

## Neural Network TASK 3

### Team ID: Bio36

20191701034	امانى يحيى ابراهيم محمد
20191701097	سارة كمال محمود محمد
20191701199	مصطفى حسام احمد احمد
20191701188	مريانة اشرف شحاتة
20191701263	يمنى صلاح محروس

#### 1) Test case using sigmoid function:

The image shows a Python IDE with a script for training and testing a neural network. The script defines a `testing` function and uses it to evaluate the model. A 'Back Propagation' configuration window is open, showing settings for Learning Rate (0.1), Epochs (1000), Neurons (3,4), Layers (2), and the Sigmoid activation function. The output window displays the training and testing results, including accuracy and confusion matrices.

```
def testing(xtest,bias,functions, layers,numofneurons, ytest):  
    trainweight = training(xtrain,ytrain,bias,etanum,epochnum, layers,numofneurons, functions)  
    print("Training Accuracy: ")  
    testing(xtrain, bias, functions, layers, numofneurons, ytrain)  
    print("-----")  
    print('Testing Accuracy: ')  
    totting(xtest,bias,functions, layers,numofneurons, ytest)  
    print("-----")
```

Back Propagation

Learning Rate: 0.1  
Epochs: 1000  
Neurons: 3,4  
Layers: 2  
Bias: ☒  
Sigmoid: ☒  
Submit

ANN3.py

C:\Users\hp\PycharmProjects\ANN\venv\Scripts\python.exe C:/Users/hp/PycharmProjects/ANN/ANN3.py

C:\Users\hp\PycharmProjects\ANN\venv\Scripts\python.exe C:/Users/hp/PycharmProjects/ANN/ANN3.py

Training Accuracy:  
Confusion Matrix  
[[14 6]  
 [45 25]]  
Overall accuracy is 65.0

-----

Testing Accuracy:  
Confusion Matrix  
[[ 6 14]  
 [21 19]]  
Overall accuracy is 41.66666666666667

Sigmoid

## 2) Test case using tangent function:

The image shows a PyCharm IDE environment with three main components:

- Code Editor:** Displays a Python script named `ANN3.py`. The script includes logic for calculating a confusion matrix and overall accuracy. Key lines include:

```
fn += 1
confusionMatrix = np.array([[tp, fp], [fn, tn]])
print("Confusion Matrix")
print(confusionMatrix)
print("Overall accuracy is ", ((tp + tn) / 60) * 100)

trainweight = training(xtrain, ytrain, bias, etanum, epochsnum, layers, numofneurons)
print("Training Accuracy: ")
testing(xtrain, bias, functions, layers, numofneurons, ytrain)
print("-----")
print('Testing Accuracy: ')
testing(xtest, bias, functions, layers, numofneurons, ytest)
print("-----")
```
- Back Propagation Window:** A configuration panel with the following settings:
  - Learning Rate:
  - Epochs:
  - Neurons:
  - Layers:
  - ☒ Bias
  - 
  -
- Terminal:** Shows the execution output:

```
Testing Accuracy:
Confusion Matrix
[[ 8 12]
 [11 29]]
Overall accuracy is  61.66666666666667
-----

Process finished with exit code 0
```