Aprendizagem 2024/25

**Homework III – Group 99**

(ist1106059, ist1106329)

# I. Pen-and-paper

| D | $\Phi$ ($y_1$, $y_2$) | $y_{num}$ |
|---|---|---|
| $x_1$ | 1 | 1.25 |
| $x_2$ | 3 | 7.0 |
| $x_3$ | 6 | 2.7 |
| $x_4$ | 9 | 3.2 |
| $x_5$ | 8 | 5.5 |

Nos cálculos seguintes usamos: $z = y_{num}$ e $\Phi$ ($y_1$, $y_2$) = y'

**1)**

OLS closed form:

$$w = (X^T \cdot X)^{-1} \cdot X^T \cdot z = \left( \begin{bmatrix} 1 & 1 & 1 & 1 & 1 \\ 1 & 3 & 6 & 9 & 8 \end{bmatrix} \begin{bmatrix} 1 & 1 \\ 1 & 3 \\ 1 & 6 \\ 1 & 9 \\ 1 & 8 \end{bmatrix} \right)^{-1} \cdot \begin{bmatrix} 1 & 1 & 1 & 1 & 1 \\ 1 & 3 & 6 & 9 & 8 \end{bmatrix} \begin{bmatrix} 1.25 \\ 7.0 \\ 2.7 \\ 3.2 \\ 5.5 \end{bmatrix} = \begin{bmatrix} 3.31593 \\ 0.11372 \end{bmatrix}$$

Modelo de Regressão no espaço transformado:

$\hat{z} = w_0 + y' w_1 = 3.31593 + y' \times 0.11372$

**2)**

Ridge Regression:

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

$$= \left( \begin{bmatrix} 1 & 1 & 1 & 1 & 1 \\ 1 & 3 & 6 & 9 & 8 \end{bmatrix} \begin{bmatrix} 1 & 1 \\ 1 & 3 \\ 1 & 6 \\ 1 & 9 \\ 1 & 8 \end{bmatrix} + 1 \cdot \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \right)^{-1} \cdot \begin{bmatrix} 1 & 1 & 1 & 1 & 1 \\ 1 & 3 & 6 & 9 & 8 \end{bmatrix} \begin{bmatrix} 1.25 \\ 7.0 \\ 2.7 \\ 3.2 \\ 5.5 \end{bmatrix}$$

$$= \begin{bmatrix} 1.81809 \\ 0.32376 \end{bmatrix}$$

$\hat{z} = w_0 + y' w_1 = 1.81809 + y' \times 0.32376$

Para comparar os coeficientes aprendidos dos dois modelos anteriores, temos de comparar a norma dos coeficientes de cada um:

- Para o Modelo de Regressão com OLS:

  $\|w\| = \sqrt{3.31593^2 + 0.11372^2} = 3.31788$

- Para o Modelo de Regressão de Ridge:

  $\|w\| = \sqrt{1.81809^2 + 0.32376^2} = 1.84669$

Devido à natureza da regularização de Ridge ($E(w) = SSE(w) + \frac{\lambda}{2}\|w\|^2$) e observando os resultados anteriores, a norma no Modelo de Regressão de Ridge é menor, já que esta regularização faz com que haja uma penalização no custo de coeficientes maiores, ou seja, reduz a sensibilidade a ruído nos dados. Logo, com a regressão de Ridge obtemos um modelo com maior capacidade de generalização e um menor risco de overfitting.

**3)**

| D | $y_1$ | $y_2$ | $\Phi$ ($y_1$, $y_2$) | $y_{num}$ |
|---|---|---|---|---|
| $x_1$ | 1 | 1 | 1 | 1.25 |
| $x_2$ | 1 | 3 | 3 | 7.0 |
| $x_3$ | 3 | 2 | 6 | 2.7 |
| $x_4$ | 3 | 3 | 9 | 3.2 |
| $x_5$ | 2 | 4 | 8 | 5.5 |
| $x_6$ | 2 | 2 | 4 | 0.7 |
| $x_7$ | 1 | 2 | 2 | 1.1 |
| $x_8$ | 5 | 1 | 5 | 2.2 |

