LEIC-T 2023/2024
Aprendizagem - Machine Learning

Homework III - Group 007

Miguel Teixeira - 103449

Rodrigo Alves - 103299

I

a)

$x_1 = -0{,}8$ $\qquad$ $T_1 = -20$

$x_2 = 1$ $\qquad$ $T_2 = 20$

$x_3 = -1{,}2$ $\qquad$ $T_3 = 10$

$x_4 = 1{,}4$ $\qquad$ $T_4 = 13$

$x_5 = 1{,}9$ $\qquad$ $T_5 = 12$

$\Phi_j(x) = x^j$

$$\Phi = \begin{bmatrix} 1 & x_1 & x_1^2 & x_1^3 \\ 1 & x_2 & x_2^2 & x_2^3 \\ 1 & x_3 & x_3^2 & x_3^3 \\ 1 & x_4 & x_4^2 & x_4^3 \\ 1 & x_5 & x_5^2 & x_5^3 \end{bmatrix} = \begin{bmatrix} 1 & -0{,}8 & 0{,}64 & -0{,}512 \\ 1 & 1 & 1 & 1 \\ 1 & -1{,}2 & 1{,}44 & -1{,}728 \\ 1 & 1{,}4 & 1{,}96 & 2{,}744 \\ 1 & 1{,}9 & 3{,}61 & 6{,}859 \end{bmatrix}$$

b)

$$y = \begin{pmatrix} -20 \\ 20 \\ -10 \\ 13 \\ 12 \end{pmatrix}$$

$$w = \boxed{(X^T X)^{-1} X^T y} = \left(\begin{bmatrix} 1 & 1 & 1 & 1 & 1 \\ -0{,}8 & \cancel{1} & -1{,}2 & 1{,}4 & 1{,}9 \\ 0{,}64 & 1 & 1{,}44 & 1{,}96 & 3{,}61 \\ -0{,}512 & 1 & -1{,}728 & 2{,}744 & 6{,}859 \end{bmatrix} \cdot \begin{bmatrix} 1 & -0{,}8 & 0{,}64 & -0{,}512 \\ 1 & 1 & 1 & 1 \\ 1 & -1{,}2 & 1{,}44 & -1{,}728 \\ 1 & 1{,}4 & 1{,}96 & 2{,}744 \\ 1 & 1{,}9 & 3{,}61 & 6{,}859 \end{bmatrix} \right)^{-1} \times$$

$$\times \begin{bmatrix} 1 & 1 & 1 & 1 & 1 \\ -0{,}8 & 1 & -1{,}2 & 1{,}4 & 1{,}9 \\ 0{,}64 & 1 & 1{,}44 & 1{,}96 & 3{,}61 \\ -0{,}512 & 1 & -1{,}728 & 2{,}744 & 6{,}859 \end{bmatrix} \times \begin{bmatrix} -20 \\ 20 \\ -10 \\ 13 \\ 12 \end{bmatrix} = \begin{bmatrix} 5 & 2{,}3 & 8{,}65 & 8{,}363 \\ 2{,}3 & 8{,}65 & 8{,}363 & 20{,}36 \\ 8{,}65 & 8{,}3663 & 20{,}35 & 28{,}32 \\ 8{,}363 & 20{,}36 & 28{,}73 & 52{,}37 \end{bmatrix}^{-1} \times \begin{bmatrix} 1 & 1 & 1 & 1 & \dots \\ -0{,}8 & 1 & -1{,}2 & 1 & \dots \\ 0{,}64 & 1 & \dots \\ -0{,}512 & 1 & \dots \end{bmatrix}$$

$$\times \begin{bmatrix} -20 \\ 20 \\ -10 \\ 13 \\ 12 \end{bmatrix} = \cancel{\begin{bmatrix} 0{,}55 & 0{,}59 & -0{,}096 & -0{,}275 \\ 0{,}59 & -0{,}057 & -0{,}55 & 0{,}234 \\ -0{,}096 & -0{,}58 & -0{,}098 & 0{,}24 \\ -0{,}275 & 0{,}234 & 0{,}24 & -0{,}18 \end{bmatrix} \times \begin{bmatrix} 1 & 1 & 1 & 1 & 1 \\ -0{,}8 & 1 & -1{,}2 & 1{,}4 & 1{,}9 \\ 0{,}64 & 1 & 1{,}44 & 1{,}96 & 3{,}6 \\ -0{,}512 & 1 & -1{,}728 & 2{,}744 & 6{,}9 \end{bmatrix}}$$

