Aprendizagem 2021/22

Homework I – Group 047

I. Pen-and-paper

1)

a)

Activation Function: $tanh(x)$ **Loss Function:** $E = \frac{1}{n}\sum_{i=1}^{n}\left(T_i - X_i^{[m]}\right)^2, m = last\ layer\ of\ the\ MLP$

Forward Propagation: $Z^{[n]} = W^{[n]} \times X^{[n-1]} + B^{[n]}$ $X^{[n]} = \tanh\left(Z^{[n]}\right)$

Input Layer							Hidden Layer 1								Hidden Layer 2						Output Layer		
X[0]	B[1]	W[1]					Z[1]	X[1]	δ[1]	B[2]	W[2]			Z[2]	X[2]	δ[2]	B[3]	W[3]		Z[3]	X[3]	δ[3]	TARGET
1	1	1	1	1	1	1	6	0,999988		1	1	1	1	3,76157	0,99892		0	0	0	0	0		1
1	1	0	0	0	0	0	1	0,761594		1	1	1	1	3,76157	0,99892		0	0	0	0	0		-1
1	1	1	1	1	1	1	6	0,999988															
1																							
1																							

$$Error = \frac{1}{2}\sum_{i=1}^{2}\left(T_i - X_i^{[3]}\right)^2 = 1$$

Backward Propagation:

1. Deltas:

$$\delta^{[3]} = \frac{\partial E}{\partial Z^{[3]}} = \frac{\partial E}{\partial X^{[3]}}\frac{\partial X^{[3]}}{\partial Z^{[3]}} = \left(X^{[3]} - T\right) \circ \left(1 - \tanh\left(Z^{[3]}\right)^2\right)$$

$$\delta^{[2]} = \frac{\partial E}{\partial Z^{[2]}} = \frac{\partial E}{\partial X^{[3]}}\frac{\partial X^{[3]}}{\partial Z^{[3]}}\frac{\partial Z^{[3]}}{\partial X^{[2]}}\frac{\partial X^{[2]}}{\partial Z^{[2]}} = \delta^{[3]}\frac{\partial Z^{[3]}}{\partial X^{[2]}}\frac{\partial X^{[2]}}{\partial Z^{[2]}} = \left(W^{[3]}\right)^T \cdot \delta^{[3]}\frac{\partial X^{[2]}}{\partial Z^{[2]}}$$

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

$$\delta^{[1]} = \frac{\partial E}{\partial Z^{[1]}} = \frac{\partial E}{\partial X^{[3]}}\frac{\partial X^{[3]}}{\partial Z^{[3]}}\frac{\partial Z^{[3]}}{\partial X^{[2]}}\frac{\partial X^{[2]}}{\partial Z^{[2]}}\frac{\partial Z^{[2]}}{\partial X^{[1]}}\frac{\partial X^{[1]}}{\partial Z^{[1]}} = \delta^{[2]}\frac{\partial Z^{[2]}}{\partial X^{[1]}}\frac{\partial X^{[1]}}{\partial Z^{[1]}} = \left(W^{[2]}\right)^T \cdot \delta^{[2]}\frac{\partial X^{[1]}}{\partial Z^{[1]}}$$

$$= \left(W^{[2]}\right)^T \cdot \delta^{[2]} \circ \left(1 - \tanh\left(Z^{[1]}\right)^2\right)$$

Input Layer							Hidden Layer 1								Hidden Layer 2						Output Layer		
X[0]	B[1]	W[1]					Z[1]	X[1]	δ[1]	B[2]	W[2]			Z[2]	X[2]	δ[2]	B[3]	W[3]		Z[3]	X[3]	δ[3]	TARGET
1	1	1	1	1	1	1	6	0,999988	0	1	1	1	1	3,76157	0,99892	0	0	0	0	0	0	-1	1
1	1	0	0	0	0	0	1	0,761594	0	1	1	1	1	3,76157	0,99892	0	0	0	0	0	0	1	-1
1	1	1	1	1	1	1	6	0,999988	0														
1																							
1																							

