COMP- 237 Online lab Assignment 'Neural Networks'

Exercise: 1 - Single layer feed forward to recognize sum pattern

Step2: Generate Input Data

We create two sets of ten numbers using numpy, which are taken from the uniform distribution between -0.6 and +0.6.

Step3: Create the target data

The total of the two random values for every instance of the input data is the target. The values are summed along the second axis using numpy, and the output is reshaped into a 10x1 array.

Step5: Create a Neural Network

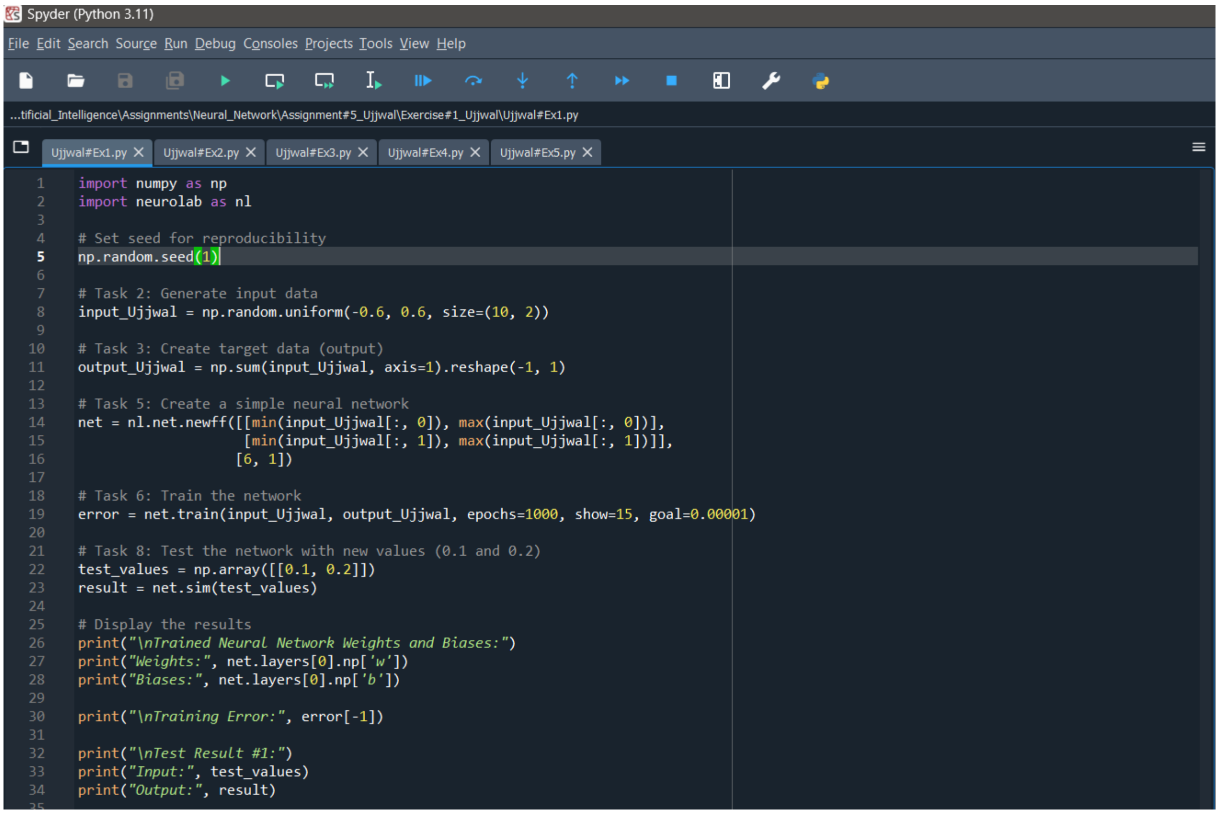
We build a basic neural network consisting of one output, one hidden layer with six neurons, and two inputs.

Step 6: Train the Network

Using the input and output data and the predetermined parameters (epochs=1000, show=15, goal=0.00001), we train the network.

Step 8: Test the Network

We apply new values (0.1 and 0.2) to the trained network and show the outcome.


Spyder (Python 3.11)
File Edit Search Source Run Debug Consoles Projects Tools View Help
...tificial_Intelligence\Assignments\Neural_Network\Assignment#5_Ujjwal\Exercise#1_Ujjwal\Ujjwal#Ex1.py
Ujjwal#Ex1.py
Ujjwal#Ex2.py
Ujjwal#Ex3.py
Ujjwal#Ex4.py
Ujjwal#Ex5.py

```python
import numpy as np
import neurolab as nl

# Set seed for reproducibility
np.random.seed(1)

# Task 2: Generate input data
input_Ujjwal = np.random.uniform(-0.6, 0.6, size=(10, 2))

# Task 3: Create target data (output)
output_Ujjwal = np.sum(input_Ujjwal, axis=1).reshape(-1, 1)

# Task 5: Create a simple neural network
net = nl.net.newff([[min(input_Ujjwal[:, 0]), max(input_Ujjwal[:, 0])],
                    [min(input_Ujjwal[:, 1]), max(input_Ujjwal[:, 1])]],
                   [6, 1])

# Task 6: Train the network
error = net.train(input_Ujjwal, output_Ujjwal, epochs=1000, show=15, goal=0.00001)

# Task 8: Test the network with new values (0.1 and 0.2)
test_values = np.array([[0.1, 0.2]])
result = net.sim(test_values)

# Display the results
print("\nTrained Neural Network Weights and Biases:")
print("Weights:", net.layers[0].np['w'])
print("Biases:", net.layers[0].np['b'])

print("\nTraining Error:", error[-1])

print("\nTest Result #1:")
print("Input:", test_values)
print("Output:", result)
```

Output of Exercise 1 is attached in the last section.

Exercise 2: Multi-layer feed forward to recognize sum pattern

Step 1: Establish a Two-Layer Feedforward Network

Two hidden layers—the first with five neurons and the second with three neurons—are added to the network design.

a. epochs=1000
b. show=100
c. goal=0.00001

Step 2: Test the Network
We apply new values (0.1 and 0.2) to the trained network and show the outcome. .

Comparing the Outcomes:

Network Structure:

Result #1: a single-layer network.

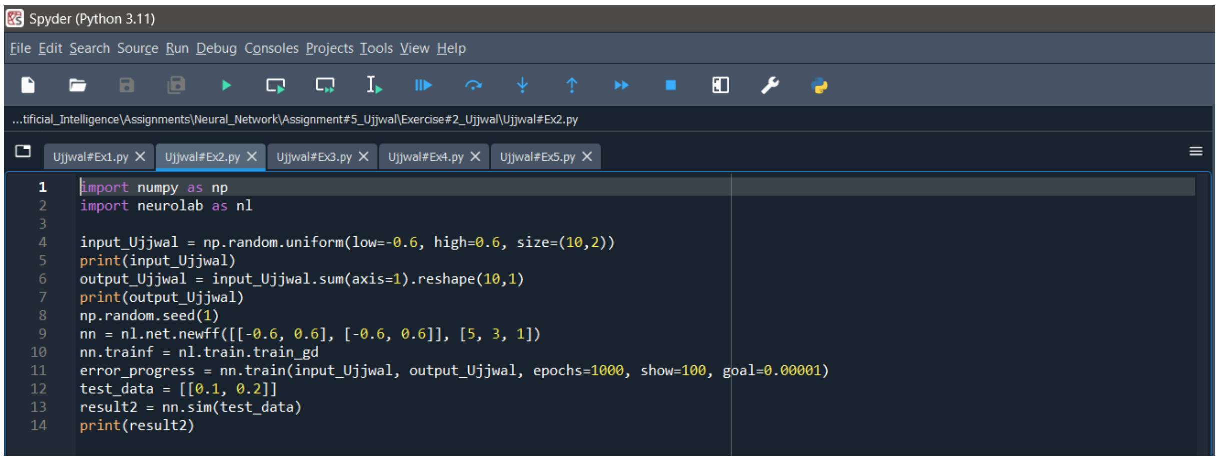
Result #2: A two-layer network with a hidden layer on top of it.