Nos cálculos seguintes usamos: $z = y_{num}$ e $\Phi$ (y1, y2) = y'

$$RMSE(\hat{z}, z) = \sqrt{\frac{1}{n} \times \sum_{i=1}^{n} (z_i - \hat{z}_i)^2}$$

Aprendizagem 2024/25

**Homework III – Group 99**

(ist1106059, ist1106329)

Para o Modelo de Regressão com OLS:

- Train RMSE:

$$\hat{z} = X \cdot w = \begin{bmatrix} 1 & 1 \\ 1 & 3 \\ 1 & 6 \\ 1 & 9 \\ 1 & 8 \end{bmatrix} \begin{bmatrix} 3.31593 \\ 0.11372 \end{bmatrix} = \begin{bmatrix} 3.42965 \\ 3.65709 \\ 3.99825 \\ 4.33941 \\ 4.22569 \end{bmatrix}$$

$$RMSE_{train}(\hat{z}, z) = \sqrt{\frac{(1.25-3.42965)^2+(7.0-3.65709)^2+(2.7-3.99825)^2+(3.2-4.33941)^2+(5.5-4.22569)^2}{5}} = 2.02649$$

- Test RMSE:

$$\hat{z} = X \cdot w = \begin{bmatrix} 1 & 4 \\ 1 & 2 \\ 1 & 5 \end{bmatrix} \begin{bmatrix} 3.31593 \\ 0.11372 \end{bmatrix} = \begin{bmatrix} 3.11313 \\ 2.46561 \\ 3.43689 \end{bmatrix}$$

$$RMSE_{test}(\hat{z}, z) = \sqrt{\frac{(0.7 - 3.11313)^2 + (1.1 - 2.46561)^2 + (2.2 - 3.43689)^2}{3}} = 2.46560$$

Para o Modelo de Regressão de Ridge:

- Train RMSE:

$$\hat{z} = X \cdot w = \begin{bmatrix} 1 & 1 \\ 1 & 3 \\ 1 & 6 \\ 1 & 9 \\ 1 & 8 \end{bmatrix} \begin{bmatrix} 1.81809 \\ 0.32376 \end{bmatrix} = \begin{bmatrix} 2.14185 \\ 2.78937 \\ 3.76065 \\ 4.73193 \\ 4.40817 \end{bmatrix}$$

$$RMSE_{train}(\hat{z}, z) = \sqrt{\frac{(1.25-2.14185)^2+(7.0-2.78937)^2+(2.7-3.76065)^2+(3.2-4.73193)^2+(5.5-4.40817)^2}{5}} = 2.15354$$

- Test RMSE:

$$\hat{z} = X \cdot w = \begin{bmatrix} 1 & 4 \\ 1 & 2 \\ 1 & 5 \end{bmatrix} \begin{bmatrix} 1.81809 \\ 0.32376 \end{bmatrix} = \begin{bmatrix} 3.11313 \\ 2.46561 \\ 3.43689 \end{bmatrix}$$

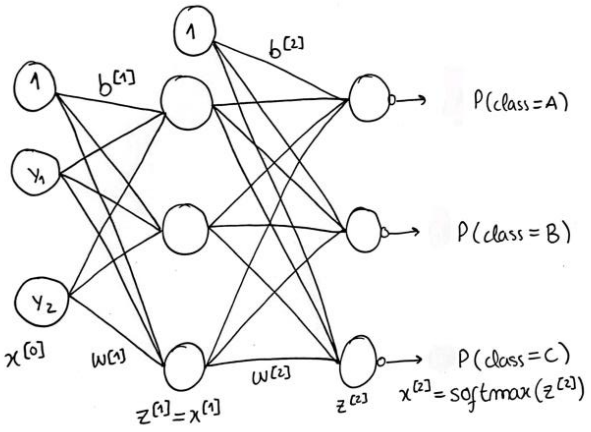
$$RMSE_{test}(\hat{z}, z) = \sqrt{\frac{(0.7 - 3.11313)^2 + (1.1 - 2.46561)^2 + (2.2 - 3.43689)^2}{3}} = 1.75290$$

Os resultados apresentados acima estão de acordo com o esperado.