2. Bias and Weight derivatives:

$$\frac{\partial E}{\partial W^{[3]}} = \frac{\partial E}{\partial X^{[3]}}\frac{\partial X^{[3]}}{\partial Z^{[3]}}\frac{\partial Z^{[3]}}{\partial W^{[3]}} = \delta^{[3]}\frac{\partial Z^{[3]}}{\partial W^{[3]}} = \delta^{[3]} \cdot \left(X^{[2]}\right)^T$$

$$\frac{\partial E}{\partial W^{[2]}} = \frac{\partial E}{\partial X^{[3]}}\frac{\partial X^{[3]}}{\partial Z^{[3]}}\frac{\partial Z^{[3]}}{\partial X^{[2]}}\frac{\partial X^{[2]}}{\partial Z^{[2]}}\frac{\partial Z^{[2]}}{\partial W^{[2]}} = \delta^{[2]}\frac{\partial Z^{[2]}}{\partial W^{[2]}} = \delta^{[2]} \cdot \left(X^{[1]}\right)^T$$

$$\frac{\partial E}{\partial W^{[1]}} = \frac{\partial E}{\partial X^{[3]}}\frac{\partial X^{[3]}}{\partial Z^{[3]}}\frac{\partial Z^{[3]}}{\partial X^{[2]}}\frac{\partial X^{[2]}}{\partial Z^{[2]}}\frac{\partial Z^{[2]}}{\partial X^{[1]}}\frac{\partial X^{[1]}}{\partial Z^{[1]}}\frac{\partial Z^{[1]}}{\partial W^{[1]}} = \delta^{[1]}\frac{\partial Z^{[1]}}{\partial W^{[1]}} = \delta^{[1]} \cdot \left(X^{[0]}\right)^T$$

$$\frac{\partial E}{\partial B^{[3]}} = \frac{\partial E}{\partial X^{[3]}}\frac{\partial X^{[3]}}{\partial Z^{[3]}}\frac{\partial Z^{[3]}}{\partial B^{[3]}} = \delta^{[3]}\frac{\partial Z^{[3]}}{\partial B^{[3]}} = \delta^{[3]}$$

$$\frac{\partial E}{\partial B^{[2]}} = \frac{\partial E}{\partial X^{[3]}}\frac{\partial X^{[3]}}{\partial Z^{[3]}}\frac{\partial Z^{[3]}}{\partial X^{[2]}}\frac{\partial X^{[2]}}{\partial Z^{[2]}}\frac{\partial Z^{[2]}}{\partial B^{[2]}} = \delta^{[2]}\frac{\partial Z^{[2]}}{\partial B^{[2]}} = \delta^{[2]}$$

$$\frac{\partial E}{\partial B^{[1]}} = \frac{\partial E}{\partial X^{[3]}}\frac{\partial X^{[3]}}{\partial Z^{[3]}}\frac{\partial Z^{[3]}}{\partial X^{[2]}}\frac{\partial X^{[2]}}{\partial Z^{[2]}}\frac{\partial X^{[1]}}{\partial Z^{[1]}}\frac{\partial Z^{[1]}}{\partial B^{[1]}} = \delta^{[1]}\frac{\partial Z^{[1]}}{\partial B^{[1]}} = \delta^{[1]}$$

B'[1]	W'[1]				
0	0	0	0	0	0
0	0	0	0	0	0
0	0	0	0	0	0

B'[2]	W'[2]		
0	0	0	0
0	0	0	0

B'[3]	W'[3]	
-1	-0,99892	-0,99892
1	0,99892	0,99892

3. Update Bias and Weights:

$$Update = \begin{cases} W^{[n]} = W^{[n]} - \eta \dfrac{\partial E}{\partial W^{[n]}} \\ B^{[n]} = B^{[n]} - \eta \dfrac{\partial E}{\partial B^{[1]}} \end{cases}, \eta = 0{,}1$$

B[1]	W[1]				
1	1	1	1	1	1
1	0	0	0	0	0
1	1	1	1	1	1

B[2]	W[2]		
1	1	1	1
1	1	1	1

B[3]	W[3]	
0,1	0,099892	0,099892
-0,1	-0,09989	-0,09989

4. Forward Propagation with the new Bias and Weights:

Input Layer							Hidden Layer 1							Hidden Layer 2						Output Layer			
X[0]	B[1]	W[1]					Z[1]	X[1]	δ[1]	B[2]	W[2]			Z[2]	X[2]	δ[2]	B[3]	W[3]		Z[3]	X[3]	δ[3]	TARGET
1	1	1	1	1	1	1	6	0,999988	-1,3764E-08	1	1	1	1	3,76157	0,99892	-0,00028	0,1	0,099892	0,099892	0,299568	0,290917	-0,64907	1
1	1	0	0	0	0	0	1	0,761594	-0,0002352	1	1	1	1	3,76157	0,99892	-0,00028	-0,1	-0,09989	-0,09989	-0,29957	-0,29092	0,649071	-1
1	1	1	1	1	1	1	6	0,999988	-1,3764E-08														
1																							
1																							