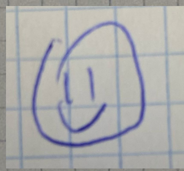
6) cont'd

$$= \begin{bmatrix} 2 & -0{,}6 & -1{,}5 & 0{,}64 \\ -0{,}6 & 0{,}96 & 0{,}02 & -0{,}54 \\ -1{,}5 & 0{,}02 & 1{,}3 & -0{,}63 \\ 0{,}64 & -0{,}54 & -0{,}63 & 0{,}42 \end{bmatrix} \times \begin{bmatrix} 1 & 1 & 1 & 1 & 1 \\ -0{,}8 & 1 & -1{,}2 & 1{,}4 & 1{,}9 \\ 0{,}64 & 1 & 1{,}44 & 1{,}96 & 3{,}61 \\ 0{,}512 & 1 & -1{,}728 & 2{,}749 & 6{,}059 \end{bmatrix} \times \begin{bmatrix} -20 \\ 20 \\ -10 \\ 13 \\ 12 \end{bmatrix} =$$

$$= \begin{bmatrix} 1{,}2 & 0{,}53 & -0{,}55 & -0{,}014 & -0{,}12 \\ -0{,}09 & 0{,}02 & 0{,}021 & 0{,}46 & -0{,}27 \\ -0{,}32 & -0{,}19 & 0{,}14 & 0{,}2 & 0{,}011 \\ 0{,}46 & -0{,}11 & -0{,}33 & -0{,}021 & 0{,}2 \end{bmatrix} \times \begin{bmatrix} -20 \\ 20 \\ -10 \\ 13 \\ 12 \end{bmatrix} = \begin{bmatrix} -9 \\ 24 \\ 3{,}7 \\ -3{,}5 \end{bmatrix}$$

c) $\|w\|_1$ has no derivative, so $\|w\|_1$ has no closed form solution

$$\text{sign}(w_1) = \begin{cases} -1 & \text{if } w_1 < 0 \\ 0 & \text{if } w_1 = 0 \\ 1 & \text{if } w_1 > 0 \end{cases}$$



2

$$W^{(1)} = \begin{pmatrix} 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \end{pmatrix} \qquad b^{(1)} = \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix}$$

$$W^{(2)} = \begin{pmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \end{pmatrix} \qquad b^{(2)} = \begin{pmatrix} 0 \\ 0 \end{pmatrix}$$

$x = (1,1,1,1,1)^T$

$t = (1,0)^T$

$$z^{[1]} = \begin{pmatrix} 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \end{pmatrix} \cdot \begin{pmatrix} 1 \\ 1 \\ 1 \\ 1 \\ 1 \end{pmatrix} + \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix} = \begin{pmatrix} 5 \\ 5 \\ 5 \end{pmatrix}$$

$$x^{[1]} = \max\left(\begin{pmatrix} 5 \\ 5 \\ 5 \end{pmatrix}\right) = \begin{pmatrix} 5 \\ 5 \\ 5 \end{pmatrix}$$

$$z^{[2]} = \begin{pmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \end{pmatrix} \cdot \begin{pmatrix} 5 \\ 5 \\ 5 \end{pmatrix} + \begin{pmatrix} 0 \\ 0 \end{pmatrix} = \begin{pmatrix} 15 \\ 15 \end{pmatrix}$$

$$x^{[2]} = \mathrm{softmax}\left(\begin{pmatrix} 15 \\ 15 \end{pmatrix}\right) = \begin{cases} \dfrac{\exp(15)}{2\exp(15)} \\ \dfrac{\exp(15)}{2\exp(15)} \end{cases} = \begin{cases} \dfrac{1}{2} \\ \dfrac{1}{2} \end{cases}$$

$$\delta^{[2]} = \frac{\partial E}{\partial x^{[2]}} = \left(x^{[2]} - t\right) = \left(\begin{pmatrix} 0{,}5 \\ 0{,}5 \end{pmatrix} - \begin{pmatrix} 1 \\ 0 \end{pmatrix}\right) = \begin{pmatrix} -0{,}5 \\ 0{,}5 \end{pmatrix}$$

$$\delta^{[1]} = (W^{2})^{T} \cdot \delta^{[2]} \cdot \mathrm{sgn}(x^{1}) = \begin{pmatrix} 1 & 1 \\ 1 & 1 \\ 1 & 1 \end{pmatrix} \cdot \begin{pmatrix} -0{,}5 \\ 0{,}5 \end{pmatrix} \odot \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix} =$$