Primeiramente, faz todo o sentido o modelo de Regressão de Ridge, por ter a regularização mencionada na pergunta anterior, ter um RMSE menor para os dados de teste do que o outro modelo, visto que, como tem uma maior capacidade de generalização e, por consequência, um risco de overfitting menor, obtém melhores resultados para a previsão de novas observações.

Por outro lado, este mesmo modelo apresenta um RMSE maior no que toca aos dados de treino em comparação com o modelo de Regressão com OLS. Este resultado continua alinhado com o esperado já que, precisamente por evitar overfitting, é menos sensível a certos dados de treino e, por isso, tem uma performance pior que o outro modelo que é muito mais ajustado aos mesmos.

**4)** Observando as dimensões dos dados presentes no enunciado, conseguimos desenhar o seguinte MLP:




Fazendo um Stochastic Gradient Descent update dos pesos e biases usando a observação $x_1$, temos:

<u>Forward Propagation:</u>

$$X^{[0]} = \begin{bmatrix}1\\1\end{bmatrix}$$

$$Z^{[1]} = W^{[1]} \cdot X^{[0]} + b^{[1]} = \begin{bmatrix}0.1 & 0.1\\0.1 & 0.2\\0.2 & 0.1\end{bmatrix}\begin{bmatrix}1\\1\end{bmatrix} + \begin{bmatrix}0.1\\0\\0.1\end{bmatrix} = \begin{bmatrix}0.3\\0.3\\0.4\end{bmatrix}$$

Como estamos numa hidden layer,

$$X^{[1]} = Z^{[1]}$$

**Homework III – Group 99**

(ist1106059, ist1106329)

$$Z^{[2]} = W^{[2]} \cdot X^{[1]} + b^{[2]} = \begin{bmatrix} 1 & 2 & 2 \\ 1 & 2 & 1 \\ 2 & 1 & 1 \end{bmatrix} \begin{bmatrix} 0.3 \\ 0.3 \\ 0.4 \end{bmatrix} + \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} = \begin{bmatrix} 2.7 \\ 2.3 \\ 2.0 \end{bmatrix}$$

Como estamos na output layer,

$$X^{[2]} = softmax\left(Z^{[2]}\right) = \begin{bmatrix} 0.46149 \\ 0.30934 \\ 0.22917 \end{bmatrix}$$

Backwards Propagation:

$$W_{new}^{[n]} = W^{[n]} - \eta \frac{\partial E(W)}{\partial W^{[n]}}$$

$$\frac{\partial E(W)}{\partial W^{[n]}} = \delta^{[n]} \cdot \left(\frac{\partial Z^{[n]}}{\partial W^{[n]}}\right)^T = \delta^{[n]} \cdot \left(X^{[n-1]}\right)^T$$

$$b_{new}^{[n]} = b^{[n]} - \eta \frac{\partial E(W)}{\partial b^{[n]}} = b^{[n]} - \eta \delta^{[n]}$$

Para a layer de output, como estamos a usar o cross-entropy error e temos uma função de ativação softmax, $\delta^{[2]} = X^{[2]} - t$, sendo t o vetor dos outputs reais, ou seja:

$$\delta^{[2]} = \begin{bmatrix} 0.46149 \\ 0.30934 \\ 0.22917 \end{bmatrix} - \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix} = \begin{bmatrix} 0.46149 \\ -0.69066 \\ 0.22917 \end{bmatrix}$$

$$\frac{\partial E(W)}{\partial W^{[2]}} = \delta^{[2]} \cdot \left(X^{[1]}\right)^T = \begin{bmatrix} 0.46149 \\ -0.6907 \\ 0.22917 \end{bmatrix} \cdot \left(\begin{bmatrix} 0.3 \\ 0.3 \\ 0.4 \end{bmatrix}\right)^T = \begin{bmatrix} 0.13845 & 0.13846 & 0.18460 \\ -0.20720 & -0.20720 & -0.27626 \\ 0.06875 & 0.06875 & 0.09167 \end{bmatrix}$$

