

# **КОГНИТИВНЕ МОДЕЛЮВАННЯ**

**122 «Комп'ютерні науки»**

**КНМ-21**

**2021 / 2022 навчальний рік**

# КУРСОВИЙ ПРОЕКТ

# ТЕМАТИКА КУРСОВОГО ПРОЕКТУ

Розробка програмного модуля на довільній мові (переважно Python), що реалізує штучну нейронну мережу для розпізнавання заданого класу графічних візуальних образів.

## ЗАГАЛЬНЕ ЗАВДАННЯ

спроектувати і розробити програмний модуль штучної нейронної мережі розпізнавання визначеного класу зображень та дослідити його роботу

# Обозначения

- $k$  – эпоха (опустим для упрощения),
- $L$  – количество слоев (у нас 3 (для начала)),
- $l$  – номер слоя,  $l=0,1,2$ .
- $N^{(l)}$  – количество нейронов в  $l$ -ом слое,
- $i$  – номер нейрона в  $(l-1)$ -ом слое,  
 $i=1,\dots,N^{(l-1)}$
- $j$  – номер нейрона в  $(l)$ -ом слое,
- $w_{ij}^{(l)}$  – вес связи  $i$ -го нейрона в  $(l-1)$ -ом слое с  $j$ -м нейроном  $l$ -го слоя,
- $b_j^{(l)} = w_j^{(l)}$  – смещение  $j$ -го нейрона  $l$ -го слоя на (вес связи к 1-ому входу),
- $\{x_m, d_m\}$  – пары (вход, выход) обучающих векторов,  $m$  - номер пары,  $m = 1,\dots,P$ ,
- $P$  - количество обучающих пар.

# Алгоритм

**Step 1. (первая эпоха).** Задаем все пороги и веса равномерно распределенные на интервале (0, 1) или (-0.5 +0.5)

**Step 2.** Задаем первую обучающую пару  $m=1 \{x_1, d_1\}$ .

**Step 3. Прямой ход (RUN).** Вычисление выходных сигналов каждого нейрона каждого слоя.

$$y_i^{(0)} = x_m$$

$$l = 1: s_j^{(1)} = \sum_{i=0}^{N^{(0)}} w_{ij}^{(1)} * y_i^{(0)}; y_j^{(1)} = f(s_j^{(1)})$$

$$l = 2: s_j^{(2)} = \sum_{i=0}^{N^{(1)}} w_{ij}^{(2)} * y_i^{(1)}; y_j^{(2)} = f(s_j^{(2)})$$

# Алгоритм

## Step 4. Вычисление ошибки

$$e_j = y_j^{(2)} - d_{mj}.$$

$$E = 0.5 \sum_{j=1}^{N^{(2)}} e_j^2,$$

## Step 5. Обратное распространение. Настраиваем синаптические веса

$$w[new]_{ij}^{(l)} = w[old]_{ij}^{(l)} + \Delta_{ij}^{(l)}; \Delta_{ij}^{(l)} = -\eta * \frac{\partial E}{\partial w_{ij}^{(l)}}$$

$\eta$  - параметр, задающий скорость обучения (спуска по антиградиенту),  $0 < \eta < 1$ .

# Алгоритм

## Step 6. Проверка условия завершения

- если все пары не перебрали, и то  $m=m+1$ , на step 3.
- если перебрали все пары
- и  $\frac{1}{P} \sum_1^P E_m > \varepsilon$ , то  $k=k+1$  и на step 2 (следующая эпоха)
- если перебрали все пары
- и  $\frac{1}{P} \sum_1^P E_m < \varepsilon$ , то stop обучения.

# ОПИСАНИЕ СЕТИ *СЛОВАРЬ СПИСКОВ*

```
NET = { 'struct':Net_layers,  
        'weight':nn_weghts,  
        'sums': S,  
        'outs':Y,  
        'error':E }
```

**'struct':layers** --> список структуры сети.

**'weight': weights\_list** --> список МАТРИЦ весов каждого слоя.  
МАТРИЦА для КАЖДОГО нейрона слоя содержит вектор весов,  
размерность = количество нейронов  $L-1$  слоя + 1 (для смещения).