Error in Training: A better match to the training data is indicated by a smaller training error.

Test Findings: Examine and contrast the test value outputs (0.1, 0.2). This shows the degree to which the networks generalize to new data.

Effects of Modified Architecture: Network's performance of adding a second hidden layer with a varied number of neurons.

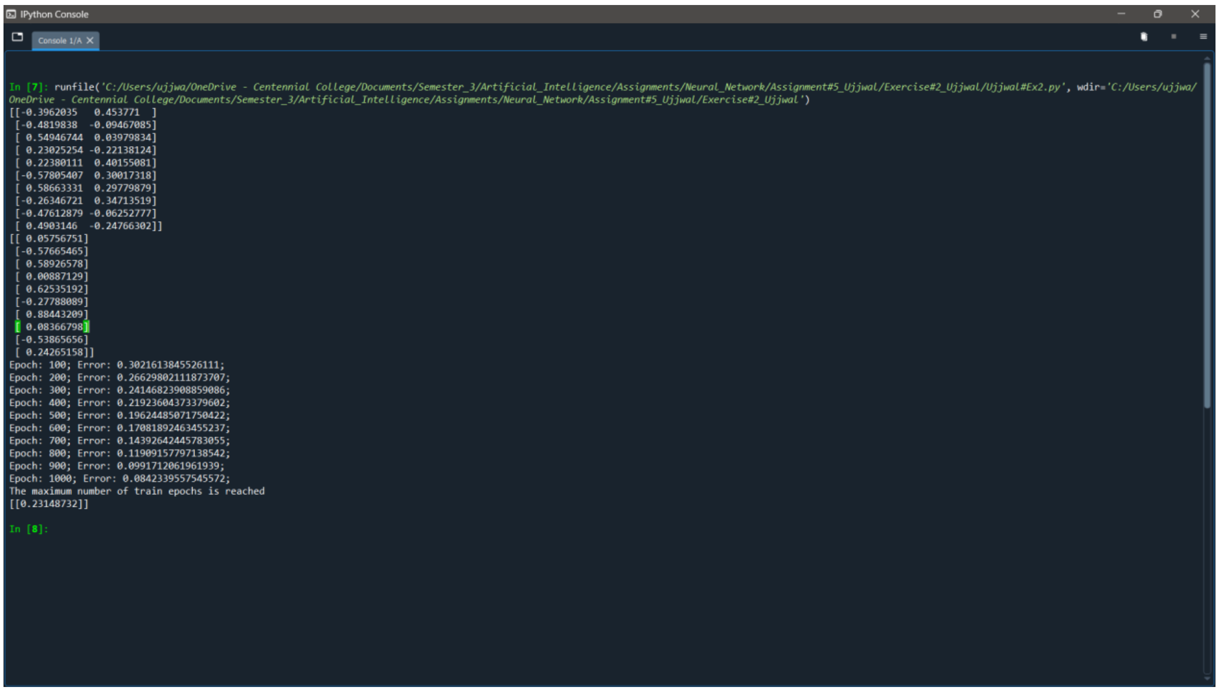
Overall Results: Provide a summary of the main conclusions and explain the implications of modifying the network architecture.


Spyder (Python 3.11)
File Edit Search Source Run Debug Consoles Projects Tools View Help
...tificial_Intelligence\Assignments\Neural_Network\Assignment#5_Ujjwal\Exercise#2_Ujjwal\Ujjwal#Ex2.py
Ujjwal#Ex1.py
Ujjwal#Ex2.py
Ujjwal#Ex3.py
Ujjwal#Ex4.py
Ujjwal#Ex5.py

```python
import numpy as np
import neurolab as nl

input_Ujjwal = np.random.uniform(low=-0.6, high=0.6, size=(10,2))
print(input_Ujjwal)
output_Ujjwal = input_Ujjwal.sum(axis=1).reshape(10,1)
print(output_Ujjwal)
np.random.seed(1)
nn = nl.net.newff([[-0.6, 0.6], [-0.6, 0.6]], [5, 3, 1])
nn.trainf = nl.train.train_gd
error_progress = nn.train(input_Ujjwal, output_Ujjwal, epochs=1000, show=100, goal=0.00001)
test_data = [[0.1, 0.2]]
result2 = nn.sim(test_data)
print(result2)
```

Output of Exercise 2:



```
In [7]: runfile('C:/Users/ujjwa/OneDrive - Centennial College/Documents/Semester_3/Artificial_Intelligence/Assignments/Neural_Network/Assignment#5_Ujjwal/Exercise#2_Ujjwal/Ujjwal#Ex2.py', wdir='C:/Users/ujjwa/
OneDrive - Centennial College/Documents/Semester_3/Artificial_Intelligence/Assignments/Neural_Network/Assignment#5_Ujjwal/Exercise#2_Ujjwal')
[[-0.3962035   0.453771  ]
 [-0.4819838  -0.09467085]
 [ 0.54946744  0.03979834]
 [ 0.23025254 -0.22138124]
 [ 0.22380111  0.40155081]
 [-0.57805407  0.30017318]
 [ 0.58663331  0.29779879]
 [-0.26346721  0.34713519]
 [-0.47612879 -0.06252777]
 [ 0.4903146  -0.24766302]]
[[ 0.05756751]
 [-0.57665465]
 [ 0.58926578]
 [ 0.00887129]
 [ 0.62535192]
 [-0.27788089]
 [ 0.88443209]
 [ 0.08366798]
 [-0.53865656]
 [ 0.24265158]]
Epoch: 100; Error: 0.3021613845526111;
Epoch: 200; Error: 0.26629802111873707;
Epoch: 300; Error: 0.24146823908859086;
Epoch: 400; Error: 0.21923604373379602;
Epoch: 500; Error: 0.19624485071750422;
Epoch: 600; Error: 0.17081892463455237;
Epoch: 700; Error: 0.14392642445783055;
Epoch: 800; Error: 0.11909157797138542;
Epoch: 900; Error: 0.0991712061961939;
Epoch: 1000; Error: 0.0842339557545572;
The maximum number of train epochs is reached
[[0.23148732]]

In [8]:
```

Exercise 3: Single-layer feed forward to recognize sum pattern with more training data

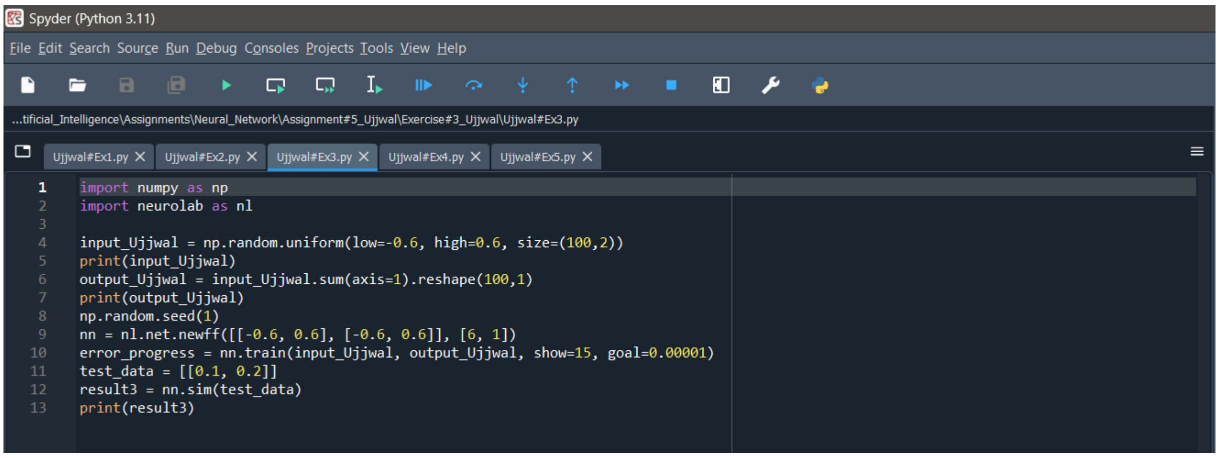
Produce 100 Instances of Input Data (Steps 1-3):

By employing the uniform distribution between -0.6 and 0.6, we get 100 random occurrences.