$$W_{new}^{[2]} = W^{[2]} - \eta \frac{\partial E(W)}{\partial W^{[2]}} = \begin{bmatrix} 1 & 2 & 2 \\ 1 & 2 & 1 \\ 2 & 1 & 1 \end{bmatrix} - 0.1 \begin{bmatrix} 0.13845 & 0.13845 & 0.18460 \\ -0.20720 & -0.20720 & -0.27626 \\ 0.06875 & 0.06875 & 0.09167 \end{bmatrix} =$$
$$= \begin{bmatrix} 0.98616 & 1.98616 & 1.98154 \\ 1.02072 & 2.02072 & 1.02763 \\ 0.99312 & 0.99312 & 0.99083 \end{bmatrix}$$

$$b_{new}^{[2]} = b^{[2]} - \eta \delta^{[2]} = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} - 0.1 \begin{bmatrix} 0.46149 \\ -0.69066 \\ 0.22917 \end{bmatrix} = \begin{bmatrix} 0.95385 \\ 1.06907 \\ 0.97708 \end{bmatrix}$$

**Homework III – Group 99**

(ist1106059, ist1106329)

Para a hidden layer, como não temos função de ativação, $\frac{\partial X^{[n]}}{\partial Z^{[n]}} = 1$, e por isso

$$\delta^{[n]} = \left(W^{[n+1]}\right)^T \cdot \delta^{[n+1]} \circ \frac{\partial X^{[n]}}{\partial Z^{[n]}} = \left(W^{[n+1]}\right)^T \cdot \delta^{[n+1]}$$

$$\delta^{[1]} = \left(W^{[2]}\right)^T \cdot \delta^{[2]} = \left(\begin{bmatrix}1 & 2 & 2\\ 1 & 2 & 1\\ 2 & 1 & 1\end{bmatrix}\right)^T \cdot \begin{bmatrix}0.46149\\ -0.69066\\ 0.22917\end{bmatrix} = \begin{bmatrix}-1.0\times10^{-8}\\ -0.22917\\ 0.46149\end{bmatrix}$$

$$\frac{\partial E(W)}{\partial W^{[1]}} = \delta^{[1]} \cdot \left(X^{[0]}\right)^T = \begin{bmatrix}-1.0\times10^{-8}\\ -0.22917\\ 0.46149\end{bmatrix} \cdot \left(\begin{bmatrix}1\\1\end{bmatrix}\right)^T = \begin{bmatrix}-1.0\times10^{-8} & -1.0\times10^{-8}\\ -0.22917 & -0.22917\\ 0.46149 & 0.46149\end{bmatrix}$$

$$W_{new}^{[1]} = W^{[1]} - \eta\frac{\partial E(W)}{\partial W^{[1]}} = \begin{bmatrix}0.1 & 0.1\\ 0.1 & 0.2\\ 0.2 & 0.1\end{bmatrix} - 0.1\begin{bmatrix}-1.0\times10^{-8} & -1.0\times10^{-8}\\ -0.22917 & -0.22917\\ 0.46149 & 0.46149\end{bmatrix} =$$
$$= \begin{bmatrix}0.1 & 0.1\\ 0.12292 & 0.22292\\ 0.15385 & 0.05385\end{bmatrix}$$

$$b_{new}^{[1]} = b^{[1]} - \eta\delta^{[1]} = \begin{bmatrix}0.1\\ 0\\ 0.1\end{bmatrix} - 0.1\begin{bmatrix}-1.0\times10^{-8}\\ -0.22917\\ 0.46149\end{bmatrix} = \begin{bmatrix}0.1\\ 0.02292\\ 0.05385\end{bmatrix}$$

Concluindo, após o Stochastic update, obtemos os seguintes pesos e biases atualizados:

$$W^{[2]} = \begin{bmatrix}0.98616 & 1.98616 & 1.98154\\ 1.02072 & 2.02072 & 1.02763\\ 0.99312 & 0.99312 & 0.99083\end{bmatrix} \qquad W^{[1]} = \begin{bmatrix}0.1 & 0.1\\ 0.12292 & 0.22292\\ 0.15385 & 0.05385\end{bmatrix}$$

$$b^{[2]} = \begin{bmatrix}0.95385\\ 1.06907\\ 0.97708\end{bmatrix} \qquad b^{[1]} = \begin{bmatrix}0.1\\ 0.02292\\ 0.05385\end{bmatrix}$$

