**Лабораторна робота 6**

ДОСЛІДЖЕННЯ РЕКУРЕНТНИХ НЕЙРОННИХ МЕРЕЖ

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

Хід роботи

**Завдання 2.1. Ознайомлення з Рекурентними нейронними мережами**

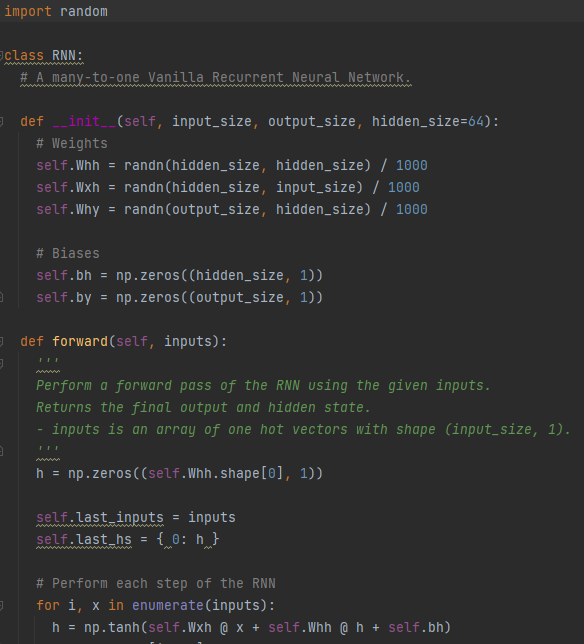


Рис. 1 Код програми

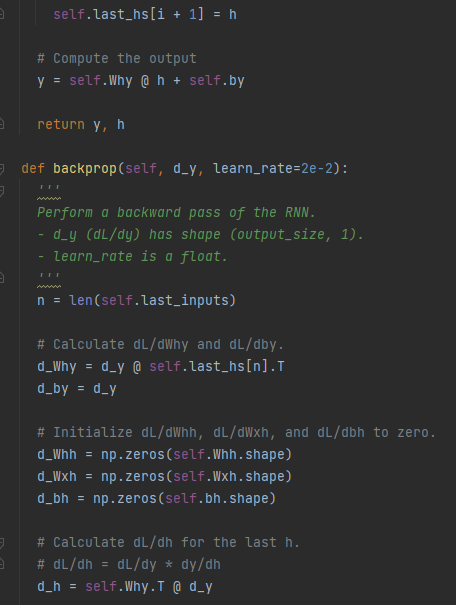


Рис. 2 Код програми

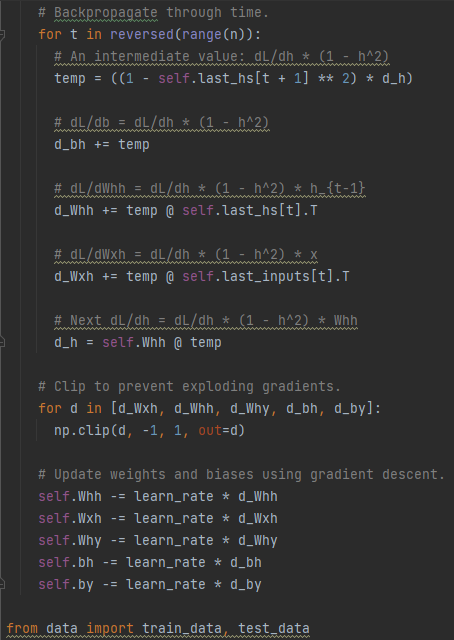


Рис. 3 Код програми

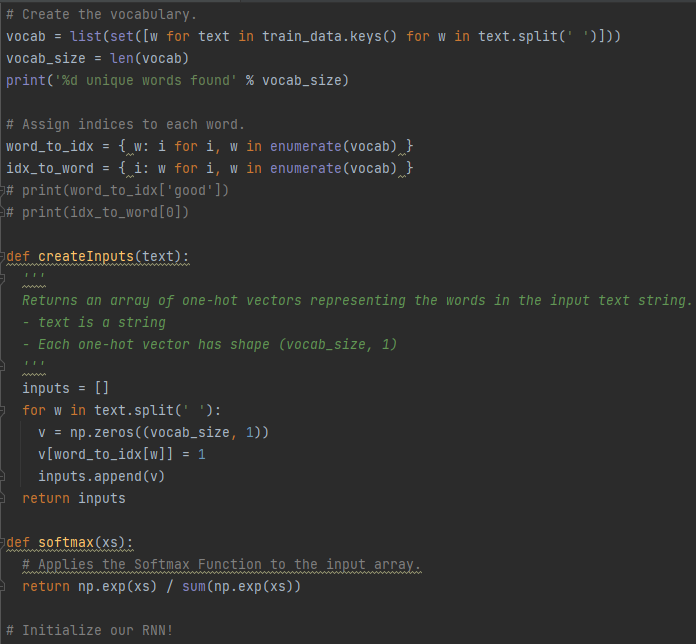


Рис. 4 Код програми