$$Error = \frac{1}{2}\sum_{i=1}^{2}\left(T_i - X_i^{[3]}\right)^2 = 0{,}502798$$

b) Mostly equal to 1a), however the difference is that the activation function of the **output layer** is softmax instead of hyperbolic tangent and the loss function is cross-entropy instead of squared error.

Activation Function: $\begin{cases} tanh(x), n < 3 \\ softmax(x), n = 3 \end{cases}$ **Loss Function:** $E = -\sum_{i=1}^{n} T_i Log_2\left(X_i^{[3]}\right)$

Forward Propagation: $Z^{[n]} = W^{[n]} \times X^{[n-1]} + B^{[n]}$ $\begin{cases} X^{[n]} = \tanh\left(Z^{[n]}\right),\ n < 3 \\ X^{[n]} = \dfrac{e^{Z^{[n]}}}{\sum e^{Z^{[n]}}}, n = 3 \end{cases}$

Input Layer							Hidden Layer 1							Hidden Layer 2						Output Layer			
X[0]	BIAS[1]	WEIGHT[1]					Z[1]	X[1]	δ[1]	BIAS[2]	WEIGHT[2]			Z[2]	X[2]	δ[2]	BIAS[3]	WEIGHT[3]		Z[3]	X[3]	δ[3]	TARGET
1	1	1	1	1	1	1	6	0,999988		1	1	1	1	3,76157	0,99892		0	0	0	0	0,5		1
1	1	0	0	0	0	0	1	0,761594		1	1	1	1	3,76157	0,99892		0	0	0	0	0,5		0
1	1	1	1	1	1	1	6	0,999988															
1																							
1																							

$$Error = -\sum_{i=1}^{2} T_i Log_2\left(X_i^{[3]}\right) = 1$$

Backward Propagation:

1. Deltas:

$$\delta^{[3]} = \frac{\partial E}{\partial Z^{[3]}} = \frac{\partial E}{\partial X^{[3]}}\frac{\partial X^{[3]}}{\partial Z^{[3]}} = -\sum_{i \neq j}\left(\frac{T_i}{X_i^{[3]}} \times -X_i^{[3]} \times X_j^{[3]}\right) - \frac{T_i}{X_i^{[3]}} \times X_i^{[3]} \times \left(1 - X_i^{[3]}\right) = \sum_{i \neq j}\left(T_i X_j^{[3]}\right) + T_i X_i^{[3]} - T_i$$

$$= \sum_i \left(T_i X_j^{[3]}\right) - T_i = X_i^{[3]} - T_i = X^{[3]} - T$$

$$\frac{\partial E}{\partial X^{[3]}} = -\sum_i \frac{t_i}{X_i^{[3]}} \qquad \frac{\partial X^{[3]}}{\partial Z^{[3]}} = \begin{cases} X_i^{[3]} \times \left(1 - X_i^{[3]}\right), i = j \\ -X_i^{[3]} \times X_j^{[3]}, i \neq j \end{cases}$$

Aprendizagem 2021/22

Homework I – Group 047

We can see that $\delta^{[3]}$ is the only delta different from 1a), due to the fact that the others layers have the same activation function as before, they are the same

$$\delta^{[2]} = \frac{\partial E}{\partial Z^{[2]}} = \left(W^{[3]}\right)^T \cdot \delta^{[3]} \circ \left(1 - \tanh\left(Z^{[2]}\right)^2\right)$$

$$\delta^{[1]} = \frac{\partial E}{\partial Z^{[1]}} = \left(W^{[2]}\right)^T \cdot \delta^{[2]} \circ \left(1 - \tanh\left(Z^{[1]}\right)^2\right)$$