Build and train a neural network (Steps 4-8):

Five neurons make up the first hidden layer, three neurons make up the second hidden layer, and one neuron serves as the output of our two-layer feedforward neural network.

Step 8: Test the Network


Spyder (Python 3.11)
File Edit Search Source Run Debug Consoles Projects Tools View Help
...tificial_Intelligence\Assignments\Neural_Network\Assignment#5_Ujjwal\Exercise#3_Ujjwal\Ujjwal#Ex3.py
Ujjwal#Ex1.py
Ujjwal#Ex2.py
Ujjwal#Ex3.py
Ujjwal#Ex4.py
Ujjwal#Ex5.py

```python
import numpy as np
import neurolab as nl

input_Ujjwal = np.random.uniform(low=-0.6, high=0.6, size=(100,2))
print(input_Ujjwal)
output_Ujjwal = input_Ujjwal.sum(axis=1).reshape(100,1)
print(output_Ujjwal)
np.random.seed(1)
nn = nl.net.newff([[-0.6, 0.6], [-0.6, 0.6]], [6, 1])
error_progress = nn.train(input_Ujjwal, output_Ujjwal, show=15, goal=0.00001)
test_data = [[0.1, 0.2]]
result3 = nn.sim(test_data)
print(result3)
```

Output of Exercise 3 is In the last section

Exercise 4: Multi-layer feed forward to recognize sum pattern with more training data

Create Input Data (First Step):

Using the uniform distribution between -0.6 and 0.6, we produce 100 random samples.

Build and train a neural network (steps 2–6):

Five neurons make up the first hidden layer, three neurons make up the second hidden layer, and one neuron serves as the output of our two-layer feedforward neural network.

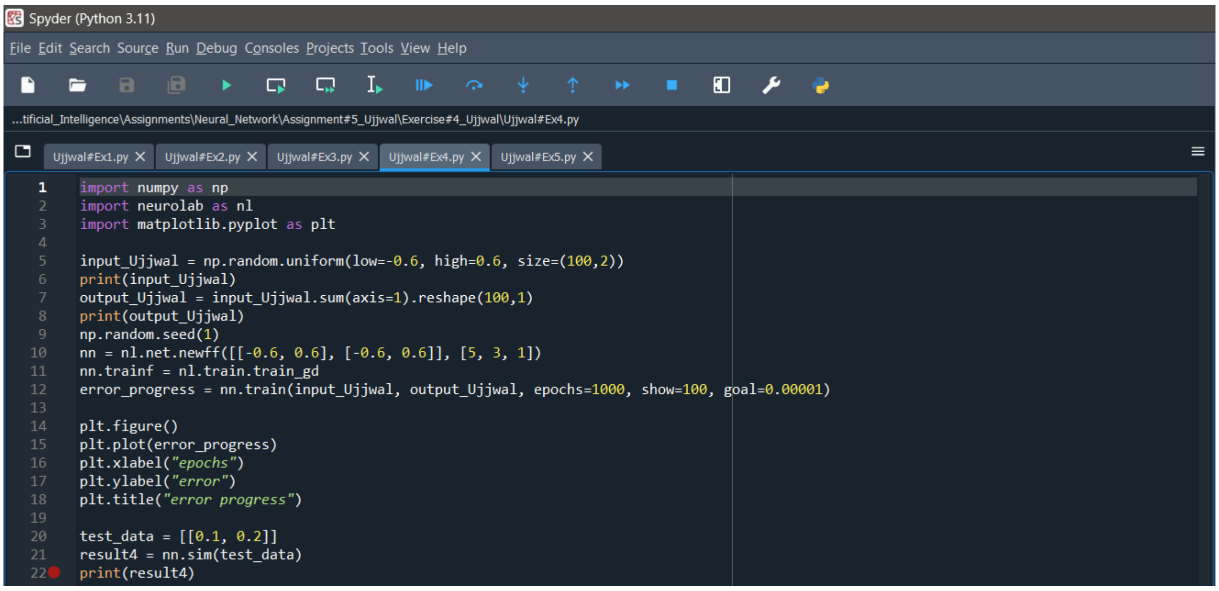
Utilizing the updated input and output data, train the network.

Mistake in Plot Training (Step 5):

To see how the training process is convergent, we depict the training error across epochs.

Step 6: Test the Network

Using the designated test values (0.1 and 0.2), we test the trained network.


Spyder (Python 3.11)
File Edit Search Source Run Debug Consoles Projects Tools View Help
...tificial_Intelligence\Assignments\Neural_Network\Assignment#5_Ujjwal\Exercise#4_Ujjwal\Ujjwal#Ex4.py
Ujjwal#Ex1.py
Ujjwal#Ex2.py
Ujjwal#Ex3.py
Ujjwal#Ex4.py
Ujjwal#Ex5.py

```python
import numpy as np
import neurolab as nl
import matplotlib.pyplot as plt

input_Ujjwal = np.random.uniform(low=-0.6, high=0.6, size=(100,2))
print(input_Ujjwal)
output_Ujjwal = input_Ujjwal.sum(axis=1).reshape(100,1)
print(output_Ujjwal)
np.random.seed(1)
nn = nl.net.newff([[-0.6, 0.6], [-0.6, 0.6]], [5, 3, 1])
nn.trainf = nl.train.train_gd
error_progress = nn.train(input_Ujjwal, output_Ujjwal, epochs=1000, show=100, goal=0.00001)

plt.figure()
plt.plot(error_progress)
plt.xlabel("epochs")
plt.ylabel("error")
plt.title("error progress")

test_data = [[0.1, 0.2]]
result4 = nn.sim(test_data)
print(result4)
```

Output of Exercise 4 is in the last section.

Exercise 5: Three input multi-layer feed forward to recognize sum pattern with more training data

Step 1: Repeat exercises # 1 but instead of having two inputs generate three inputs.