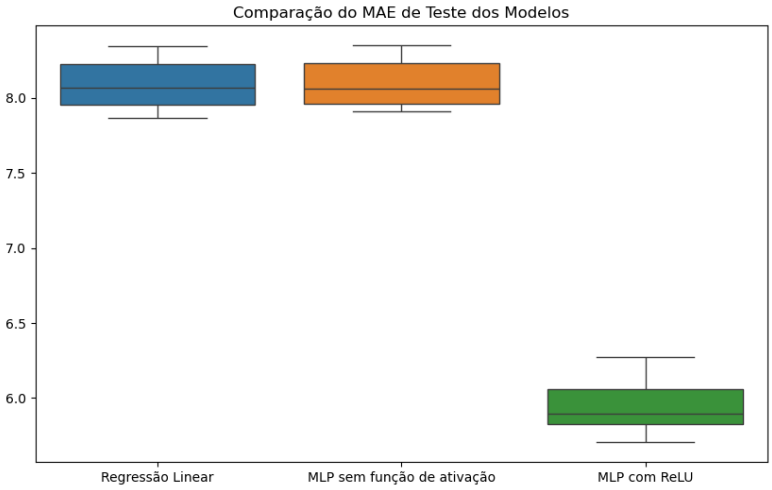
Aprendizagem 2024/25

**Homework III – Group 99**

(ist1106059, ist1106329)

## II. Programming and critical analysis

**5)** Nesta questão gerámos um boxplot que compara o erro absoluto médio (MAE) dos três modelos de regressão pedidos ao longo de 10 iterações com diferentes amostras de treino e teste.




**6)** Ao analisar os boxplots da pergunta anterior, podemos tirar 2 conclusões.

Em primeiro lugar, a performance do modelo de Regressão Linear é quase idêntica à do MLP sem função de ativação. Isto acontece porque, sem a função de ativação, o valor de output dos neurónios de cada camada não passa de uma combinação linear dos pesos com os inputs e, por isso, não consegue captar relações complexas e não lineares nos dados.

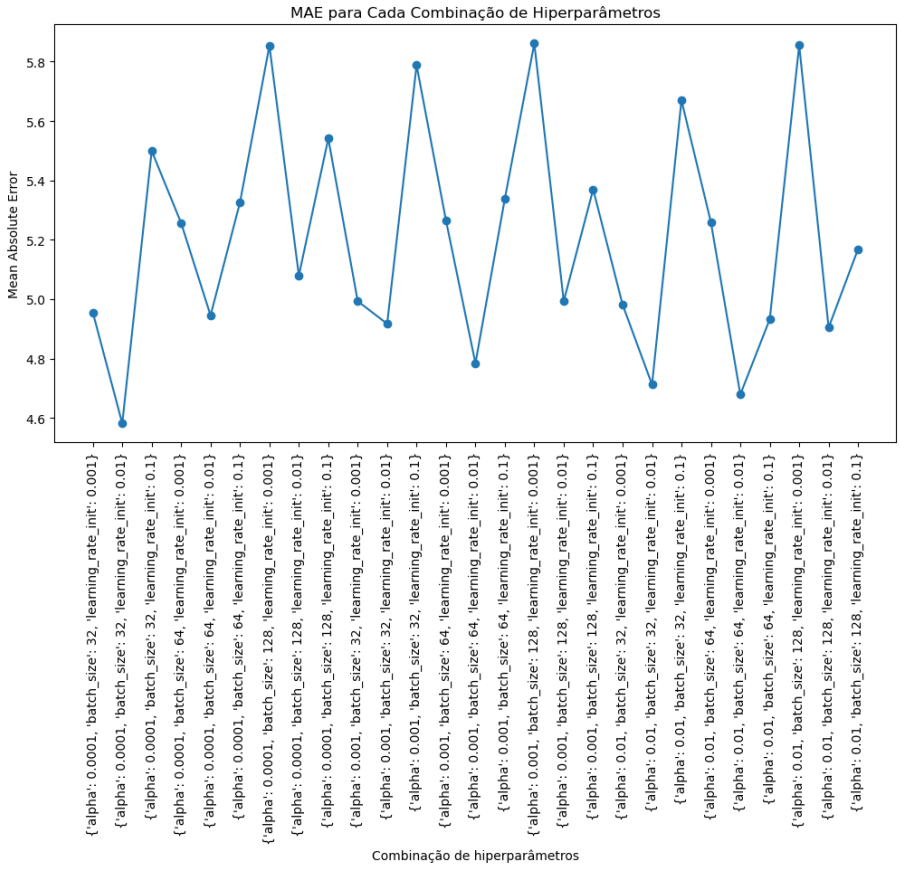
Em contraste com isso, o MLP com a função de ativação ReLu já consegue captar estes padrões não lineares e, por isso, apresenta um desempenho significativamente melhor e mais estável que os outros dois modelos (que é equivalente a apresentar um MAE menor), como pode ser observado no boxplot.

