

МОДУЛЬ №2
ДИСЦИПЛІНА: «ОБЧИСЛЮВАЛЬНИЙ ІНТЕЛЕКТ»
Ступінь вищої освіти — магістр
Галузь знань — 12 “Інформаційні технології”
Спеціальність — 122 “Комп’ютерні науки”
Освітня програма — “Комп’ютерні науки”

Тема: Дослідження класифікатора на основі нейронних мереж прямого поширення (FeedFoward Neural Networks).

Мета роботи: Дослідження можливостей роботи нейронних мереж прямого поширення для класифікації зображень.

1. ТЕОРЕТИЧНІ ВІДОМОСТІ

Використання нейромережевого класифікатора є одним із найпоширеніших і найпотужніших методів штучного інтелекту. Для дослідження можливостей даного класифікатора потрібно:

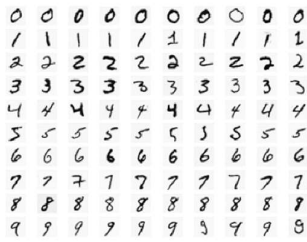
- познайомитись із принципами роботи штучного нейрона (див. презентацію);
- познайомитись із принципами будови багатошарових нейронних мереж прямого поширення (див. презентацію);
- познайомитись із роботою нейронної мережі в прямому поширенні сигналів (forward) - (див. презентацію);
- познайомитись із основами навчання нейронної мережі та її роботою в зворотному поширенні похибок(backforward) - (див. презентацію);
- вивчити можливості по застосуванні нейронної мережі для розв’язку задач класифікації із використанням середовище *JupyterNotebook/JupyterLab* із пакету *Anaconda* (у файлі «Lecture_MLP.ipynb» наведено приклад реалізації нейронної мережі для задачі класифікації).

2 ЗАВДАННЯ ДЛЯ ВИКОНАННЯ МОДУЛЯ

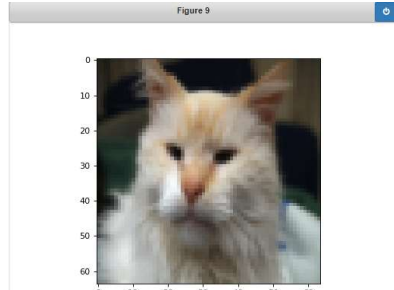
(виконання завдань здійснюється у *JupyterNotebook/JupyterLab* чи *Python*)

Для виконання модульної роботи необхідно дослідити роботу нейронної мережі з різною архітектурою, для чого

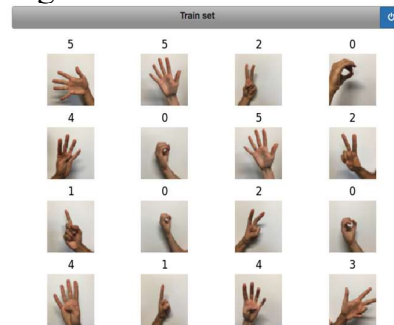
1. Завантажити вхідні дані згідно варіанту (порядковим номером у журналі групи розбивши список на тріади) обираючи один із трьох можливих наборів даних для задачі класифікації. Файли розміщуються у каталозі “../data/”
 - Handwritten Digits data set



– Cat vs Noncat data set



– Signs data set



2. Забезпечте розділення існуючих даних на навчальну і тестові вибірки, а також проведіть нормалізацію
3. Створіть структури нейронних мереж:
 - одношарову (повинна містити 100 нейронів)
 - двошарову (із розподілом нейронів [3,3])
 - тришарову (із розподілом нейронів [20,7,10])
4. Встановіть параметри навчання нейронних мереж:
 - метод навчання ({'lbfgs', 'sgd', 'adam'}, default='adam');
 - функцію активації нейронів ({'identity', 'logistic', 'tanh', 'relu'}, default='relu')
 - кількість епох для навчання (max_iter : int, default=200);
 - параметр регуляризації «alpha» (float, default=0.0001).
5. Здійсніть навчання нейронних мереж та дослідіть точність навчальних і тренувальних наборів даних, обравши найкращий випадок на множині параметрів навчання (п.4) нейронних мереж з різними архітектурами (п.3) згідно варіанту.
6. Порівняйте результати точності класифікації із результатами методу SupportVectorMachine (SVM).
7. Для виконання завдання використовуйте допоміжний файл «Module2_task_MLP.ipynb» у *JupyterNotebook/JupyterLab* із пакету *Anaconda* або у *Python*

3. ЗАВАНТАЖИТИ РЕЗУЛЬТАТИ ВИКОНАННЯ ПОСТАВЛЕНОГО ЗАВДАННЯ:

- Приєднати файл із звітом про виконанні завдання у СДН Moodle.
- Завантажити файли із результатами виконання завдання у *JupyterNotebook/JupyterLab* у github (<https://github.com/>) та приєднати лінк у СДН Moodle (вимагається створення свого профілю у github).