Step 2: Test/Simulate the following test sample [0.2,0.1,0.2] record the results in result #5

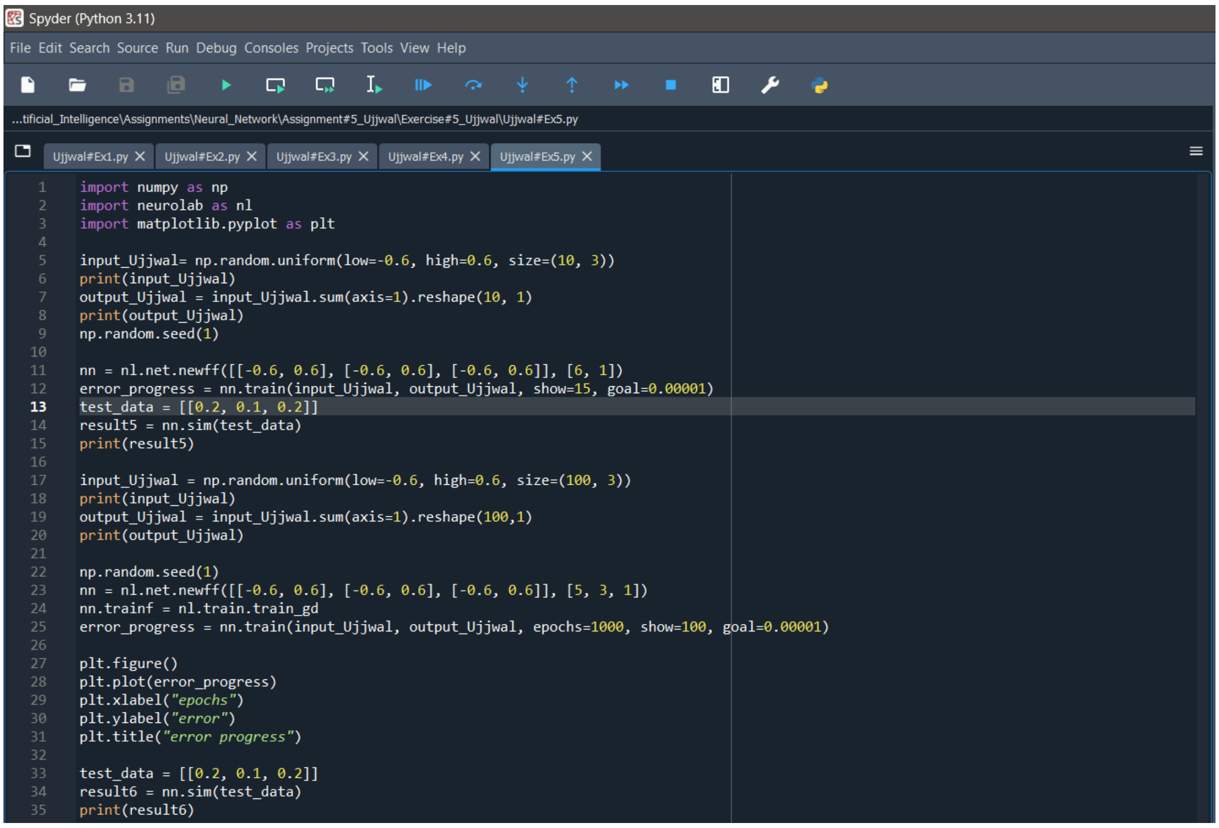
Step 3: Repeat exercise #4 but instead of having two inputs generate three inputs

Step 4: Test/Simulate the same test in point 10 i.e [0.2,0.1,0.2] record the results in result #6

Step 5: Comparing and Analysing

Three inputs with gradient descent backpropagation from Exercise #4.

Effects of training method (gradient descent backpropagation vs. standard training) and number of inputs on training error and network generalization to a new sample.


Spyder (Python 3.11)
File Edit Search Source Run Debug Consoles Projects Tools View Help
...tificial_Intelligence\Assignments\Neural_Network\Assignment#5_Ujjwal\Exercise#5_Ujjwal\Ujjwal#Ex5.py
Ujjwal#Ex1.py
Ujjwal#Ex2.py
Ujjwal#Ex3.py
Ujjwal#Ex4.py
Ujjwal#Ex5.py

```python
import numpy as np
import neurolab as nl
import matplotlib.pyplot as plt

input_Ujjwal= np.random.uniform(low=-0.6, high=0.6, size=(10, 3))
print(input_Ujjwal)
output_Ujjwal = input_Ujjwal.sum(axis=1).reshape(10, 1)
print(output_Ujjwal)
np.random.seed(1)

nn = nl.net.newff([[-0.6, 0.6], [-0.6, 0.6], [-0.6, 0.6]], [6, 1])
error_progress = nn.train(input_Ujjwal, output_Ujjwal, show=15, goal=0.00001)
test_data = [[0.2, 0.1, 0.2]]
result5 = nn.sim(test_data)
print(result5)

input_Ujjwal = np.random.uniform(low=-0.6, high=0.6, size=(100, 3))
print(input_Ujjwal)
output_Ujjwal = input_Ujjwal.sum(axis=1).reshape(100,1)
print(output_Ujjwal)

np.random.seed(1)
nn = nl.net.newff([[-0.6, 0.6], [-0.6, 0.6], [-0.6, 0.6]], [5, 3, 1])
nn.trainf = nl.train.train_gd
error_progress = nn.train(input_Ujjwal, output_Ujjwal, epochs=1000, show=100, goal=0.00001)

plt.figure()
plt.plot(error_progress)
plt.xlabel("epochs")
plt.ylabel("error")
plt.title("error progress")

test_data = [[0.2, 0.1, 0.2]]
result6 = nn.sim(test_data)
print(result6)
```

