

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
“КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ ІМЕНІ ІГОРЯ СІКОРСЬКОГО”



Системи штучного інтелекту

Логістична регресія

Практична робота #3

Виконав(ла):
Ім'я Прізвище
Група:
ІО-xxx
Курс:
X

27 жовтня 2023 р.

1 Логістична регресія

1.1 Реалізація логістичної регресії

Подайте у звіті імplementовані Вами функції. Це можна зробити через зовнішній файл, наприклад так:

Лістинг 1: Перевірка на парність введеного числа.

```
1 # Python program to check if the input number is odd or even.
2 # A number is even if division by 2 gives a remainder of 0.
3 # If the remainder is 1, it is an odd number.
4
5 num = int(input("Enter a number: "))
6 if (num % 2) == 0:
7     print("{0} is Even".format(num))
8 else:
9     print("{0} is Odd".format(num))
```

Або вставивши безпосередньо потрібні рядки коду в оточення `lstlisting`:

Лістинг 2: Перетин двох масивів.

```
1 def intersection(array_1, array_2):
2     element_1 = set()
3     intersected = []
4     for i in range(0, len(array_1)):
5         element_1.add(array_1[i])
6     already_added = set()
7     for j in range(0, len(array_2)):
8         if array_2[j] in element_1 and array_2[j] not in already_added:
9             intersected.append(array_2[j])
10            # MISSING LINE HERE.
11            already_added.add(array_2[j])
12    return intersected
```

Без рамки та без підпису:

```
1 def intersection(array_1, array_2):
2     element_1 = set()
3     intersected = []
4     for i in range(0, len(array_1)):
5         element_1.add(array_1[i])
6     already_added = set()
7     for j in range(0, len(array_2)):
8         if array_2[j] in element_1 and array_2[j] not in already_added:
9             intersected.append(array_2[j])
10            # MISSING LINE HERE.
11            already_added.add(array_2[j])
12    return intersected
```

1.2 Результати експериментів

Тут Ви повинні подати детальну інформацію про результати свого дослідження щодо впливу швидкості навчання та ітерацій навчання на значення цільової функції та точність моделі логістичної регресії на тестовій вибірці.

Для представлення результатів використовуйте таблиці, графіки, рисунки. Приклад таблиці 1. Приклад рисунку 1.

Табл. 1: Приклад таблиці.

| Ім'я оператора | | Синтаксис |
|---------------------|---------|-----------|
| Присвоєння | | $a = b$ |
| Додавання | | $a + b$ |
| Віднімання | | $a - b$ |
| Унарний плюс | | $+a$ |
| Унарний мінус | | $-a$ |
| Множення | | $a * b$ |
| Ділення | | a / b |
| Залишок від ділення | | $a \% b$ |
| Інкремент | префікс | $++a$ |
| | суфікс | $a++$ |
| Декремент | префікс | $--a$ |
| | суфікс | $a--$ |

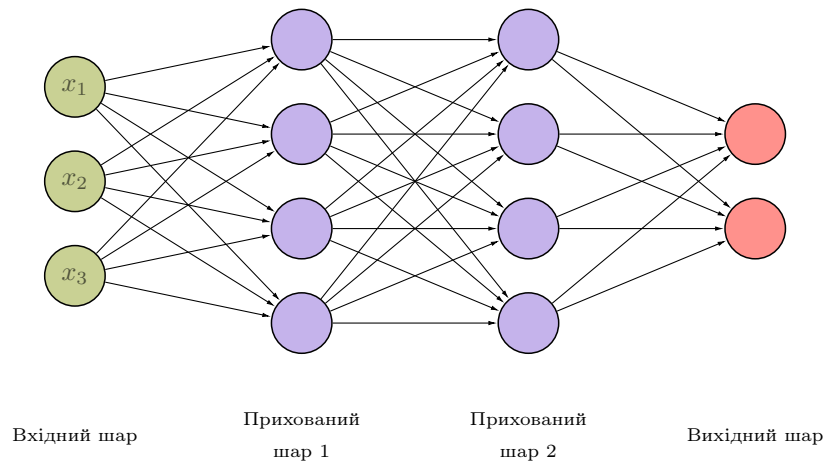


Рис. 1: Приклад представлення нейронної мережі.

1.3 Допомога

У цьому розділі потрібно залишити додаткові деталі про виконання цієї роботи. Тобто, якщо Ви обговорювали завдання або працювали над ним у групі – потрібно це зазначати: з ким працювали та що кожен учасник виконував. Якщо отримували допомогу – вказати від кого та яку допомогу було отримано. Якщо використовували додаткові матеріали (блокноти Kaggle, github тощо) – залиште посилання на джерела. Якщо завдання було виконано особисто Вами і Ви не отримували ніякої допомоги – так і напишіть. Використання будь-яких чужих матеріалів та представлення їх за свої напрацювання є плагіатом та серйозним порушенням основних академічних стандартів.

Приклад для використання цитування [1, 2].

1.4 Висновки

Резюмуйте пророблену Вами роботу, відобразіть свої спостереження щодо змін цільової функції та точності моделі залежно від швидкості навчання та кількості ітерацій навчання.

Література

- [1] M. Mathur. Cnn using keras(100% accuracy). Kaggle. [Online]. Available: <https://www.kaggle.com/madz2000/cnn-using-keras-100-accuracy>
- [2] R. Jain. Deep learning using sign language. Kaggle. [Online]. Available: <https://www.kaggle.com/ranjeetjain3/deep-learning-using-sign-language>