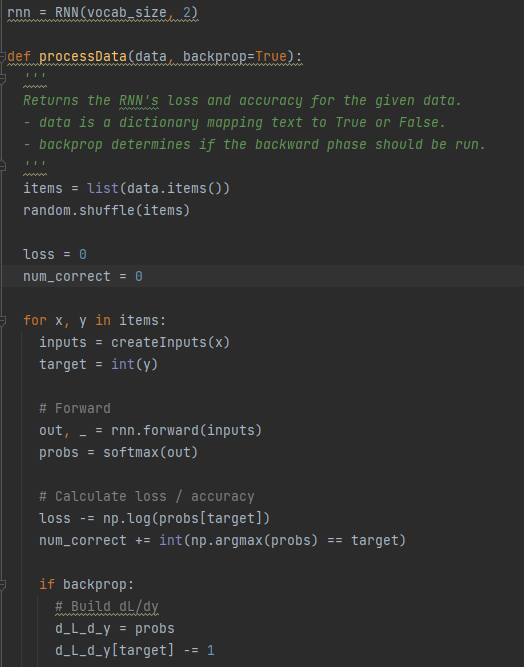


Рис. 5 Код програми

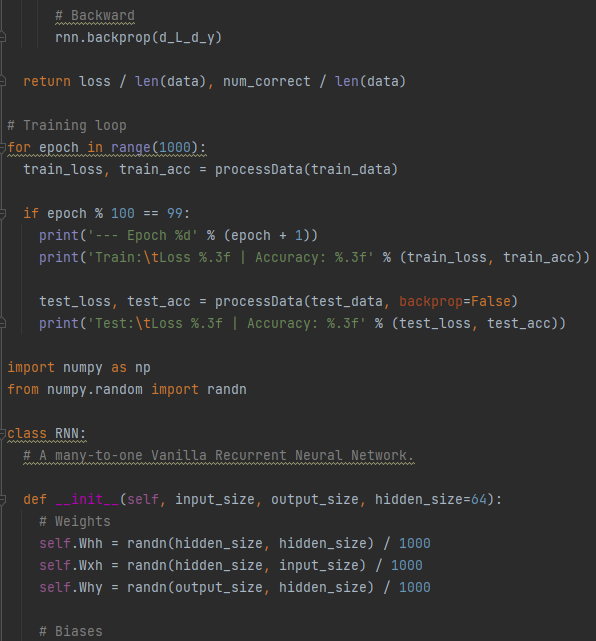


Рис. 6 Код програми

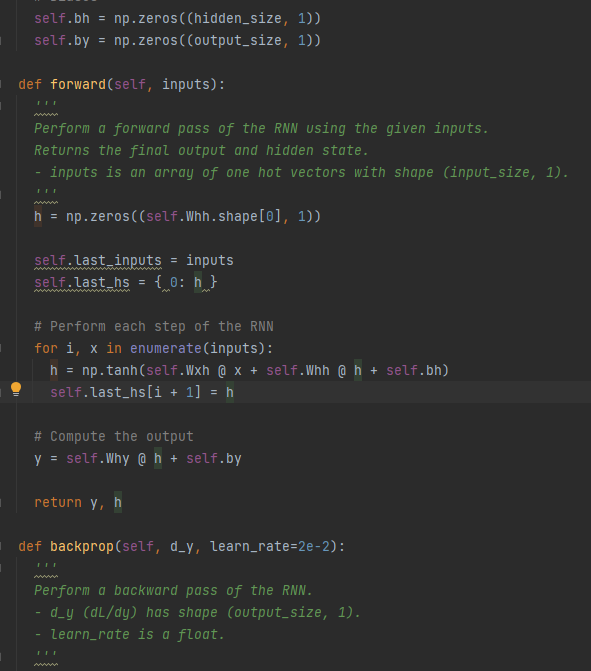


Рис. 7 Код програми

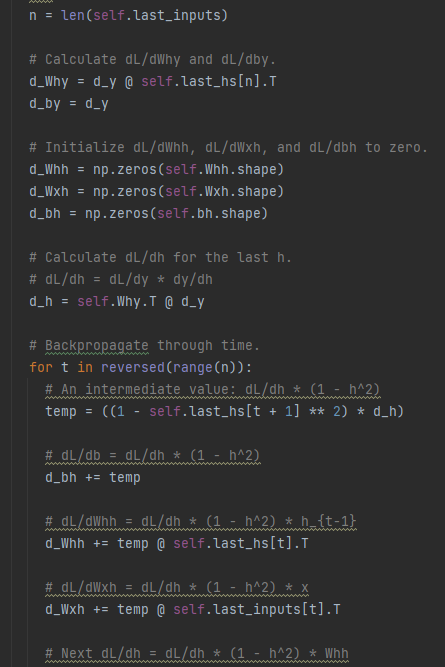


Рис. 8 Код програми

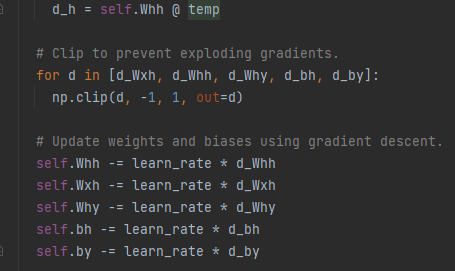


Рис. 9 Код програми

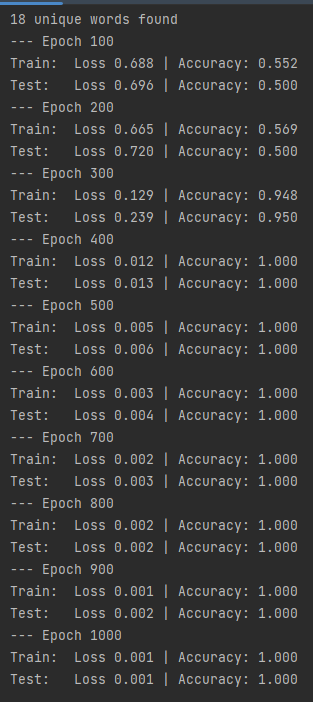


Рис. 10 Виконання файлу main.py

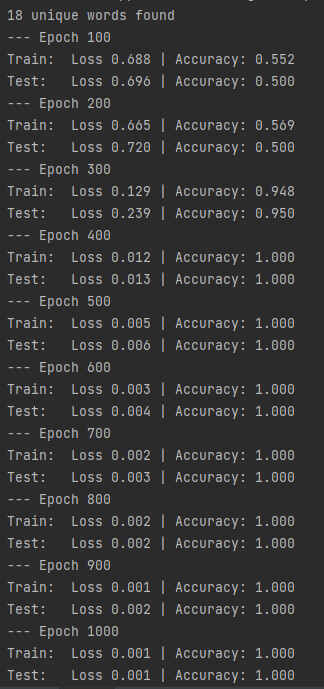


Рис. 11 Виконання файлу LR\_6\_task\_1.py

Висновок: На рисунку 10 -11 ми бачимо виведенне повідомлення “18 unique words found” це означає, що зміна vocab тепер буде мати перелік всіх слів, які вживаються щонайменше в одному навчальному тексті. Рекурентна нейронна мережа не розрізняє слів – лише числа. Тому у словнику 18 унікальних слів, кожне буде 18-мірним унітарним вектором. І далі відбувається тренування мережі. Виведення кожної сотої епохи для відслідковування прогресу.