Output of Exercise is in the last.

```
In [4]:         'C:/Users/ujjwa/OneDrive - Centennial College/Documents/Semester_3/
Artificial_Intelligence/Assignments/Neural_Network/Assignment#5_Ujjwal/Exercise#1_Ujjwal/
Ujjwal#Ex1.py'      ='C:/Users/ujjwa/OneDrive - Centennial College/Documents/Semester_3/
Artificial_Intelligence/Assignments/Neural_Network/Assignment#5_Ujjwal/Exercise#1_Ujjwal'
Epoch: 15; Error: 0.05504985044608453;
Epoch: 30; Error: 0.02459366461928936;
Epoch: 45; Error: 0.006163642909379176;
Epoch: 60; Error: 0.0032412571807674787;
Epoch: 75; Error: 0.001868578300121664;
Epoch: 90; Error: 0.0017156740696474245;
Epoch: 105; Error: 0.0016110178975384736;
Epoch: 120; Error: 0.0015441792320397944;
Epoch: 135; Error: 0.0015034566232988154;
Epoch: 150; Error: 0.0014809588093948785;
Epoch: 165; Error: 0.0014638087310227751;
Epoch: 180; Error: 0.0014522863629449862;
Epoch: 195; Error: 0.0014430313744104675;
Epoch: 210; Error: 0.0014359297520802727;
Epoch: 225; Error: 0.0014289235332432388;
Epoch: 240; Error: 0.00142372640725318;
Epoch: 255; Error: 0.0014190792303657223;
Epoch: 270; Error: 0.0014156573939232164;
Epoch: 285; Error: 0.0014131038174272574;
Epoch: 300; Error: 0.0014113533458953531;
Epoch: 315; Error: 0.00140984330619382 4;
Epoch: 330; Error: 0.001408491998601416;
Epoch: 345; Error: 0.0014072408052989305;
Epoch: 360; Error: 0.0014062850518774847;
Epoch: 375; Error: 0.0014054042010134481;
Epoch: 390; Error: 0.0014046418417884904;
Epoch: 405; Error: 0.00140390947004059;
Epoch: 420; Error: 0.0014032553786383225;
Epoch: 435; Error: 0.001402595996500435;
Epoch: 450; Error: 0.0014020534904563957;
Epoch: 465; Error: 0.0014015212315318327;
Epoch: 480; Error: 0.0014011431073354351;
Epoch: 495; Error: 0.0014007848328517311;
Epoch: 510; Error: 0.0014004338773034714;
Epoch: 525; Error: 0.00140016493008776;
Epoch: 540; Error: 0.0013999037717544209;
Epoch: 555; Error: 0.0013996529193850355;
Epoch: 570; Error: 0.001399451377316713;
Epoch: 585; Error: 0.0013992437880473423;
Epoch: 600; Error: 0.0013990702244150114;
Epoch: 615; Error: 0.0013989063360933796;
Epoch: 630; Error: 0.001398764348582256;
Epoch: 645; Error: 0.001398625415868699;
Epoch: 660; Error: 0.001398514939287011;
Epoch: 675; Error: 0.001398405895755039;
Epoch: 690; Error: 0.0013983215752101953;
Epoch: 705; Error: 0.0013982239157246;
Epoch: 720; Error: 0.0013981584484014874;
Epoch: 735; Error: 0.0013981030575059405;
```

```
Epoch: 750; Error: 0.0013980358558683536;
Epoch: 765; Error: 0.0013979792946293825;
Epoch: 780; Error: 0.0013979222842967196;

Trained Neural Network Weights and Biases:
Weights: [[  0.52822462   0.54604801]
 [ 18.91579205  -8.00151204]
 [  4.50569876   0.93121747]
 [ -8.0020944   -4.37280869]
 [ -8.77084625   5.38155914]
 [-13.38243726  -1.40189591]]
Biases: [  3.84763837 -64.46612253  22.65569844 -15.8778973  -14.66157037
 -22.02513957]

Training Error: 0.0013979222842967196

Test Result #1:
Input: [[0.1 0.2]]
Output: [[0.2354452]]

In [5]:
```

```
In [9]:         'C:/Users/ujjwa/OneDrive - Centennial College/Documents/Semester_3/
Artificial_Intelligence/Assignments/Neural_Network/Assignment#5_Ujjwal/Exercise#3_Ujjwal/
Ujjwal#Ex3.py'      ='C:/Users/ujjwa/OneDrive - Centennial College/Documents/Semester_3/
Artificial_Intelligence/Assignments/Neural_Network/Assignment#5_Ujjwal/Exercise#3_Ujjwal'
[[-0.3962035   0.453771  ]
 [-0.4819838  -0.09467085]
 [ 0.54946744  0.03979834]
 [ 0.23025254 -0.22138124]
 [ 0.22380111  0.40155081]
 [-0.57805407  0.30017318]
 [ 0.58663331  0.29779879]
 [-0.26346721  0.34713519]
 [-0.47612879 -0.06252777]
 [ 0.4903146  -0.24766302]
 [-0.25466959 -0.44396571]
 [-0.57675965  0.21460264]
 [-0.34604626 -0.28134401]
 [-0.01011221 -0.53596495]
 [ 0.08894113 -0.42392571]
 [ 0.10716664  0.23971003]
 [-0.47719869 -0.10313281]
 [ 0.23328019 -0.10298488]
 [-0.54005585  0.04307569]
 [ 0.19655357  0.01786693]
 [ 0.53351371  0.10386605]
 [ 0.4840823  -0.43503036]
 [-0.43286838  0.36886955]
 [-0.1227878  -0.40157496]
 [ 0.5130103  -0.18268097]
 [ 0.30097452  0.27119758]
 [ 0.45996731  0.14840665]
 [ 0.30113092 -0.18132199]
 [-0.27608653  0.47506346]
 [-0.08629057  0.55780806]
 [ 0.1961298   0.14603486]
 [-0.46230483  0.53938711]
 [-0.06010544  0.09406754]
 [-0.11023584 -0.31556762]
 [ 0.48405542  0.08841538]
 [-0.59655561  0.1405739 ]
 [-0.20802612  0.03246972]
 [ 0.46313052 -0.17127629]
 [ 0.49024218  0.14803214]
 [-0.58101451  0.51532468]
 [ 0.2290763   0.59678742]