$$= \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix} \odot \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix}$$

$$\frac{\partial E}{\partial W^{(1)}} = \delta^{[1]} \cdot \left(a^{[0]}\right)^T =$$

$$= \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix} \cdot \begin{pmatrix} 1, 1, 1, 1, 1 \end{pmatrix} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

$$W^{(1)} = W^{(1)} - \eta \frac{\partial E}{\partial W^{(1)}} = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \end{bmatrix} - 0{,}1 \begin{pmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

$$= \begin{bmatrix} 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \end{bmatrix}$$

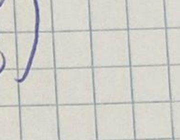
$$\frac{\partial E}{\partial b^{1}} = \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix}$$

$$b^{1} = b^{1} - \eta \frac{\partial E}{\partial b^{(1)}} = \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix} - 0{,}1 \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix}$$

$$\frac{\partial E}{\partial W^{2}} = \begin{pmatrix} -0{,}5 \\ 0{,}5 \end{pmatrix} \cdot \left(a^{(1)^T}\right) = \begin{pmatrix} -0{,}5 \\ 0{,}5 \end{pmatrix} \cdot (5, 5, 5) = \begin{pmatrix} -2{,}5 & -2{,}5 & -2{,}5 \\ 2{,}5 & 2{,}5 & 2{,}5 \end{pmatrix}$$

$$W^{(2)} = W^{(2)} - 0{,}1 \frac{\partial E}{\partial W^{(1)}} = \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix} - 0{,}1 \begin{bmatrix} -2{,}5 & -2{,}5 & -2{,}5 \\ 2{,}5 & 2{,}5 & 2{,}5 \end{bmatrix} =$$

$$= \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix} + \begin{pmatrix} 0{,}25 & 0{,}25 & 0{,}25 \\ -0{,}25 & -0{,}25 & -0{,}25 \end{pmatrix} = \begin{bmatrix} 1{,}25 & 1{,}25 & 1{,}25 \\ 0{,}75 & 0{,}75 & 0{,}75 \end{bmatrix}$$

