2 & 1 \\ 1 & 1 & 1 \end{bmatrix} - 0.1 \begin{bmatrix} 0.13845 & 0.13845 & 0.18460 \\ -0.20720 & -0.20720 & -0.27626 \\ 0.06875 & 0.06875 & 0.09167 \end{bmatrix} =$$

$$= \begin{bmatrix} 0.98616 & 1.98616 & 1.98154 \\ 1.02072 & 2.02072 & 1.02763 \\ 0.99312 & 0.99312 & 0.99083 \end{bmatrix}$$

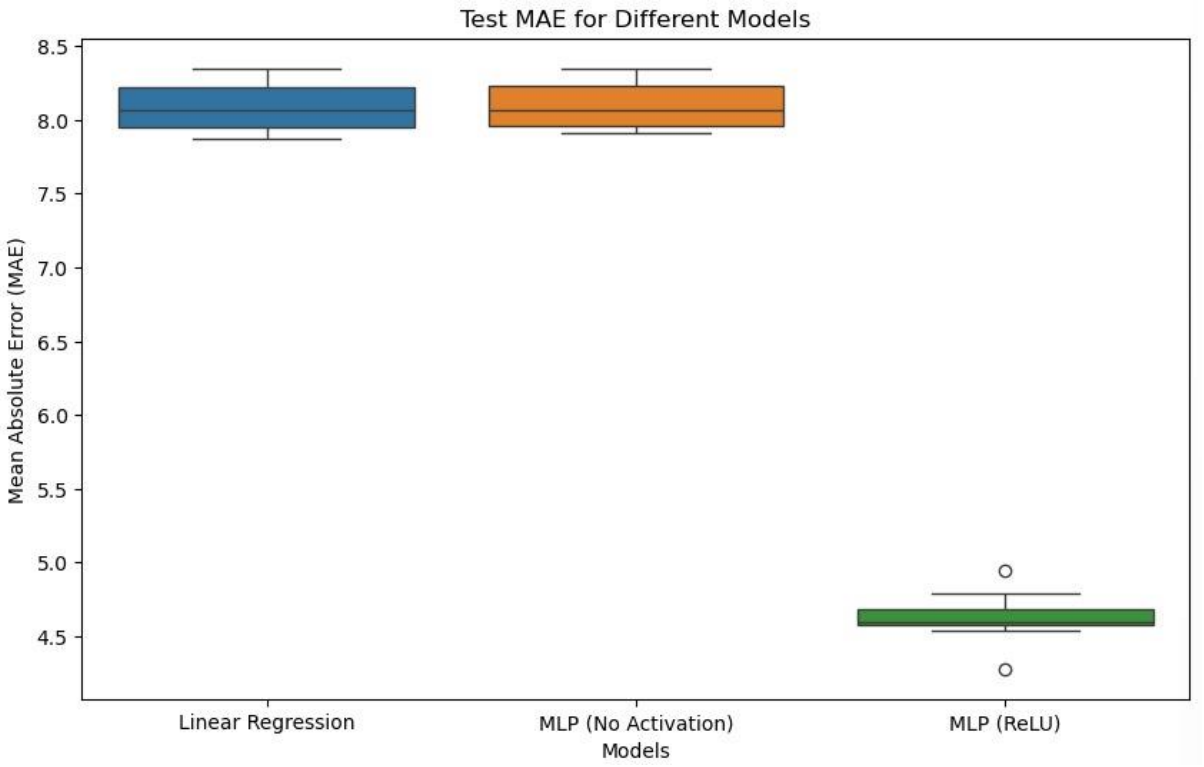
$$b^{[2]\prime} = b^{[2]} - \eta \frac{\partial E\left(b^{[2]}\right)}{\partial b^{[2]}} = \begin{bmatrix} 0.95385 \\ 1.06907 \\ 0.97708 \end{bmatrix}$$

$$W^{[1]\prime} = W^{[1]} - \eta \frac{\partial E\left(W^{[1]}\right)}{\partial W^{[1]}} = \begin{bmatrix} 0.10000 & 0.10000 \\ 0.12292 & 0.22292 \\ 0.15385 & 0.05385 \end{bmatrix}$$

$$b^{[1]\prime} = b^{[1]} - \eta \frac{\partial E\left(b^{[1]}\right)}{\partial b^{[1]}} = \begin{bmatrix} 0.10000 \\ 0.02292 \\ 0.05385 \end{bmatrix}$$

II. Programming and critical analysis

5)