 [-0.39319139 -0.4354371 ]
 [ 0.51911456  0.23618179]
 [-0.52079979  0.30655566]
 [ 0.30465143  0.50762944]
 [ 0.25382971 -0.45087485]
 [-0.57614384 -0.56854682]
 [-0.56603221 -0.30454672]
 [ 0.43203354  0.04659728]
 [ 0.06338637  0.41043707]
 [-0.45099202 -0.26497959]
 [ 0.10291113  0.5635149 ]
 [ 0.07323626 -0.57762325]
 [ 0.36075921 -0.32043087]
 [ 0.36852623 -0.13456723]
 [ 0.43625023  0.29654597]
 [ 0.06748828 -0.43625373]
 [-0.52809877 -0.45438785]
 [-0.54653775 -0.47100705]
 [-0.32914879  0.25558678]
 [ 0.07166038 -0.58493282]
 [-0.51363086  0.5607316 ]
 [ 0.08172055 -0.35604812]
 [-0.29720911  0.29259102]
 [-0.36548462  0.09763071]
 [ 0.56402399  0.41619456]
 [-0.31218269 -0.00747634]
 [ 0.14394686  0.39477708]
 [-0.41185033 -0.57770856]
 [-0.51597343 -0.01638587]
 [ 0.12759535  0.08262172]
 [-0.21916511  0.58633939]
 [ 0.09569426 -0.14383059]
 [ 0.06113786  0.29440132]
 [ 0.20307947 -0.28209653]
 [-0.5203982  -0.15589896]
 [ 0.15566101 -0.34779119]
 [ 0.30330666 -0.52015622]
 [-0.28762188  0.36570548]
 [-0.36787886  0.16735306]
 [ 0.02960437  0.50976956]
 [-0.28404388 -0.52084669]
 [ 0.28207916  0.32661364]
 [ 0.48937902  0.51836648]
 [-0.58325811 -0.3187655 ]
 [ 0.14013403  0.53881958]
 [ 0.54021134  0.06798383]
 [ 0.49872762  0.16987945]
 [-0.13199074 -0.0168112 ]
 [ 0.12517258  0.05945751]
 [ 0.51141771  0.50248012]
 [-0.12614926  0.55591503]
 [-0.3912532  -0.44840458]
 [-0.43790501  0.0067946 ]
 [-0.57417023  0.53756425]
 [ 0.39253857 -0.58197722]
 [-0.38856449 -0.20152371]
 [-0.44280379  0.37138883]
 [-0.18631602  0.52812898]
 [ 0.09841702  0.45459838]]
[[ 0.05756751]
 [-0.57665465]
 [ 0.58926578]
 [ 0.00887129]
[ 0.62535192]
[-0.27788089]
[ 0.88443209]
[ 0.08366798]
[-0.53865656]
[ 0.24265158]
[-0.69863531]
[-0.36215701]
[-0.62739027]
[-0.54607715]
[-0.33498458]
[ 0.34687668]
[-0.5803315 ]
[ 0.13029531]
[-0.49698016]
[ 0.21442051]
[ 0.63737976]
[ 0.04905194]
[-0.06399884]
[-0.52436276]
[ 0.33032933]
[ 0.57217211]
[ 0.60837396]
[ 0.11980893]
[ 0.19897693]
[ 0.47151748]
[ 0.34216466]
[ 0.07708228]
[ 0.0339621 ]
[-0.42580346]
[ 0.57247081]
[-0.45598171]
[-0.1755564 ]
[ 0.29185423]
[ 0.63827432]
[-0.06568983]
[ 0.82586372]
[-0.82862849]
[ 0.75529635]
[-0.21424413]
[ 0.81228087]
[-0.19704514]
[-1.14469066]
[-0.87057893]
[ 0.47863082]
[ 0.47382345]
[-0.71597161]
[ 0.66642602]
[-0.50438699]
[ 0.04032834]
[ 0.23395901]
[ 0.7327962 ]
[-0.36876545]
[-0.98248663]
[-1.01754479]
```

```
 [-0.07356202]
 [-0.51327245]
 [ 0.04710073]
 [-0.27432756]
 [-0.00461808]
 [-0.26785391]
 [ 0.98021855]
 [-0.31965903]
 [ 0.53872394]
 [-0.98955888]
 [-0.53235929]
 [ 0.21021708]
 [ 0.36717428]
 [-0.04813633]
 [ 0.35553918]
 [-0.07901706]
 [-0.67629716]
 [-0.19213018]
 [-0.21684956]
 [ 0.07808359]
 [-0.2005258 ]
 [ 0.53937394]
 [-0.80489057]
 [ 0.60869279]
 [ 1.00774551]
 [-0.90202361]
 [ 0.67895361]
 [ 0.60819517]
 [ 0.66860707]
 [-0.14880194]
 [ 0.18463009]
 [ 1.01389783]
 [ 0.42976577]
 [-0.83965778]
 [-0.43111041]
 [-0.03660598]
 [-0.18943866]
 [-0.5900882 ]
 [-0.07141496]
 [ 0.34181296]
 [ 0.5530154 ]]
Epoch: 15; Error: 0.4117307475895261;
Epoch: 30; Error: 0.07563208119057685;
Epoch: 45; Error: 0.04387226102452976;
Epoch: 60; Error: 0.038839866394623015;
Epoch: 75; Error: 0.03204282046919139;
Epoch: 90; Error: 0.03085450171243764;
Epoch: 105; Error: 0.02961350501074536;
Epoch: 120; Error: 0.02656667992624036;
Epoch: 135; Error: 0.025333253894044946;
Epoch: 150; Error: 0.02450498119084577 5;
Epoch: 165; Error: 0.023023529554614165;
Epoch: 180; Error: 0.022296128915630153;
Epoch: 195; Error: 0.019298073990046002;
Epoch: 210; Error: 0.016275626775981208;
```

```
Epoch: 225; Error: 0.014783805037098205;
Epoch: 240; Error: 0.014650049428199315;
Epoch: 255; Error: 0.014244047303976076;
Epoch: 270; Error: 0.013927297416481979;
Epoch: 285; Error: 0.013620518802874043;
Epoch: 300; Error: 0.013568497033586067;
Epoch: 315; Error: 0.013560645646398034;
Epoch: 330; Error: 0.013425077532912876;
Epoch: 345; Error: 0.013359895801873403;
Epoch: 360; Error: 0.01329426964589358;
Epoch: 375; Error: 0.013271968666888756;
Epoch: 390; Error: 0.01324900004495394;
Epoch: 405; Error: 0.013207881149717639;
Epoch: 420; Error: 0.013158237001113179;
Epoch: 435; Error: 0.013131582680022638;
Epoch: 450; Error: 0.013089023884041652;
Epoch: 465; Error: 0.013020358876329474;
Epoch: 480; Error: 0.012994497306822358;
Epoch: 495; Error: 0.012986000538451762;
The maximum number of train epochs is reached
[[0.30388377]]
```