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

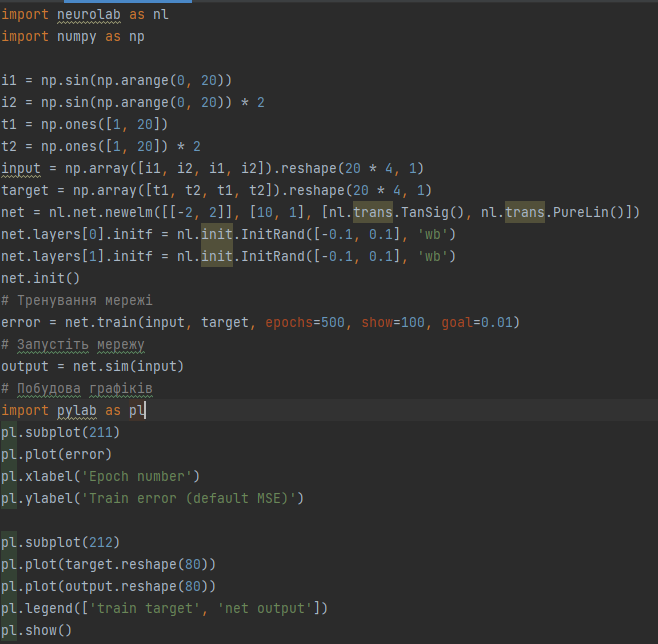


Рис. 12 Код програми

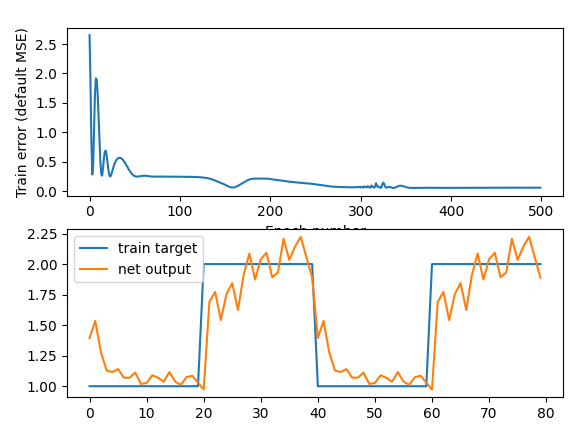


Рис. 13 Виконання програми

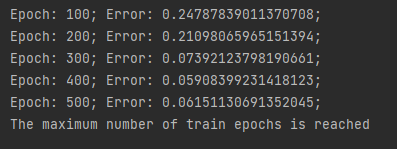


Рис. 14 Виконання програми

Висновок: Під час виконання 2 завдання імпортував neurolab та numpy, створив модель сигналу для навчання мережі, створив мережу з двома прошарками, натренерував мережу та запустив. Результат можна побачити на рис. 13-14