Input Layer							Hidden Layer 1							Hidden Layer 2						Output Layer			
X[0]	BIAS[1]	WEIGHT[1]					Z[1]	X[1]	δ[1]	BIAS[2]	WEIGHT[2]			Z[2]	X[2]	δ[2]	BIAS[3]	WEIGHT[3]		Z[3]	X[3]	δ[3]	TARGET
1	1	1	1	1	1	1	6	0,999988	0	1	1	1	1	3,76157	0,99892	0	0	0	0	0	0,5	-0,5	1
1	1	0	0	0	0	0	1	0,761594	0	1	1	1	1	3,76157	0,99892	0	0	0	0	0	0,5	0,5	0
1	1	1	1	1	1	1	6	0,999988	0														
1																							
1																							

2. Bias and Weight derivatives:

$$\frac{\partial E}{\partial W^{[3]}} = \delta^{[3]} \cdot \left(X^{[2]}\right)^T \qquad \frac{\partial E}{\partial W^{[2]}} = \delta^{[2]} \cdot \left(X^{[1]}\right)^T \qquad \frac{\partial E}{\partial W^{[1]}} = \delta^{[1]} \cdot \left(X^{[0]}\right)^T$$

$$\frac{\partial E}{\partial B^{[3]}} = \delta^{[3]} \qquad \frac{\partial E}{\partial B^{[2]}} = \delta^{[2]} \qquad \frac{\partial E}{\partial B^{[1]}} = \delta^{[1]}$$

B'[1]	W'[1]				
0	0	0	0	0	0
0	0	0	0	0	0
0	0	0	0	0	0

B'[2]	W'[2]		
0	0	0	0
0	0	0	0

B'[3]	W'[3]	
-0,5	-0,49946	-0,49946
0,5	0,49946	0,49946

3. Update Bias and Weights:

$$Update = \begin{cases} W^{[n]} = W^{[n]} - \eta \dfrac{\partial E}{\partial W^{[n]}} \\ B^{[n]} = B^{[n]} - \eta \dfrac{\partial E}{\partial B^{[1]}} \end{cases}, \eta = 0{,}1$$

B[1]	W[1]				
1	1	1	1	1	1
1	0	0	0	0	0
1	1	1	1	1	1

B[2]	W[2]		
1	1	1	1
1	1	1	1

B[3]	W[3]	
0,05	0,049946	0,049946
-0,05	-0,04995	-0,04995

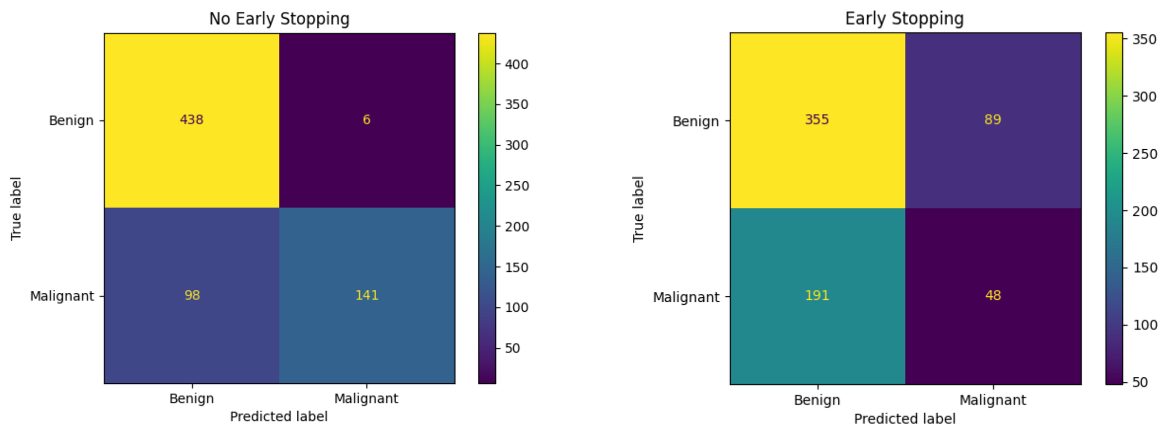
4. Forward Propagation with the new Bias and Weights:

Input Layer							Hidden Layer 1							Hidden Layer 2						Output Layer			
X[0]	BIAS[1]	WEIGHT[1]					Z[1]	X[1]	δ[1]	BIAS[2]	WEIGHT[2]			Z[2]	X[2]	δ[2]	BIAS[3]	WEIGHT[3]		Z[3]	X[3]	δ[3]	TARGET
1	1	1	1	1	1	1	6	0,999988	-4,5E-09	1	1	1	1	3,76157	0,99892	-9,2E-05	0,05	0,049946	0,049946	0,149784	0,574337	-0,42566	1
1	1	0	0	0	0	0	1	0,761594	-7,7E-05	1	1	1	1	3,76157	0,99892	-9,2E-05	-0,05	-0,04995	-0,04995	-0,14978	0,425663	0,425663	0
1	1	1	1	1	1	1	6	0,999988	-4,5E-09														
1																							
1																							

$$Error = -\sum_{i=1}^{2} T_i Log_2\left(X_i^{[3]}\right) = 0{,}800031$$

II. Programming and critical analysis

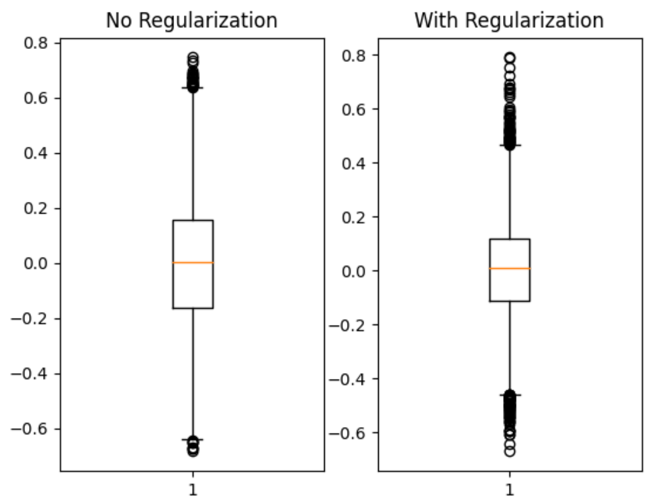
2)



We can observe that the predictions with no early stopping are closer to the real values, which can be because:

1. Early stopping made the algorithm stop in a local minimum, not allowing it to fully learn;
2. Not all the data was used to train, which can diminish the quality of the model, especially if the distribution of the data wasn't random.

3)



Four strategies that would minimize the error are:

1. A good choice for the regularization value;
2. Using early stopping that could reduce overfitting;
3. Choosing a different algorithm that may be better at fitting the training data and predict better results, too complex may give overfitting, but too simple may be unable to adapt to the data;
4. Choosing a different variant of the given data, it's said to be the variant 8nm, but can be others that better adjust the given model.

III. APPENDIX

```
from sklearn.metrics import confusion_matrix, ConfusionMatrixDisplay
from sklearn.neural_network import MLPClassifier, MLPRegressor
from sklearn.model_selection import KFold, cross_val_predict
import matplotlib.pyplot as plt
from matplotlib import pylab
from scipy.io import arff
import numpy as np

Kfol = KFold(n_splits=5, random_state=0, shuffle=True)

#*2

file = open("breast.w.arff", "r")
data, meta = arff.loadarff(file)
input = data[meta.names()[:-1]].tolist()
output = data["Class"].tolist()

for earlyStop in [False, True]:
    classifier = MLPClassifier(hidden_layer_sizes=(3,2), activation='relu', alpha=0.2,
max_iter=2000, early_stopping=earlyStop)
    prevision = cross_val_predict(classifier, input, output, cv=Kfol)
    conf_mat = confusion_matrix(output, prevision)
    disp  = ConfusionMatrixDisplay(conf_mat, display_labels=['Benign','Malignant'])
    disp.plot()
    if earlyStop:
        plt.title("Early Stopping")
    else:
        plt.title("No Early Stopping")

plt.show()

#*3
file.close()
file = open("kin8nm.arff", "r")
data, meta = arff.loadarff(file)

input = data[meta.names()[:-1]].tolist()
output = data["y"].tolist()

for alpha, graph in zip([0, 0.2], [1,2]):
    classifier = MLPRegressor(hidden_layer_sizes=(3,2), activation='relu', alpha=alpha,
max_iter=2000)
    classifier.fit(input, output)
    prevision = cross_val_predict(classifier, input, output, cv=Kfol)
```

```
    residues = np.subtract(output, prevision)
    plt.subplot(1, 2, graph)
    plt.boxplot(residues)
    if graph == 1:
        plt.title("No Regularization")
    else:
        plt.title("With Regularization")

plt.show()
```

END