**7)** O gráfico abaixo ilustra o desempenho do MLP regressor (em termos de MAE) para cada uma das combinações de hiperparâmetros testadas na Grid Search. Cada ponto no gráfico representa uma combinação única de valores para o L2 penalty (alpha), learning rate (learning_rate_init) e batch size. O MAE (Mean Absolute Error) é representado no eixo vertical, com valores mais baixos indicando melhor desempenho

Aprendizagem 2024/25

**Homework III – Group 99**

(ist1106059, ist1106329)




A melhor combinação para este MLP regressor é:
L2 penalty (alpha) = 0.0001;
Learning rate = 0.01;
Batch size = 32.

Para começar, o L2 penalty é um termo de regularização da função de custo para penalizar coeficientes com valores maiores, de forma a tornar o modelo mais ou menos simples e ajustar a sua generalização. Quanto menor for este valor, menos regularizado é o modelo, logo, irá apresentar um maior risco de overfitting e o contrário pode levar a um aumento no risco de underfitting.

Quanto ao learning rate, este é um parâmetro que controla o tamanho dos passos/ajustes que o modelo dá aos seus coeficientes (pesos e biases) durante o treino, baseando-se no gradiente da função de custo. Este valor deve ser equilibrado uma vez que learning rates demasiado elevadas levam a um modelo instável com um maior risco de evitar/saltar o mínimo global, já que as atualizações dos coeficientes são maiores. Por outro lado, learning rates muito pequenas aumentam o número de atualizações necessárias para encontrar este ponto ótimo dos coeficientes.

Por fim, o batch size é um parâmetro que evita fazer o cálculo do gradiente da função de custo para todos os dados, dando-nos o número de observações usadas de cada vez para fazer a atualização dos coeficientes. Quanto menor for este número, mais rápida e "barata" é cada atualização de computar, mas mais frequente são as atualizações e mais sensível é a ruídos nos dados. Isto faz com que a função de custo tenha mais oscilações e demore mais tempo a convergir para um mínimo global (ainda que,

por explorar mais o espaço de soluções, seja melhor a escapar de mínimos locais do que se o batch size for maior). Valores maiores de batch size levam a consequências opostas.

Estes valores e trade-offs dependem de modelo para modelo e do objetivo e natureza dos dados. Ao observar o gráfico acima e a melhor combinação de parâmetros obtida para este MLP em concreto, conseguimos concluir que:

- o melhor valor de L2 penalty é o menor (0.0001) dos disponibilizados, indicando que valores maiores aumentam o underfitting do modelo e, por consequência, diminuem o seu desempenho.
- o melhor valor da learning rate é 0.01 que corresponde ao valor intermédio, ou seja, um valor equilibrado para o tamanho dos saltos das atualizações dos coeficientes, em oposição aos outros valores que apresentam maior risco de saltar o mínimo global ou de nem sequer encontrar este ponto.
- o melhor valor para o batch size é 32, o menor, dando a entender que, para este modelo e dados, valores maiores têm maior risco de ficar presos em mínimos locais e por isso terem um desempenho menor.

END