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

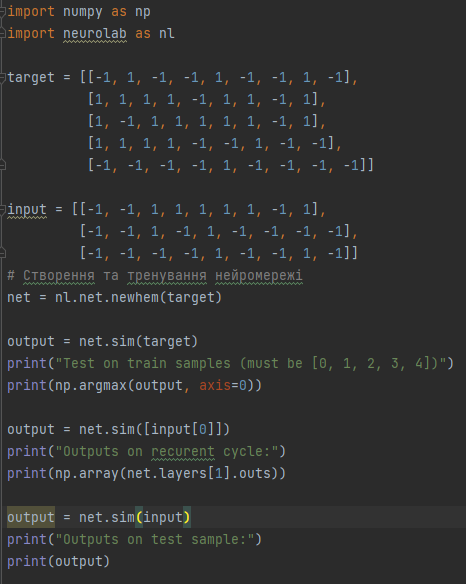


Рис. 15 Код програми

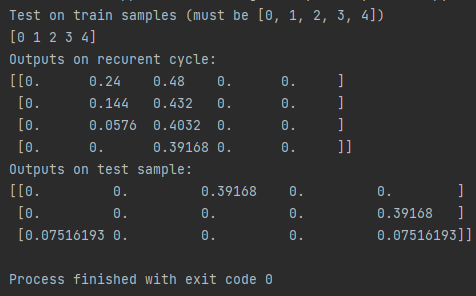


Рис. 16 Виконання програми

**Завдання 2.4. Дослідження рекурентної нейронної мережі Хопфілда Hopfield Recurrent network (newhop)**

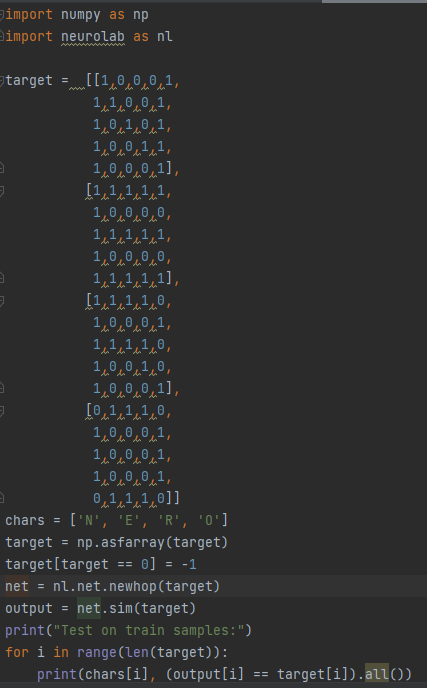


Рис. 17 Код програми

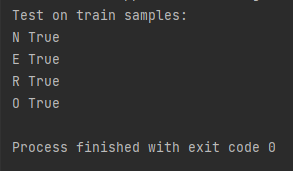


Рис. 18 Виконання програми

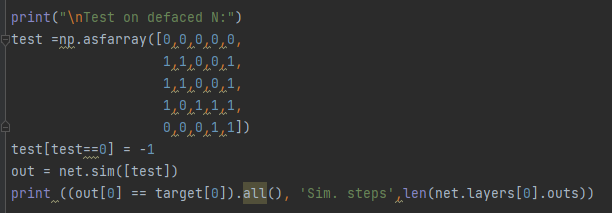


Рис. 19 Код програми

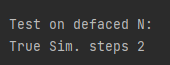


Рис. 20 Виконання програми

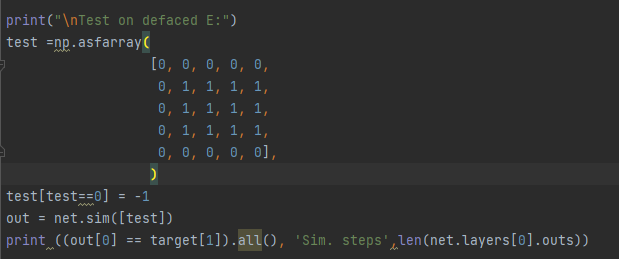


Рис. 21 Код програми



Рис. 22 Виконання програми

Висновок: Під час виконання 4 завдання імпортував neurolab та numpy, заніс вхідні дані у вигляді складного списку та привів до форми, що сприймається функцією з бібліотеки, Створив та навчив нейронну мережу Хопфілда. Протестував навчену нейронну мережу Хопфілда. Для цього замінив деякі білі піксели стали чорними і навпаки. Результат був True(рис. 20). Якщо навчання пройшло правильно то мережа при невеликій кількості помилок буде вгадувати букву правильно. Значить все вірно. Потім вирішив протестувати настпуну букву повністю змінив білі та чорні пікселі. Результат був успішний(рис. 22).

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

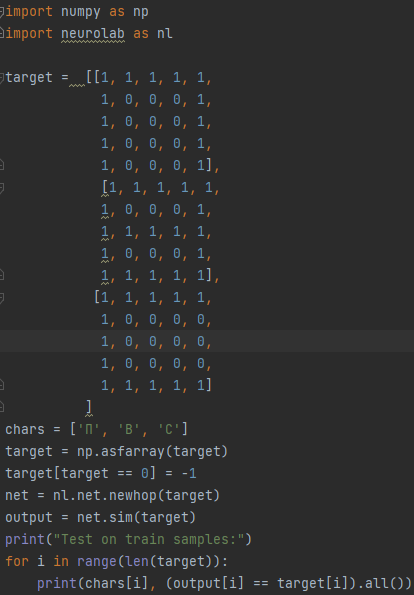


Рис. 23 Код програми

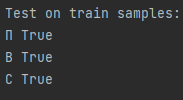


Рис. 24 Виконання програми

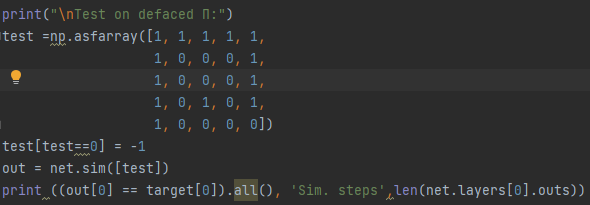


Рис. 25 Код програми



Рис. 26 Виконання програми

Висновок: Під час виконання 5 завдання імпортував neurolab та numpy, заніс вхідні дані у вигляді складного списку та привів до форми, що сприймається функцією з бібліотеки, створив та навчив нейронну мережу Хопфілда. Протестував навчену нейронну мережу Хопфілда. Для цього замінив деякі білі піксели стали чорними і навпаки. Результат був True(рис. 26). Якщо навчання пройшло правильно то мережа при невеликій кількості помилок буде вгадувати букву правильно. Значить все вірно.

**ВИСНОВОК:** під час виконання лабараторної роботи, використовуючи спеціалізовані бібліотеки та мову програмування Python навчився досліджувати деякі типи нейронних мереж***.***

**Репозиторій:**