

# ЛАБОРАТОРНА РОБОТА № 6

## ДОСЛІДЖЕННЯ РЕКУРЕНТНИХ НЕЙРОННИХ МЕРЕЖ

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

**Хід роботи:**

### Завдання 2.1. Ознайомлення з Рекурентними нейронними мережами

```
class RNN:
    # A many-to-one Vanilla Recurrent Neural Network.

    def __init__(self, input_size, output_size, hidden_size=64):
        # Weights
        self.Whh = randn(hidden_size, hidden_size) / 1000
        self.Wxh = randn(hidden_size, input_size) / 1000
        self.Why = randn(output_size, hidden_size) / 1000

        # Biases
        self.bh = np.zeros((hidden_size, 1))
        self.by = np.zeros((output_size, 1))

    2 usages
    def forward(self, inputs):
        """
        Perform a forward pass of the RNN using the given inputs.
        Returns the final output and hidden state.
        - inputs is an array of one hot vectors with shape (input_size, 1).
        """
        h = np.zeros((self.Whh.shape[0], 1))

        self.last_inputs = inputs
        self.last_hs = {0: h}

        # Perform each step of the RNN
        for i, x in enumerate(inputs):
            h = np.tanh(self.Wxh @ x + self.Whh @ h + self.bh)
            self.last_hs[i + 1] = h

        # Compute the output
        y = self.Why @ h + self.by
```

Рис. 1. Неповне відображення коду програми.

|           |      |             |        |      |   |                    |         |
|-----------|------|-------------|--------|------|---|--------------------|---------|
|           |      |             |        |      | ЖИТОМИРСЬКА ПОЛІТЕХНІКА.23.125.08.000 – Лр6 |                    |         |
| Зм        | Арк. | № докум.    | Підпис | Дата |   |                    |         |
| Розроб.   |      | Куліш М.В.  |        |      | Звіт з лабораторної роботи №6               | Літ.               | Арк.    |
| Перевір.  |      | Пулеко І.В. |        |      |   |                    | Акрушів |
| Реценз.   |      |             |        |      |   |                    | 1       |
| Н. Контр. |      |             |        |      |   | ФІКТ, гр. КБм-22-1 |         |
| Зав.каф.  |      |             |        |      |   |                    |         |
|           |      |             |        |      |   |                    | 8       |

```

/home/xtr99/labs/ai/lab06/venv/bin/python /home/xtr99/labs/ai/lab06/LR_6_task_1.py
18 unique words found
--- Epoch 100
Train: Loss 0.689 | Accuracy: 0.552
Test: Loss 0.699 | Accuracy: 0.500
--- Epoch 200
Train: Loss 0.661 | Accuracy: 0.655
Test: Loss 0.733 | Accuracy: 0.650
--- Epoch 300
Train: Loss 0.244 | Accuracy: 0.914
Test: Loss 0.103 | Accuracy: 0.950
--- Epoch 400
Train: Loss 0.013 | Accuracy: 1.000
Test: Loss 0.012 | Accuracy: 1.000
--- Epoch 500
Train: Loss 0.005 | Accuracy: 1.000
Test: Loss 0.004 | Accuracy: 1.000
--- Epoch 600
Train: Loss 0.003 | Accuracy: 1.000
Test: Loss 0.003 | Accuracy: 1.000
--- Epoch 700
Train: Loss 0.002 | Accuracy: 1.000
Test: Loss 0.002 | Accuracy: 1.000
--- Epoch 800
Train: Loss 0.001 | Accuracy: 1.000
Test: Loss 0.001 | Accuracy: 1.000
--- Epoch 900
Train: Loss 0.001 | Accuracy: 1.000
Test: Loss 0.001 | Accuracy: 1.000
--- Epoch 1000
Train: Loss 0.001 | Accuracy: 1.000
Test: Loss 0.001 | Accuracy: 1.000

```

Рис. 2. Результат виконання.

Висновок: після тренування моделі на протязі 1000 епох було отримано дуже високі показники точності та дуже низькі значення втрати. Це свідчить про те, що модель ефективно навчилася класифікувати дані. В результаті, можна зробити висновок, що модель є дуже успішною у виконанні своєї задачі.

## Завдання 2.2. Дослідження рекурентної нейронної мережі Елмана(Elman Recurrent network (newelm))

|      |      |             |        |      |   |      |
|------|------|-------------|--------|------|---|------|
|      |      | Куліш М.В.  |        |      | ЖИТОМИРСЬКА ПОЛІТЕХНІКА.23.125.08.000 - Лр6 | Арк. |
|      |      | Пулеко І.В. |        |      |   | 2    |
| Змн. | Арк. | № докум.    | Підпис | Дата |   |      |

```

1  import ...
3
4  i1 = np.sin(np.arange(0, 20))
5  i2 = np.sin(np.arange(0, 20)) * 2
6  t1 = np.ones([1, 20])
7  t2 = np.ones([1, 20]) * 2
8  input = np.array([i1, i2, i1, i2]).reshape(20 * 4, 1)
9  target = np.array([t1, t2, t1, t2]).reshape(20 * 4, 1)
10 net = nl.net.newelm([-2, 2], [10, 1], [nl.trans.TanSig(), nl.trans.PureLin()])
11 net.layers[0].initf = nl.init.InitRand([-0.1, 0.1], 'wb')
12 net.layers[1].initf = nl.init.InitRand([-0.1, 0.1], 'wb')
13 net.init()
14 # Тренування мережі
15 error = net.train(input, target, epochs=500, show=100, goal=0.01)
16 # Запустити мережу
17 output = net.sim(input)
18 # Побудова графіків
19 from pylab import ylabel, plot, legend, show, xlabel, subplot
20 subplot(211)
21 plot(error)
22 xlabel('Epoch number')
23 ylabel('Train error (default MSE)')
24 subplot(212)
25 plot(target.reshape(80))
26 plot(output.reshape(80))
27 legend(['train target', 'net output'])
28 show()

```

Рис. 3. Код програми.

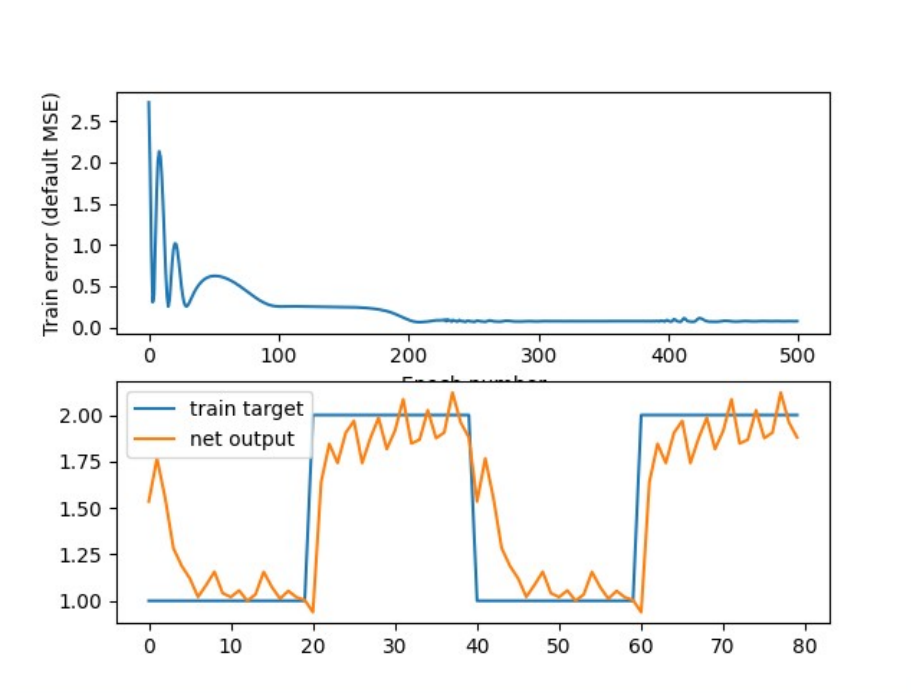


Рис. 4. Результат побудованих графіків.

|      |      |             |        |      |   |      |
|------|------|-------------|--------|------|---|------|
|      |      | Куліш М.В.  |        |      | ЖИТОМИРСЬКА ПОЛІТЕХНІКА.23.125.08.000 - Лр6 | Арк. |
|      |      | Пулеко І.В. |        |      |   | 3    |
| Змн. | Арк. | № докум.    | Підпис | Дата |   |      |

```

/home/xtr99/labs/ai/lab06/venv/bin/python /home/xtr99/labs/ai/lab06/LR_6_task_2.py
Epoch: 100; Error: 0.252488874137869;
Epoch: 200; Error: 0.09547999656718961;
Epoch: 300; Error: 0.07051769352876044;
Epoch: 400; Error: 0.08608080298488426;
Epoch: 500; Error: 0.07370317892728614;
The maximum number of train epochs is reached

Process finished with exit code 0

```

Рис. 5. Результат виконання.

Висновок: під час тренування моделі помилка (error) поступово зменшувалась протягом 500 епох. Однак, досягнувши максимальної кількості епох, модель не змогла далі покращувати свою продуктивність. Це може свідчити про досягнення плато, коли модель вже навчилась максимально можливу рівню. Детальніший аналіз та оцінка ефективності моделі можуть бути проведені на основі додаткових показників якості та порівняння з іншими моделями.

### Завдання 2.3. Дослідження нейронної мережі Хемінга (Hemming Recurrent network)

```

LR_6_task_3.py
1 import numpy as np
2 import neurolab as nl
3
4 target = [[-1, 1, -1, -1, 1, -1, -1, 1, -1],
5           [1, 1, 1, 1, -1, 1, 1, -1, 1],
6           [1, -1, 1, 1, 1, 1, 1, -1, 1],
7           [1, 1, 1, 1, -1, -1, 1, -1, -1],
8           [-1, -1, -1, -1, 1, -1, -1, -1, -1]]
9
10 input = [[-1, -1, 1, 1, 1, 1, 1, -1, 1],
11          [-1, -1, 1, -1, 1, -1, -1, -1, -1],
12          [-1, -1, -1, -1, 1, -1, -1, 1, -1]]
13
14 # Створення та тренування нейромережі
15 net = nl.net.newhem(target)
16
17 output = net.sim(target)
18 print("Test on train samples (must be [0, 1, 2, 3, 4])")
19 print(np.argmax(output, axis=0))
20
21 output = net.sim([input[0]])
22 print("Outputs on recurrent cycle:")
23 print(np.array(net.layers[1].outs))
24
25 output = net.sim(input)
26 print("Outputs on test sample:")
27 print(output)

```

Рис. 6. Код програми.

|      |      |             |        |      |   |      |
|------|------|-------------|--------|------|---|------|
|      |      | Куліш М.В.  |        |      | ЖИТОМИРСЬКА ПОЛІТЕХНІКА.23.125.08.000 - Лр6 | Арк. |
|      |      | Пулеко І.В. |        |      |   | 4    |
| Змн. | Арк. | № докум.    | Підпис | Дата |   |      |

```

LR_6_task_3 x
/home/xtr99/labs/ai/lab06/venv/bin/python /home/xtr99/labs/ai/lab06/LR_6_task_3.py
Test on train samples (must be [0, 1, 2, 3, 4])
[0 1 2 3 4]
Outputs on recurent 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. 7. Результат виконання.

## Завдання 2.4. Дослідження рекурентної нейронної мережі ХопфілдаHopfield Recurrent network (newhop)

```

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
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())

```

Рис. 8. Неповне відображення коду програми.

|      |      |             |        |      |   |      |
|------|------|-------------|--------|------|---|------|
|      |      | Куліш М.В.  |        |      | ЖИТОМИРСЬКА ПОЛІТЕХНІКА.23.125.08.000 - Лр6 | Арк. |
|      |      | Пулеко І.В. |        |      |   | 5    |
| Змн. | Арк. | № докум.    | Підпис | Дата |   |      |

```

/home/xtr99/labs/ai/lab06/venv/bin/python /home/xtr99/labs/ai/lab06/LR_6_task_4.py
Test on train samples:
N True
E True
R True
O True

Test on defaced M:|
False Sim. steps 3

```

Рис. 9. Результат виконання.

Висновок: модель добре справляється з класифікацією на навчальних прикладах, але має проблему з класифікацією пошкоджених зразків. Це може бути наслідком недостатньої роботи моделі з варіаціями та змінами у вхідних даних. Для подальшого вдосконалення моделі можуть бути необхідні додаткові заходи, такі як збільшення обсягу тренувальних даних, розширення набору ознак або використання більш складних алгоритмів класифікації.

## Завдання 2.5. Дослідження рекурентної нейронної мережі Хопфілда для ваших персональних даних

|      |      |             |        |      |   |      |
|------|------|-------------|--------|------|---|------|
|      |      | Куліш М.В.  |        |      | ЖИТОМИРСЬКА ПОЛІТЕХНІКА.23.125.08.000 - Лр6 | Арк. |
|      |      | Пулеко І.В. |        |      |   | 6    |
| Змн. | Арк. | № докум.    | Підпис | Дата |   |      |



```

LR_6_task_5.py x
4 target = [[1, 0, 0, 0, 1,
5           1, 0, 0, 1, 0,
6           1, 0, 1, 0, 0,
7           1, 0, 0, 1, 0,
8           1, 0, 0, 0, 1],
9          [1, 0, 0, 0, 1,
10         1, 1, 0, 1, 1,
11         1, 0, 1, 0, 1,
12         1, 0, 0, 0, 1,
13         1, 0, 0, 0, 1],
14         [1, 1, 1, 1, 0,
15         1, 0, 0, 0, 1,
16         1, 1, 1, 1, 0,
17         1, 0, 0, 0, 1,
18         1, 1, 1, 1, 0]
19
20     ]
21     chars = ['K', 'M', 'B']
22     target = np.asfarray(target)
23     target[target == 0] = -1
24     net = nl.net.newhop(target)
25     output = net.sim(target)
26     print("Test on train samples:")
27     for i in range(len(target)):
28         print(chars[i], (output[i] == target[i]).all())
29
30     print("\nTest on defaced K:")
31     test = np.asfarray([1, 0, 1, 0, 1,
32                        1, 0, 0, 1, 0,
33                        1, 0, 1, 0, 0,
34                        1, 0, 0, 1, 0,
35                        1, 1, 0, 1, 1], )

```

Рис. 10. Неповне відображення коду програми.

|      |      |             |        |      |   |      |
|------|------|-------------|--------|------|---|------|
|      |      | Куліш М.В.  |        |      | ЖИТОМИРСЬКА ПОЛІТЕХНІКА.23.125.08.000 - Лр6 | Арк. |
|      |      | Пулеко І.В. |        |      |   | 7    |
| Змн. | Арк. | № докум.    | Підпис | Дата |   |      |

```

n: LR_6_task_5 x
/home/xtr99/labs/ai/lab06/venv/bin/python /home/xtr99/labs/ai/lab06/LR_6_task_5.py
Test on train samples:
K True
M True
B True

Test on defaced K:
True Sim. steps 1

Process finished with exit code 0

```

Рис. 11. Результат виконання.

За результатами тестування, матриця доступу точно відображає літери К, М, В. Літера К була правильно визначена як у тренувальних прикладах, так і при тестуванні на пошкоджену версію літери. Результати тестування свідчать про те, що матриця доступу ефективно відтворює характерний зразок літери К і дозволяє його надійне розпізнавання. Тому, матрицю доступу можна вважати успішним кодуванням для літери К.

#### Покликання на github:

<https://github.com/mrkulish/ai-labs/tree/bd19a56a03d6e4fd2b2ea83d218a88e530b02f83/lab06>

**Висновок:** під час виконання завдань лабораторної роботи навчилися працювати з використанням спеціалізованих бібліотек та мови програмування Python було досліджено методи регресії та неконтрольованої класифікації даних у машинному навчанні.

|      |      |             |        |      |   |      |
|------|------|-------------|--------|------|---|------|
|      |      | Куліш М.В.  |        |      | ЖИТОМИРСЬКА ПОЛІТЕХНІКА.23.125.08.000 - Лр6 | Арк. |
|      |      | Пулеко І.В. |        |      |   | 8    |
| Змн. | Арк. | № докум.    | Підпис | Дата |   |      |