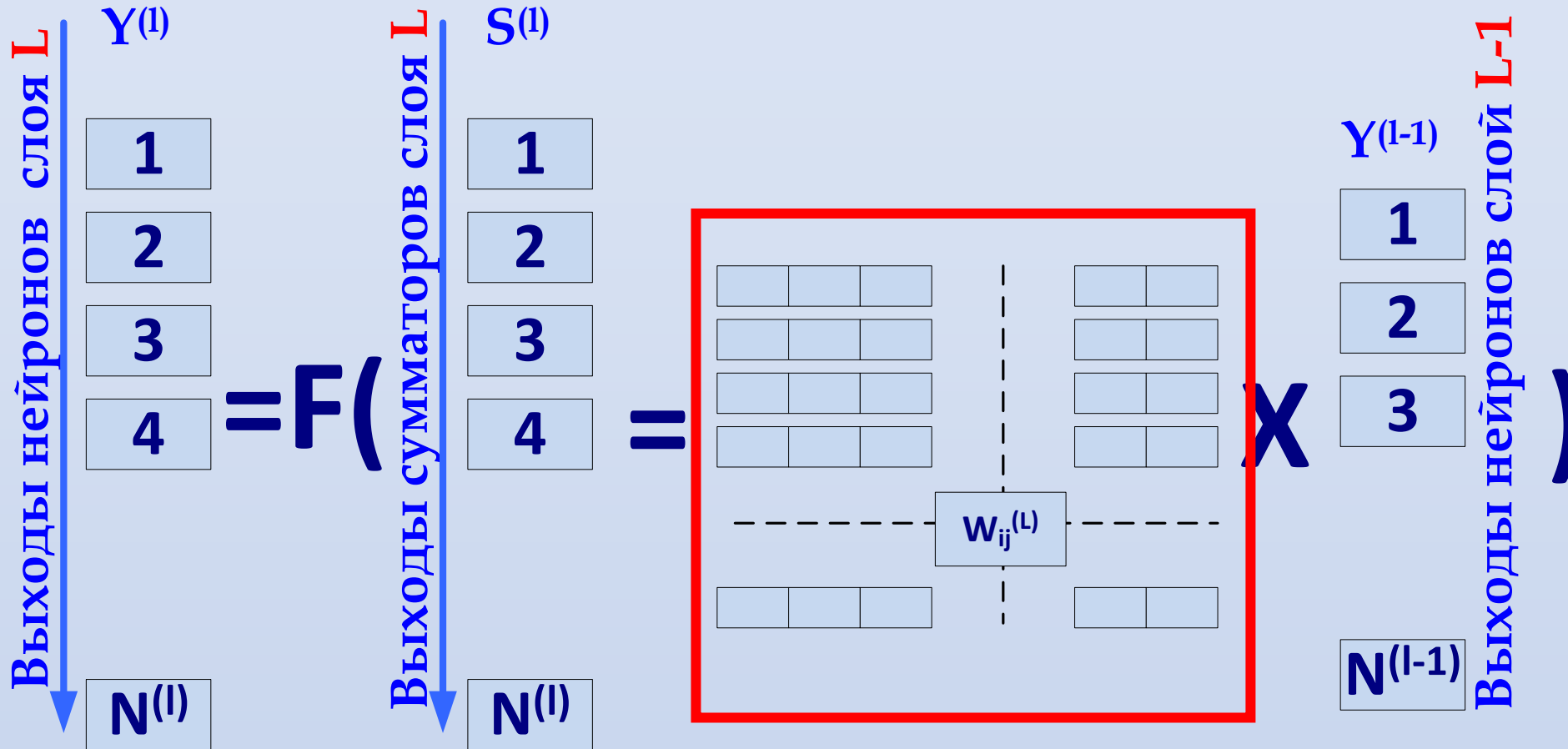
**'outs': outs\_list** --> список ВЕКТОРОВ (! векторов) выходов  
нейронов каждого слоя.