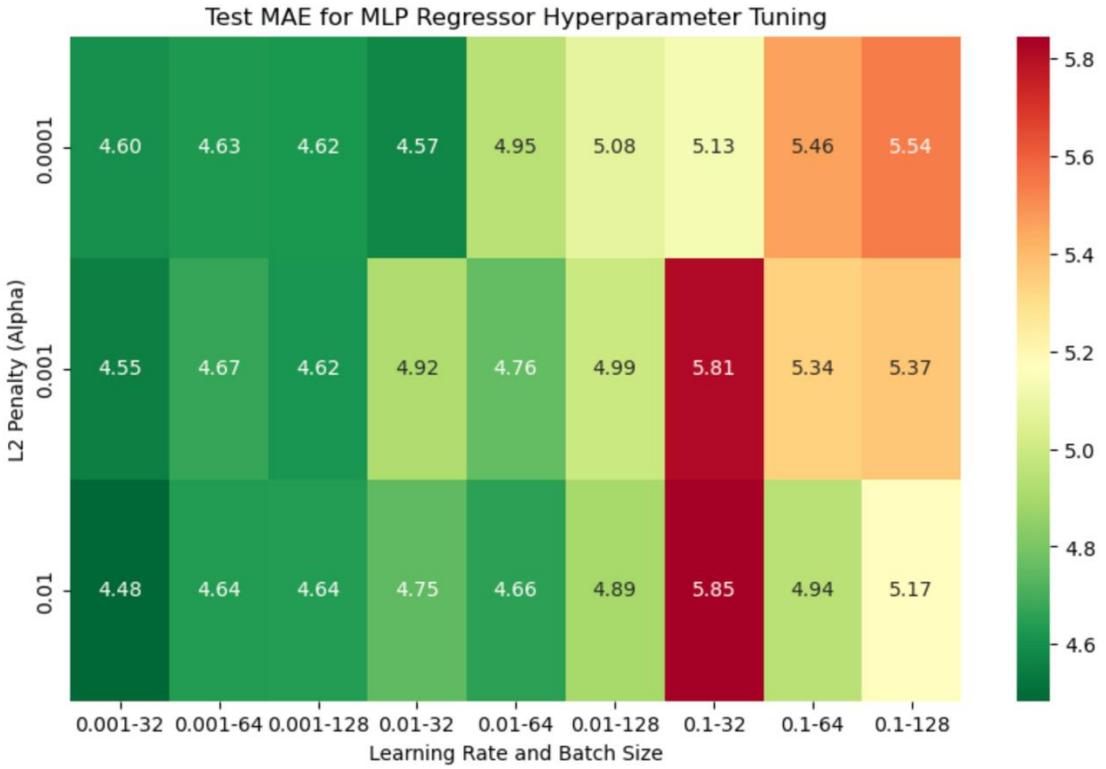
6)

Numa regressão linear, o modelo assume uma relação linear entre as features de entrada e a variável target. Um MLP sem ativações comporta-se essencialmente como um modelo linear, já que cada layer passa a weighted sum sem realizar qualquer transformação. Observando o gráfico, vemos que corrobora a nossa tese, o boxplot MAE da 'Linear Regression' é praticamente igual ao do MLP sem ativações.

Sem funções de ativação, as redes neuronais estariam limitadas a transformações lineares, tornando-as incapazes de aprender e modelar dados complexos. Ao introduzir não-linearidade, as funções de ativação permitem que as redes neuronais aproximem funções arbitrárias, tornando-as ferramentas poderosas para tarefas como o reconhecimento de imagens, o processamento de linguagem natural e muitas outras.

Como podemos ver pelo boxplot, ao usar a função de ativação ReLu, o modelo apresentou um desempenho signficamente melhor, tendo-se ajustado melhor a estes dados.

7)



Melhores parametros**:**

- alpha: 0.01

- batch_size: 32

- learning_rate_init: 0.001

Ao observarmos este mapa de calor, podemos começar por ver que à medida que o learning rate aumenta, o MAE também aumenta. Podemos ver isto fixando as outras duas variáveis (alpha e batch size), concluindo que o melhor valor de learning rate será o mínimo (0.001).

Os valores de alpha podem produzir overfitting (valor baixo) ou underfitting (valor alto), dependendo da combinação.

Ao aumentar o batch size, geralmente é mais estável o treino, levando a valores de MAE mais baixos. Observando o gráfico, vemos que não parece haver problema quando a learning rate é baixa, começando-se a notar para learning rates de 0.01 e 0.1.