

**НАЦИОНАЛЕН ВОЕНЕН УНИВЕРСИТЕТ
„ВАСИЛ ЛЕВСКИ“**

ФАКУЛТЕТ „АРТИЛЕРИЯ, ПВО и КИС“

КУРСОВА РАБОТА



На тема:

„Размито съпоставяне на низове в Python“

Разработил:

Али Мурадов

Ф. ном. : 74322401

Проверил:

проф. д.н. инж. Жанета Савова

Описание на проекта:

„Изграждане на невронна мрежа за класификация“

1. **Цел на задачата:** Целта на този проект е да се построи и обучи невронна мрежа, която използва даден набор от входни данни (матрица с характеристики) и се научава да класифицира тези данни към един от няколко класа. Задачата има за цел да демонстрира ключовите принципи на работата на невронните мрежи, включително forward propagation, backpropagation и процеса на оптимизация.

2. Описание на набора от данни

- **Входни данни (матрица x):** Всеки ред представлява обект с три характеристики (RGB цветове). RGB стойностите могат да се разглеждат като описания на продукти или изображения.

```
X = np.array([
    [0, 0, 1],
    [1, 1, 1],
    [1, 0, 1],
    [0, 1, 1],
    [1, 0, 0],
    [0, 1, 0]
])
```

- **Етикети (матрица y):** Категории на обектите. Етикетите могат да представляват различни типове продукти като дрехи, обувки или аксесоари.

```
y = np.array([
    [1, 0, 0],
    [0, 1, 0],
    [0, 0, 1],
    [1, 0, 0],
    [0, 1, 0],
    [0, 0, 1]
])
```

3. Архитектура на невронната мрежа:

- **Входен слой:** 3 входни неврона, съответстващи на трите RGB характеристики.
- **Скрит слой:** 4 неврона, които служат за разкриване на нелинейни зависимости между входните характеристики и класовете.
- **Изходен слой:** 3 неврона, които представляват вероятностите за принадлежност към всеки клас (дреха, обувка или аксесоар).

4. Процесът на обучение:

- **Forward propagation:** Предаване на входните данни през слоевете на мрежата, за да се получат предсказания.

```
hidden_layer = sigmoid(np.dot(input_layer, weights_input_hidden))
output_layer = sigmoid(np.dot(hidden_layer, weights_hidden_output))
```

- **Backpropagation:** Изчисляване на грешката между предсказанията и реалните етикети, след което се изчисляват корекции на теглата.

```
output_error = y - output_layer
output_delta = output_error * sigmoid_derivative(output_layer)
hidden_error = np.dot(output_delta, weights_hidden_output.T)
hidden_delta = hidden_error * sigmoid_derivative(hidden_layer)
```

- **Актуализация на теглата:** Теглата се актуализират, за да минимизират грешката при следващите итерации.

```
weights_hidden_output += np.dot(hidden_layer.T, output_delta)
weights_input_hidden += np.dot(input_layer.T, hidden_delta)
```

5. Резултати:

- **Тестване:** При вход [1, 0, 0] (червен цвят), предсказанието е:

```
Предсказание за новите данни: [0.00143196 0.99053543 0.00958792]
```

Мрежата успешно определя, че обектът принадлежи към втория клас (обувка).

- **Обяснение на резултатите:**

- Стойностите в изходния вектор представляват вероятности за принадлежност към всеки клас.
- Най-високата стойност (0.9905) указва, че мрежата е с висока увереност, че обектът е обувка.

6. Заключение: Този проект представя основите на невронните мрежи и ясно демонстрира как работят forward и backpropagation. Програмата показва ефективността на мрежата за класификация на данни, базирани на прости характеристики (RGB цветове). Тя може да бъде разширена за по-сложни задачи, като например разпознаване на изображения или текст, чрез добавяне на повече слоеве и използване на по-големи набори от данни.