

Лабораторна робота №6

з дисципліни **Основи штучного інтелекту**
студента групи **ЗІПЗк-22-1**

Перехватова Алевтина Олександрівна

дата виконання: 06.12.2023

Мета роботи: : використовуючи спеціалізовані бібліотеки та мову програмування Python навчитися дослідити деякі типи нейронних мереж.

Виконання роботи

Завдання 2.2

Дія1:

```
!pip install neurolab
import neurolab as nl
import numpy as np
# Створення мого сигналу для навчання
i1 = np.sin(np.arange(0, 20))
i2 = np.sin(np.arange(0, 20)) * 2

t1 = np.ones([1, 20])
t2 = np.ones([1, 20]) * 2

input = np.array([i1, i2, i1, i2]).reshape(20 * 4, 1)
target = np.array([t1, t2, t1, t2]).reshape(20 * 4, 1)
# Створення мережі з 2 прошарками
net = nl.net.newelm([-2, 2], [10, 1], [nl.trans.TanSig(), nl.trans.PureLin()])
# Ініціалізуйте початкові функції вагів
net.layers[0].initf = nl.init.InitRand([-0.1, 0.1], 'wb')
net.layers[1].initf = nl.init.InitRand([-0.1, 0.1], 'wb')
net.init()
# Тренування мережі
error = net.train(input, target, epochs=500, show=100, goal=0.01)
# Запустіть мережу
output = net.sim(input)
# Побудова графіків
import pylab as pl
pl.subplot(211)
pl.plot(error)
pl.xlabel('Epoch number')
pl.ylabel('Train error (default MSE)')

pl.subplot(212)
pl.plot(target.reshape(80))
pl.plot(output.reshape(80))
pl.legend(['train target', 'net output'])
pl.show()
```

Рис.1.1– Код програми

					ЗІПЗк-22-1	Арк.
						1
Змін.	Арк.	№ докум.	Підпис	Дата		

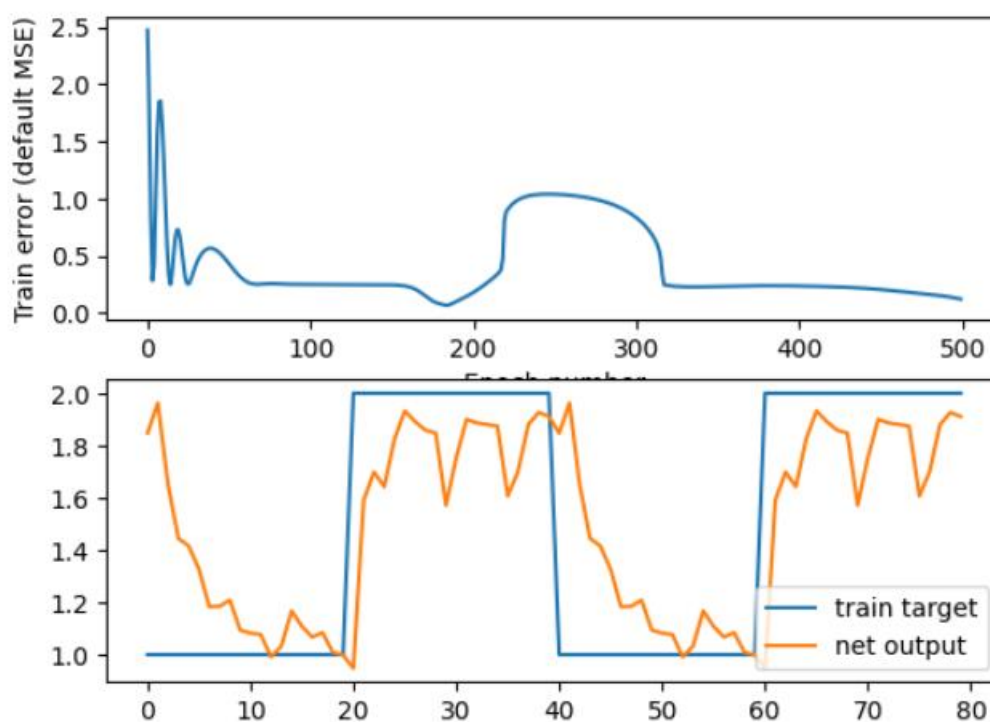


Рис.1.2 – Реакція програми на дію

Завдання 2.3

Дія1:

```
import numpy as np
!pip install neurolab
import neurolab as nl

target = [[-1, 1, -1, -1, 1, -1, -1, 1, -1],
          [1, 1, 1, 1, -1, 1, 1, -1, 1],
          [1, -1, 1, 1, 1, 1, 1, -1, 1],
          [1, 1, 1, 1, -1, -1, 1, -1, -1],
          [-1, -1, -1, -1, 1, -1, -1, -1, -1]]

input = [[-1, -1, 1, 1, 1, 1, 1, -1, 1],
         [-1, -1, 1, -1, 1, -1, -1, -1, -1],
         [-1, -1, -1, -1, 1, -1, -1, 1, -1]]

# Створення та тренування нейромережі
net = nl.net.newhem(target)

output = net.sim(target)
print("Test on train samples (must be [0, 1, 2, 3, 4])")
print(np.argmax(output, axis=0))

output = net.sim([input[0]])
print("Outputs on recurent cycle:")
print(np.array(net.layers[1].outs))

output = net.sim(input)
print("Outputs on test sample:")
print(output)
```