Варіанти завдань:

Вар	Data Set	1-Layer			2-Layers [3,3]			3-Layers [20,7,10]		
		solver	activations	max_iter	solver	activations	max_iter	solver	activations	max_iter
1.	Handwritten Digits	'lbfgs'	'identity'	200	'lbfgs'	('logistic', 'tanh')	200	'lbfgs'	('logistic', 'tanh', 'relu')	200
2.	Cat vs Noncat	'sgd'	'logistic'	300	'sgd'	('logistic', 'relu')	300	'sgd'	('relu', 'logistic', 'relu')	300
3.	Signs	'adam'	'tanh'	100	'adam'	('tanh', 'relu')	100	'adam'	('tanh', 'relu', 'relu')	100
4.	Handwritten Digits	'adam'	'relu'	500	'adam'	('identity', 'relu')	500	'adam'	('identity', 'relu', 'relu')	500
5.	Cat vs Noncat	'lbfgs'	'identity'	200	'lbfgs'	('logistic', 'tanh')	200	'lbfgs'	('logistic', 'tanh', 'relu')	200
6.	Signs	'sgd'	'logistic'	300	'sgd'	('logistic', 'relu')	300	'sgd'	('relu', 'logistic', 'relu')	300
7.	Handwritten Digits	'adam'	'tanh'	100	'adam'	('tanh', 'relu')	100	'adam'	('tanh', 'relu', 'relu')	100
8.	Cat vs Noncat	'adam'	'relu'	500	'adam'	('identity', 'relu')	500	'adam'	('identity', 'relu', 'relu')	500
9.	Signs	'lbfgs'	'identity'	200	'lbfgs'	('logistic', 'tanh')	200	'lbfgs'	('logistic', 'tanh', 'relu')	200
10.	Handwritten Digits	'sgd'	'logistic'	300	'sgd'	('logistic', 'relu')	300	'sgd'	('relu', 'logistic', 'relu')	300
11.	Cat vs Noncat	'adam'	'tanh'	100	'adam'	('tanh', 'relu')	100	'adam'	('tanh', 'relu', 'relu')	100
12.	Signs	'adam'	'relu'	500	'adam'	('identity', 'relu')	500	'adam'	('identity', 'relu', 'relu')	500
13.	Handwritten Digits	'lbfgs'	'identity'	200	'lbfgs'	('logistic', 'tanh')	200	'lbfgs'	('logistic', 'tanh', 'relu')	200
14.	Cat vs Noncat	'sgd'	'logistic'	300	'sgd'	('logistic', 'relu')	300	'sgd'	('relu', 'logistic', 'relu')	300
15.	Signs	'adam'	'tanh'	100	'adam'	('tanh', 'relu')	100	'adam'	('tanh', 'relu', 'relu')	100
16.	Handwritten Digits	'adam'	'relu'	500	'adam'	('identity', 'relu')	500	'adam'	('identity', 'relu', 'relu')	500
17.	Cat vs Noncat	'lbfgs'	'identity'	200	'lbfgs'	('logistic', 'tanh')	200	'lbfgs'	('logistic', 'tanh', 'relu')	200
18.	Signs	'sgd'	'logistic'	300	'sgd'	('logistic', 'relu')	300	'sgd'	('relu', 'logistic', 'relu')	300
19.	Handwritten Digits	'adam'	'tanh'	100	'adam'	('tanh', 'relu')	100	'adam'	('tanh', 'relu', 'relu')	100
20.	Cat vs Noncat	'sgd'	'relu'	500	'sgd'	('identity', 'relu')	500	'sgd'	('identity', 'relu', 'relu')	500
21.	Signs	'adam'	'tanh'	700	'adam'	('relu', 'relu')	700	'adam'	('relu', 'relu', 'relu')	700