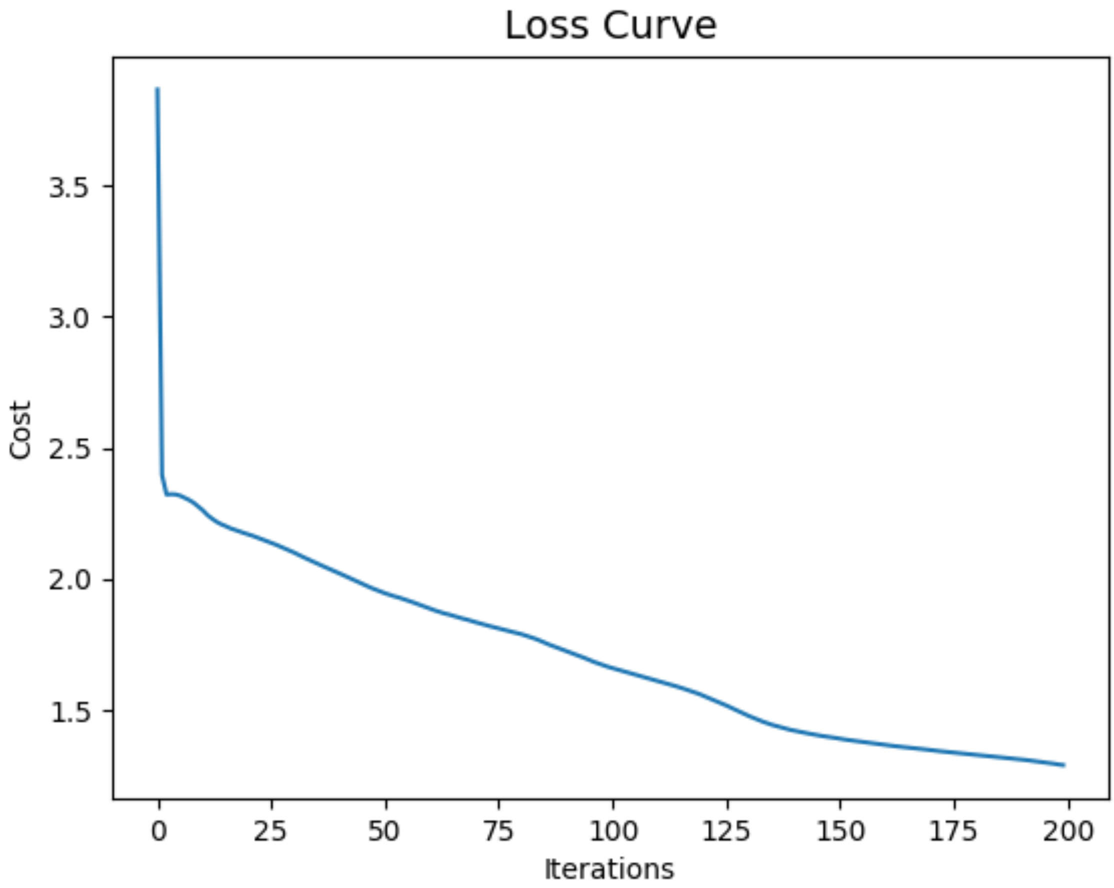


$$\frac{\partial E}{\partial b^{1}} = \begin{pmatrix} -0{,}5 \\ 0{,}5 \end{pmatrix}$$

$$b^{(2)} = b^{(2)} - \eta \frac{\partial E}{\partial b^{(2)}} = \begin{pmatrix} 0 \\ 0 \end{pmatrix} - 0{,}1 \begin{pmatrix} -0{,}5 \\ 0{,}5 \end{pmatrix} = \begin{pmatrix} 0{,}05 \\ -0{,}05 \end{pmatrix}$$

III Software Experiments

```python
import matplotlib.pyplot as plt
from sklearn import metrics, datasets
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LogisticRegression
from sklearn.neural_network import MLPClassifier

# Load the dataset and split it
dt = datasets.load_digits()
X, y = dt.data, dt.target

X_train, X_test, y_train, y_test = train_test_split(X, y, train_size=0.7, stratify=y, random_state=7)

# Train and evaluate the Logistic Regression model
logistic_regression = LogisticRegression(max_iter=10000)
logistic_regression.fit(X_train, y_train)

y_pred_lr = logistic_regression.predict(X_test)
accuracy_lr = round(metrics.accuracy_score(y_test, y_pred_lr), 2)
print("Accuracy on testing set (Logistic Regression):", accuracy_lr)

# Train and evaluate the MLPClassifier with the initial hidden_layer_sizes=(10,4) configuration
mlp_classifier = MLPClassifier(hidden_layer_sizes=(10, 4), random_state=7, activation='relu', solver='sgd')
mlp_classifier.fit(X_train, y_train)

y_pred_mlp = mlp_classifier.predict(X_test)
accuracy_mlp = round(metrics.accuracy_score(y_test, y_pred_mlp), 4)
print("Accuracy on testing set (MLP with (10,4) hidden layers):", accuracy_mlp)

# Plot the loss curve for the MLPClassifier
plt.plot(mlp_classifier.loss_curve_)
plt.title("Loss Curve", fontsize=14)
plt.xlabel('Iterations')
plt.ylabel('Cost')
plt.show()

# Experiment with different hidden_layer_sizes configurations
hidden_layer_sizes_list = [(10,), (10, 4), (20, 10), (30, 20), (50, 30)]

best_accuracy = 0
best_hidden_layers = None

for hidden_layers in hidden_layer_sizes_list:
    mlp_classifier = MLPClassifier(hidden_layer_sizes=hidden_layers, random_state=7, activation='relu', solver='
    mlp_classifier.fit(X_train, y_train)

    y_pred_mlp = mlp_classifier.predict(X_test)
    accuracy_mlp = metrics.accuracy_score(y_test, y_pred_mlp)

    if accuracy_mlp > best_accuracy:
        best_accuracy = accuracy_mlp
        best_hidden_layers = hidden_layers

print("Best hidden_layer_sizes:", best_hidden_layers)
print("Best accuracy on testing set:", round(best_accuracy, 4))
```



```
Accuracy on testing set (Logistic Regression): 0.97
Accuracy on testing set (MLP with (10,4) hidden layers): 0.4593
Best hidden_layer_sizes: (50, 30)
Best accuracy on testing set: 0.963
```

By experimenting with different hidden_layer_sizes configurations, the best-performing configuration was found to be (50, 30) for the hidden layers, resulting in the highest accuracy on the test set. The corresponding loss curve showed that this configuration achieved better convergence during training, indicating its effectiveness for this dataset.