Рис.1.3– Код програми

					ЗІПЗк-22-1	Арк.
						2
Змін.	Арк.	№ докум.	Підпис	Дата		

```
Test on train samples (must be [0, 1, 2, 3, 4])
[0 1 2 3 4]
Outputs on recurrent cycle:
[[0.      0.24    0.48    0.      0.      ]
 [0.      0.144   0.432   0.      0.      ]
 [0.      0.0576  0.4032  0.      0.      ]
 [0.      0.      0.39168  0.      0.      ]]
Outputs on test sample:
[[0.      0.      0.39168  0.      0.      ]
 [0.      0.      0.      0.      0.39168  ]
 [0.07516193 0.      0.      0.      0.07516193]]
```

Рис.1.4 – Реакція програми на дію

Завдання 2.4

Дія1:

```
import numpy as np
!pip install neurolab
import neurolab as nl

# N E R O

target = [[1,0,0,0,1,
           1,1,0,0,1,
           1,0,1,0,1,
           1,0,0,1,1,
           1,0,0,0,1],
          [1,1,1,1,1,
           1,0,0,0,0,
           1,1,1,1,1,
           1,0,0,0,0,
           1,1,1,1,1],
          [1,1,1,1,0,
           1,0,0,0,1,
           1,1,1,1,0,
           1,0,0,1,0,
           1,0,0,0,1],
          [0,1,1,1,0,
           1,0,0,0,1,
           1,0,0,0,1,
           1,0,0,0,1,
           0,1,1,1,0]]

chars = ['N', 'E', 'R', 'O']
target = np.asarray(target)
target[target == 0] = -1
# Create and train network
net = nl.net.newhop(target)

output = net.sim(target)
print("Test on train samples:")
for i in range(len(target)):
    print(chars[i], (output[i] == target[i]).all())
print("\nTest on defaced N:")
test = np.asarray([0,0,0,0,0,
                   1,1,0,0,1,
                   1,1,0,0,1,
                   1,0,1,1,1,
                   0,0,0,1,1])
test[test==0] = -1
out = net.sim([test])
print ((out[0] == target[0]).all(), 'Sim. steps', len(net.layers[0].outs))
```

Рис.1.5– Код програми

					<div style="text-align: center;"> <i>ЗПЗк-22-1</i> </div>	Арк.
						3
<i>Змін.</i>	<i>Арк.</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>		

```

Test on train samples:
N True
E True
R True
O True

```

```

Test on defaced N:
True Sim. steps 2

```

Рис.1.6 – Реакція програми на дію

Завдання 2.5

Дія1:

```

import numpy as np
!pip install neurolab
import neurolab as nl

# Matrices for each letter
П = np.asfarray([[1, 1, 1],
                  [1, 0, 0],
                  [1, 1, 1],
                  [1, 0, 0],
                  [1, 0, 0]])

А = np.asfarray([[0, 1, 0],
                  [1, 0, 1],
                  [1, 1, 1],
                  [1, 0, 1],
                  [1, 0, 1]])

О = np.asfarray([[0, 1, 0],
                  [1, 0, 1],
                  [1, 0, 1],
                  [1, 0, 1],
                  [0, 1, 0]])

# Combine matrices and characters
target_name = np.vstack((П.flatten(), А.flatten(), О.flatten()))

# Convert 0s to -1s
target_name[target_name == 0] = -1

# Create and train network
net_name = nl.net.newhop(target_name)

# Test on each letter
chars = ['П', 'А', 'О']
for i, letter in enumerate([П, А, О]):
    letter_test = letter.flatten()
    letter_test[letter_test == 0] = -1
    output = net_name.sim([letter_test])
    print(f"Test on {chars[i]}:", (output[0] == target_name[i]).all())

# Test with errors in one letter (e.g., flipping a pixel)
П_noisy = П.flatten()
П_noisy[3] = -П_noisy[3] # Flip one pixel
output_noisy = net_name.sim([П_noisy])
print("Test on defaced П:", (output_noisy[0] == target_name[0]).all())

```

Рис.1.7– Код програми

```

Test on П: True
Test on А: False
Test on О: True
Test on defaced П: True

```

Рис.1.8 – Реакція програми на дію

					ЗІПЗк-22-1	Арк.
						4
Змін.	Арк.	№ докум.	Підпис	Дата		