In [**10**]:

```
In [11]:         'C:/Users/ujjwa/OneDrive - Centennial College/Documents/Semester_3/
Artificial_Intelligence/Assignments/Neural_Network/Assignment#5_Ujjwal/Exercise#4_Ujjwal/
Ujjwal#Ex4.py'      ='C:/Users/ujjwa/OneDrive - Centennial College/Documents/Semester_3/
Artificial_Intelligence/Assignments/Neural_Network/Assignment#5_Ujjwal/Exercise#4_Ujjwal'
[[-0.43153567 -0.36227821]
 [ 0.36089348  0.56191389]
 [-0.22389099  0.23078714]
 [ 0.45166698  0.473528  ]
 [-0.49794695 -0.55313426]
 [-0.3962035   0.453771  ]
 [-0.4819838  -0.09467085]
 [ 0.54946744  0.03979834]
 [ 0.23025254 -0.22138124]
 [ 0.22380111  0.40155081]
 [-0.57805407  0.30017318]
 [ 0.58663331  0.29779879]
 [-0.26346721  0.34713519]
 [-0.47612879 -0.06252777]
 [ 0.4903146  -0.24766302]
 [-0.25466959 -0.44396571]
 [-0.57675965  0.21460264]
 [-0.34604626 -0.28134401]
 [-0.01011221 -0.53596495]
 [ 0.08894113 -0.42392571]
 [ 0.10716664  0.23971003]
 [-0.47719869 -0.10313281]
 [ 0.23328019 -0.10298488]
 [-0.54005585  0.04307569]
 [ 0.19655357  0.01786693]
 [ 0.53351371  0.10386605]
 [ 0.4840823  -0.43503036]
 [-0.43286838  0.36886955]
 [-0.1227878  -0.40157496]
 [ 0.5130103  -0.18268097]
 [ 0.30097452  0.27119758]
 [ 0.45996731  0.14840665]
 [ 0.30113092 -0.18132199]
 [-0.27608653  0.47506346]
 [-0.08629057  0.55780806]
 [ 0.1961298   0.14603486]
 [-0.46230483  0.53938711]
 [-0.06010544  0.09406754]
 [-0.11023584 -0.31556762]
 [ 0.48405542  0.08841538]
 [-0.59655561  0.1405739 ]
 [-0.20802612  0.03246972]
 [ 0.46313052 -0.17127629]
 [ 0.49024218  0.14803214]
 [-0.58101451  0.51532468]
 [ 0.2290763   0.59678742]
 [-0.39319139 -0.4354371 ]
 [ 0.51911456  0.23618179]
 [-0.52079979  0.30655566]
 [ 0.30465143  0.50762944]
 [ 0.25382971 -0.45087485]
 [-0.57614384 -0.56854682]
 [-0.56603221 -0.30454672]
 [ 0.43203354  0.04659728]
 [ 0.06338637  0.41043707]
 [-0.45099202 -0.26497959]
 [ 0.10291113  0.5635149 ]
 [ 0.07323626 -0.57762325]
 [ 0.36075921 -0.32043087]
 [ 0.36852623 -0.13456723]
 [ 0.43625023  0.29654597]
 [ 0.06748828 -0.43625373]
 [-0.52809877 -0.45438785]
 [-0.54653775 -0.47100705]
 [-0.32914879  0.25558678]
 [ 0.07166038 -0.58493282]
 [-0.51363086  0.5607316 ]
 [ 0.08172055 -0.35604812]
 [-0.29720911  0.29259102]
 [-0.36548462  0.09763071]
 [ 0.56402399  0.41619456]
 [-0.31218269 -0.00747634]
 [ 0.14394686  0.39477708]
 [-0.41185033 -0.57770856]
 [-0.51597343 -0.01638587]
 [ 0.12759535  0.08262172]
 [-0.21916511  0.58633939]
 [ 0.09569426 -0.14383059]
 [ 0.06113786  0.29440132]
 [ 0.20307947 -0.28209653]
 [-0.5203982  -0.15589896]
 [ 0.15566101 -0.34779119]
 [ 0.30330666 -0.52015622]
 [-0.28762188  0.36570548]
 [-0.36787886  0.16735306]
 [ 0.02960437  0.50976956]
 [-0.28404388 -0.52084669]
 [ 0.28207916  0.32661364]
 [ 0.48937902  0.51836648]
 [-0.58325811 -0.3187655 ]
 [ 0.14013403  0.53881958]
 [ 0.54021134  0.06798383]
 [ 0.49872762  0.16987945]
 [-0.13199074 -0.0168112 ]
 [ 0.12517258  0.05945751]
 [ 0.51141771  0.50248012]
 [-0.12614926  0.55591503]
 [-0.3912532  -0.44840458]
 [-0.43790501  0.0067946 ]
 [-0.57417023  0.53756425]]
[[-0.79381389]
 [ 0.92280737]
 [ 0.00689615]
 [ 0.92519498]
```

```
[-1.05108121]
[ 0.05756751]
[-0.57665465]
[ 0.58926578]
[ 0.00887129]
[ 0.62535192]
[-0.27788089]
[ 0.88443209]
[ 0.08366798]
[-0.53865656]
[ 0.24265158]
[-0.69863531]
[-0.36215701]
[-0.62739027]
[-0.54607715]
[-0.33498458]
[ 0.34687668]
[-0.5803315 ]
[ 0.13029531]
[-0.49698016]
[ 0.21442051]
[ 0.63737976]
[ 0.04905194]
[-0.06399884]
[-0.52436276]
[ 0.33032933]
[ 0.57217211]
[ 0.60837396]
[ 0.11980893]
[ 0.19897693]
[ 0.47151748]
[ 0.34216466]
[ 0.07708228]
[ 0.0339621 ]
[-0.42580346]
[ 0.57247081]
[-0.45598171]
[-0.1755564 ]
[ 0.29185423]
[ 0.63827432]
[-0.06568983]
[ 0.82586372]
[-0.82862849]
[ 0.75529635]
[-0.21424413]
[ 0.81228087]
[-0.19704514]
[-1.14469066]
[-0.87057893]
[ 0.47863082]
[ 0.47382345]
[-0.71597161]
[ 0.66642602]
[-0.50438699]
[ 0.04032834]
```

```
 [ 0.23395901]
 [ 0.7327962 ]
 [-0.36876545]
 [-0.98248663]
 [-1.01754479]
 [-0.07356202]
 [-0.51327245]
 [ 0.04710073]
 [-0.27432756]
 [-0.00461808]
 [-0.26785391]
 [ 0.98021855]
 [-0.31965903]
 [ 0.53872394]
 [-0.98955888]
 [-0.53235929]
 [ 0.21021708]
 [ 0.36717428]
 [-0.04813633]
 [ 0.35553918]
 [-0.07901706]
 [-0.67629716]
 [-0.19213018]
 [-0.21684956]
 [ 0.07808359]
 [-0.2005258 ]
 [ 0.53937394]
 [-0.80489057]
 [ 0.60869279]
 [ 1.00774551]
 [-0.90202361]
 [ 0.67895361]
 [ 0.60819517]
 [ 0.66860707]
 [-0.14880194]
 [ 0.18463009]
 [ 1.01389783]
 [ 0.42976577]
 [-0.83965778]
 [-0.43111041]
 [-0.03660598]]
Epoch: 100; Error: 2.4587448057816097;
Epoch: 200; Error: 2.362878026248127;
Epoch: 300; Error: 1.706781124190482;
Epoch: 400; Error: 1.088859093619;
Epoch: 500; Error: 0.8917458788885033;
Epoch: 600; Error: 0.8031778084434016;
Epoch: 700; Error: 0.7518957882426632;
Epoch: 800; Error: 0.7220400545147698;
Epoch: 900; Error: 0.7046726094559495;
Epoch: 1000; Error: 0.694099787696228;
The maximum number of train epochs is reached
[[0.37810819]]
```