**'sums': sums\_list** --> список ВЕКТОРОВ (! векторов) входов  
функций активации нейронов каждого слоя

**'error': error\_list** --> список ВЕКТОРОВ (! векторов) ошибок  
выходов нейронов каждого слоя



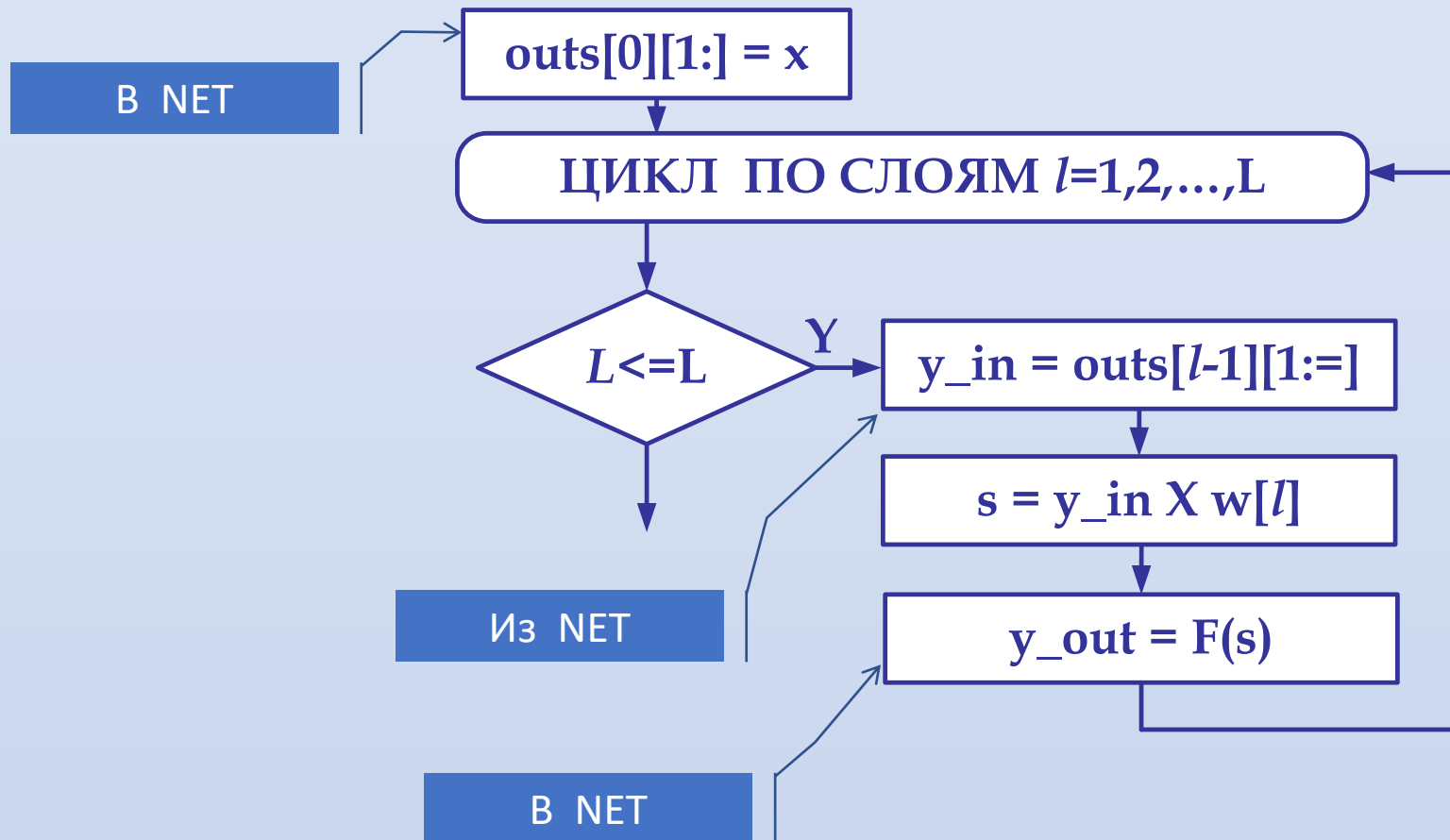
# Формирование выходов нейронов



!!! Для каждого элемента списка (СЛОЯ!)

Net\_Layers = [IN , HID1, HID2, OUT]

# Run Forward



# ЕТАПИ

- Архитектура та параметри нейронної мережі.
- Формати змінних: **Data set / Train / Test Data, W, S, Y**
- Файли **Data set**
- Функція **RUN**
- Функція **ERR**

# ПОТОЧНЕ ЗАВДАННЯ

- Розробити та налаштувати функцію формування нейронної мережі **NET**.
- Розробити та налаштувати програмний модуль **RUN()**.
- Розробити та налаштувати програмний модуль **ERR()**.

# End Curs Work 3