НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ

«Київський політехнічний інститут імені Ігоря Сікорського»

Факультет інформатики та обчислювальної техніки

Кафедра інформатики та програмної інженерії

**Лабораторна робота № 5**

з дисципліни «Прикладні задачі машинного навчання»

Тема: “Проектування та навчання штучної нейронної мережi для задач

Класифiкацiї”

|  |  |
| --- | --- |
| **Прийняв:** | **Виконав:**  **студент групи ІП-13**  **Недельчев Є.О.** |

Київ – 2023

**Завдання:**

**1. Виконати завдання iз прикладу i отримати файл iз навченою моделлю для розпiзнавання рукописних цифр. В будь-якому графiчному редакторi створити файл iз рукописною цифрою і розпiзнати її. Пояснити результат.**

**2. Спроектувати i розробити нейронну мережу на основi таких**

**наборiв даних iмплементованих в Keras:**

**1) Cifar10**

**2) FMNIST**

**Виконання**

**1. Виконати завдання iз прикладу i отримати файл iз навченою моделлю для розпiзнавання рукописних цифр. В будь-якому графiчному редакторi створити файл iз рукописною цифрою і розпiзнати її. Пояснити результат.**

1. Завантажимо набір даних:

Изображение выглядит как текст, снимок экрана, Шрифт, линия

Автоматически созданное описание

2. Переглянемо перші 25 зображень:

Изображение выглядит как текст, Шрифт, число, снимок экрана

Автоматически созданное описание

3. Перетворимо тривимірний масив в двовимірний масив (60000, 28х28) типу float32, а потім нормалізуємо його так, щоб отримати значення в інтервалі [0, 1]:

Изображение выглядит как текст, Шрифт, снимок экрана, линия

Автоматически созданное описание

4. Підготуємо мітки, тобто завантажимо масиви із відгуками на кожне тестове зображення:

**Изображение выглядит как текст, Шрифт, снимок экрана, линия

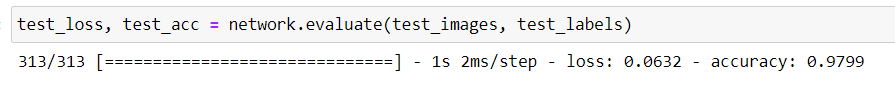
Автоматически созданное описание**

5. Передамо нейронній мережі навчальні дані train\_images і train\_labels, скомпілюємо нейронну мережу, тобто приведемо її до вигляду, сумісного із базовою бібліотекою TensorFlow. Налаштуємо при цьому три додаткові параметри (optimizer, loss, metrics):

Изображение выглядит как текст, снимок экрана, Шрифт, число

Автоматически созданное описание

6. Після навчання моделі перевіримо її на контрольному наборі даних, які ще не пред’являлися моделі:



7. Збережемо модель Keras в одному файлі HDF5:

****

8. Завантажимо попередньо збережену модель:

Изображение выглядит как текст, Шрифт, снимок экрана, линия

Автоматически созданное описание

9. Для тестування моделі було створено картинку з рукописною цифрою 2 наступного вигляду:

Изображение выглядит как зарисовка, рисунок, штриховой рисунок

Автоматически созданное описание

10. Завантажуємо її в OpenCV як сіре зображення і отримуємо данні у форматі ndarray. Потім цей масив переформатовується до того розміру на якому відбувалося тренування нашої моделі:

Изображение выглядит как текст, Шрифт, снимок экрана, линия

Автоматически созданное описание

11. Підготовлені дані передаємо функції model.predict(tst). На виході отримаємо список списків із одним елементом – списком довжини 10, у якому на і-ій позиції знаходяться ймовірність того, що на вхідному зображенні є число і. Нам потрібна позиція з максимальною ймовірністю.

**Изображение выглядит как текст, снимок экрана, Шрифт, линия

Автоматически созданное описание**

**Бачимо, що модель коректно визначила зображену цифру.**

**2. Спроектувати i розробити нейронну мережу на основi таких**

**наборiв даних iмплементованих в Keras:**

1. **Cifar10**
2. Завантажимо дані:



1. Після завантаження дані перетворюються на числа типу float32 та нормалізуються шляхом поділу на 255. Це допомагає в масштабуванні значень пікселів зображень між 0 та 1, сприяючи збільшенню швидкості навчання моделі:

Изображение выглядит как текст, Шрифт, снимок экрана, линия

Автоматически созданное описание

1. Перетворимо вектори міток y\_train та y\_test на бінарні матриці за допомогую функції ‘to\_categorical’.

Изображение выглядит как текст, Шрифт, снимок экрана, линия

Автоматически созданное описание

1. Побудуємо власне модель. Додамо до моделі шари ‘Conv2D’ з функцією активації 'relu' для здійснення згорткового перетворення зображень. Після кожного шару Conv2D додамо шар MaxPooling2D для зменшення розмірності зображення. Після останнього шару Conv2D додамо шар Flatten, який перетворює вихідні дані в одновимірний вектор. Останній повнозв'язний шар має 10 виходів та використовує функцію активації 'softmax' для визначення ймовірностей належності вхідного зображення до кожного з 10 класів.

Изображение выглядит как текст, снимок экрана, Шрифт

Автоматически созданное описание

1. Скомпілюємо та навчимо модель:

Изображение выглядит как текст, снимок экрана, число, Шрифт

Автоматически созданное описание

1. Перевіримо її точність на контрольному наборі даних:

Изображение выглядит как текст, снимок экрана, Шрифт, дисплей

Автоматически созданное описание

1. Візуалізуємо перші 10 прогнозів, зіставивши їх з відповідними зображеннями:

Изображение выглядит как текст, снимок экрана, Веб-сайт, веб-страница

Автоматически созданное описание

Точність моделі на тестових даних склала 70.7%. На представлених вище картинках можна наочно побачити результат роботи нейромережі, рівно як і її результативність.

**2) FMNIST**

1. Завантажимо дані:

Изображение выглядит как текст, Шрифт, линия, снимок экрана

Автоматически созданное описание

2. Змінимо розмірність зображень, щоб врахувати канали кольору. Для цього розмірність тренувальних і тестових даних (x\_train, x\_test) змінимо на (кількість зображень, 28, 28, 1). Далі перетворимо дані на числа типу float32 та нормалізуємо шляхом ділення на 255.

3. Перетворимо вектори міток y\_train та y\_test на бінарні матриці за допомогою функції ‘to\_categorical’.

Изображение выглядит как текст, Шрифт, снимок экрана, линия

Автоматически созданное описание

4. Побудуємо саму модель:

Изображение выглядит как текст, Шрифт, снимок экрана

Автоматически созданное описание

5. Скомпілюємо її та навчимо на тренувальному наборі даних:

Изображение выглядит как текст, снимок экрана, число, Шрифт

Автоматически созданное описание

6. Візуалізуємо результати роботи мережі на декількох випадкових прикладах:

Изображение выглядит как текст, снимок экрана, одежда, дизайн

Автоматически созданное описание

Як можна побачити, нейромережа майже безпомилково визначила усі обрані зображення.

**Висновок**

Виконуючи цю лабораторну роботу я навчився реалізовувати прості нейронні мережі на прикладі датасетів, вбудованих в бібліотеку Keras. В результаті було розроблено дві моделі, які здатні вирішувати задачі класифікації об’єктів на зображеннях з непоганим результатом. Також під час роботи я використав на практиці згорткові нейронні мережі та добився від них солідної результативності.