In [**12**]:

```
In [13]:         'C:/Users/ujjwa/OneDrive - Centennial College/Documents/Semester_3/
Artificial_Intelligence/Assignments/Neural_Network/Assignment#5_Ujjwal/Exercise#5_Ujjwal/
Ujjwal#Ex5.py'      ='C:/Users/ujjwa/OneDrive - Centennial College/Documents/Semester_3/
Artificial_Intelligence/Assignments/Neural_Network/Assignment#5_Ujjwal/Exercise#5_Ujjwal'
[[-0.3962035   0.453771   -0.4819838 ]
 [-0.09467085  0.54946744  0.03979834]
 [ 0.23025254 -0.22138124  0.22380111]
 [ 0.40155081 -0.57805407  0.30017318]
 [ 0.58663331  0.29779879 -0.26346721]
 [ 0.34713519 -0.47612879 -0.06252777]
 [ 0.4903146  -0.24766302 -0.25466959]
 [-0.44396571 -0.57675965  0.21460264]
 [-0.34604626 -0.28134401 -0.01011221]
 [-0.53596495  0.08894113 -0.42392571]]
[[-0.42441629]
 [ 0.49459493]
 [ 0.23267241]
 [ 0.12366992]
 [ 0.62096488]
 [-0.19152137]
 [-0.01201801]
 [-0.80612272]
 [-0.63750248]
 [-0.87094953]]
Epoch: 15; Error: 0.024155578331433863;
Epoch: 30; Error: 0.007138133345408161;
Epoch: 45; Error: 0.005890806198315972;
Epoch: 60; Error: 0.0030193437446344373;
Epoch: 75; Error: 0.0009771991658869411;
Epoch: 90; Error: 5.1174086546529595e-05;
The goal of learning is reached
[[0.43781437]]
[[ 4.51666983e-01  4.73527996e-01 -4.97946946e-01]
 [-5.53134260e-01 -3.96203497e-01  4.53771004e-01]
 [-4.81983799e-01 -9.46708500e-02  5.49467436e-01]
 [ 3.97983420e-02  2.30252537e-01 -2.21381243e-01]
 [ 2.23801113e-01  4.01550806e-01 -5.78054067e-01]
 [ 3.00173178e-01  5.86633307e-01  2.97798785e-01]
 [-2.63467210e-01  3.47135194e-01 -4.76128792e-01]
 [-6.25277686e-02  4.90314604e-01 -2.47663022e-01]
 [-2.54669594e-01 -4.43965713e-01 -5.76759651e-01]
 [ 2.14602640e-01 -3.46046261e-01 -2.81344009e-01]
 [-1.01122089e-02 -5.35964946e-01  8.89411266e-02]
 [-4.23925710e-01  1.07166644e-01  2.39710032e-01]
 [-4.77198685e-01 -1.03132815e-01  2.33280189e-01]
 [-1.02984877e-01 -5.40055849e-01  4.30756871e-02]
 [ 1.96553574e-01  1.78669345e-02  5.33513707e-01]
 [ 1.03866049e-01  4.84082298e-01 -4.35030355e-01]
 [-4.32868383e-01  3.68869546e-01 -1.22787796e-01]
 [-4.01574963e-01  5.13010296e-01 -1.82680968e-01]
 [ 3.00974524e-01  2.71197582e-01  4.59967309e-01]
 [ 1.48406648e-01  3.01130921e-01 -1.81321990e-01]
 [-2.76086530e-01  4.75063462e-01 -8.62905722e-02]
```

```
[ 5.57808057e-01  1.96129797e-01  1.46034864e-01]
[-4.62304832e-01  5.39387110e-01 -6.01054398e-02]
[ 9.40675373e-02 -1.10235837e-01 -3.15567624e-01]
[ 4.84055425e-01  8.84153840e-02 -5.96555608e-01]
[ 1.40573896e-01 -2.08026118e-01  3.24697227e-02]
[ 4.63130519e-01 -1.71276288e-01  4.90242181e-01]
[ 1.48032139e-01 -5.81014509e-01  5.15324680e-01]
[ 2.29076301e-01  5.96787421e-01 -3.93191390e-01]
[-4.35437100e-01  5.19114556e-01  2.36181794e-01]
[-5.20799793e-01  3.06555663e-01  3.04651426e-01]
[ 5.07629443e-01  2.53829710e-01 -4.50874846e-01]
[-5.76143839e-01 -5.68546816e-01 -5.66032214e-01]
[-3.04546719e-01  4.32033538e-01  4.65972772e-02]
[ 6.33863744e-02  4.10437071e-01 -4.50992022e-01]
[-2.64979585e-01  1.02911126e-01  5.63514898e-01]
[ 7.32362631e-02 -5.77623253e-01  3.60759207e-01]
[-3.20430871e-01  3.68526235e-01 -1.34567227e-01]
[ 4.36250225e-01  2.96545971e-01  6.74882808e-02]
[-4.36253729e-01 -5.28098773e-01 -4.54387853e-01]
[-5.46537746e-01 -4.71007045e-01 -3.29148794e-01]
[ 2.55586776e-01  7.16603785e-02 -5.84932824e-01]
[-5.13630864e-01  5.60731596e-01  8.17205543e-02]
[-3.56048118e-01 -2.97209107e-01  2.92591025e-01]
[-3.65484623e-01  9.76307127e-02  5.64023987e-01]
[ 4.16194562e-01 -3.12182689e-01 -7.47634288e-03]
[ 1.43946862e-01  3.94777079e-01 -4.11850326e-01]
[-5.77708557e-01 -5.15973428e-01 -1.63858669e-02]
[ 1.27595354e-01  8.26217245e-02 -2.19165109e-01]
[ 5.86339385e-01  9.56942631e-02 -1.43830593e-01]
[ 6.11378629e-02  2.94401317e-01  2.03079472e-01]
[-2.82096531e-01 -5.20398199e-01 -1.55898963e-01]
[ 1.55661008e-01 -3.47791188e-01  3.03306664e-01]
[-5.20156222e-01 -2.87621882e-01  3.65705476e-01]
[-3.67878861e-01  1.67353057e-01  2.96043709e-02]
[ 5.09769564e-01 -2.84043875e-01 -5.20846691e-01]
[ 2.82079156e-01  3.26613635e-01  4.89379023e-01]
[ 5.18366483e-01 -5.83258112e-01 -3.18765497e-01]
[ 1.40134028e-01  5.38819585e-01  5.40211343e-01]
[ 6.79838258e-02  4.98727620e-01  1.69879451e-01]
[-1.31990743e-01 -1.68111995e-02  1.25172580e-01]
[ 5.94575058e-02  5.11417712e-01  5.02480123e-01]
[-1.26149264e-01  5.55915034e-01 -3.91253200e-01]
[-4.48404577e-01 -4.37905010e-01  6.79459881e-03]
[-5.74170234e-01  5.37564253e-01  3.92538565e-01]
[-5.81977223e-01 -3.88564493e-01 -2.01523711e-01]
[-4.42803786e-01  3.71388831e-01 -1.86316017e-01]
[ 5.28128979e-01  9.84170159e-02  4.54598381e-01]
[ 4.13681334e-01  4.86470782e-01 -4.81436810e-02]
[ 5.56161792e-02  3.58324309e-01 -2.57137378e-01]
[-1.16957729e-02  1.18932369e-01 -5.81360069e-01]
[ 1.12177690e-01 -7.95883812e-02  3.68832635e-01]
[-2.21706236e-01  4.71466450e-01  9.34286583e-02]
[-3.79187758e-01  3.45515081e-01  1.34437412e-01]
[-5.35308874e-01 -9.57675840e-02  2.14882604e-01]
[ 5.02322134e-01 -5.99517570e-01  5.72110979e-01]
```

```
 [-1.48103622e-01  5.68540246e-01  1.25659321e-01]
 [ 3.94614970e-01  8.96538056e-02  1.53691438e-01]
 [-2.57308462e-01  1.04200009e-01  3.00026116e-01]
 [ 4.29976604e-01  3.06098626e-01  2.37668698e-01]
 [ 4.37375316e-01 -2.12782804e-01  2.04946549e-01]
 [-5.89512763e-02 -1.41476698e-01 -1.07026380e-01]
 [-1.18224500e-01 -2.19139265e-01  1.46303242e-01]
 [-8.37032750e-02  5.68562494e-01  2.13361070e-01]
 [-3.61716134e-01 -8.79587888e-02 -1.87984512e-01]
 [ 3.57166565e-01  4.55997946e-01  4.84610347e-01]
 [ 1.95263775e-01 -2.75750086e-01 -2.97159958e-01]
 [ 4.25877531e-01  3.32575756e-02  3.62593301e-01]
 [ 8.69862206e-02  2.79771030e-01  2.28139530e-02]
 [ 3.25060693e-01  8.26295888e-02 -4.11481457e-02]
 [-1.88773310e-01 -5.18148782e-01 -1.46490985e-01]
 [-5.04448707e-01  5.79380536e-01 -3.82064578e-01]
 [ 3.74230437e-01  4.49953974e-01  2.26095903e-01]
 [ 8.33932953e-02 -4.06834276e-01 -3.97439727e-02]
 [-1.85793539e-01 -3.29952051e-01  1.11014243e-01]
 [-2.25276195e-01  4.99566664e-01  4.91562630e-01]
 [-2.91458047e-01 -4.66930439e-01 -3.68444722e-01]
 [-4.98995185e-04  2.74302802e-01 -3.50166674e-01]
 [-3.02359730e-01  4.22006250e-01 -1.00981538e-01]
 [ 1.40022081e-01 -3.19600633e-01 -4.77639289e-01]]
[[ 0.42724803]
 [-0.49556675]
 [-0.02718721]
 [ 0.04866964]
 [ 0.04729785]
 [ 1.18460527]
 [-0.39246081]
 [ 0.18012381]
 [-1.27539496]
 [-0.41278763]
 [-0.45713603]
 [-0.07704903]
 [-0.34705131]
 [-0.59996504]
 [ 0.74793422]
 [ 0.15291799]
 [-0.18678663]
 [-0.07124564]
 [ 1.03213942]
 [ 0.26821558]
 [ 0.11268636]
 [ 0.89997272]
 [ 0.01697684]
 [-0.33173592]
 [-0.0240848 ]
 [-0.0349825 ]
 [ 0.78209641]
 [ 0.08234231]
 [ 0.43267233]
 [ 0.31985925]
 [ 0.0904073 ]
[ 0.31058431]
[-1.71072287]
[ 0.1740841 ]
[ 0.02283142]
[ 0.40144644]
[-0.14362778]
[-0.08647186]
[ 0.80028448]
[-1.41874035]
[-1.34669358]
[-0.25768567]
[ 0.12882129]
[-0.3606662 ]
[ 0.29617008]
[ 0.09653553]
[ 0.12687362]
[-1.11006785]
[-0.00894803]
[ 0.53820306]
[ 0.55861865]
[-0.95839369]
[ 0.11117648]
[-0.44207263]
[-0.17092143]
[-0.295121  ]
[ 1.09807181]
[-0.38365713]
[ 1.21916496]
[ 0.7365909 ]
[-0.02362936]
[ 1.07335534]
[ 0.03851257]
[-0.87951499]
[ 0.35593259]
[-1.17206543]
[-0.25773097]
[ 1.08114438]
[ 0.85200844]
[ 0.15680311]
[-0.47412347]
[ 0.40142194]
[ 0.34318887]
[ 0.10076473]
[-0.41619385]
[ 0.47491554]
[ 0.54609594]
[ 0.63796021]
[ 0.14691766]
[ 0.97374393]
[ 0.42953906]
[-0.30745435]
[-0.19106052]
[ 0.69822029]
[-0.63765943]
[ 1.29777486]
```

```
 [-0.37764627]
 [ 0.82172841]
 [ 0.3895712 ]
 [ 0.36654214]
 [-0.85341308]
 [-0.30713275]
 [ 1.05028031]
 [-0.36318495]
 [-0.40473135]
 [ 0.7658531 ]
 [-1.12683321]
 [-0.07636287]
 [ 0.01866498]
 [-0.65721784]]
Epoch: 100; Error: 1.9116650975992675;
Epoch: 200; Error: 1.415138889792522;
Epoch: 300; Error: 1.1010837218393295;
Epoch: 400; Error: 0.9538536066178438;
Epoch: 500; Error: 0.8795249083076757;
Epoch: 600; Error: 2.974644225207735;
Epoch: 700; Error: 1.2454030287682334;
Epoch: 800; Error: 1.2172531222541405;
Epoch: 900; Error: 1.2076483510500398;
Epoch: 1000; Error: 1.203793698234653;
The maximum number of train epochs is reached
[[0.59333749]]
```

In [**14**]: