

Міністерство освіти та науки України
Львівський національний університет імені Івана Франка

Лабораторна робота №8

**на тему: «Реалізація класифікатора даних на основі нейронної
мережі з одним прихованим шаром»**

Виконав:
студент групи ФеС-31
Козак Дмитро
Перевірів:
Рибак А. В.

Львів – 2021

Мета: Нехай ми маємо деякий набір даних, який надано у додатку.

Реалізувати модель нейронної мережі з одним прихованим шаром для класифікації даних засобами Python + NumPy. Візуалізувати функцію витрат і результат нейронної мережі.

Хід роботи:

1. Реалізувати модель нейронної мережі з одним прихованим шаром для класифікації даних.

```
class NeuralNetwork:
    def __init__(self, neuron_nums=(2, 4, 1)):
        self.input_neurons = neuron_nums[0]
        self.hidden_neurons = neuron_nums[1]
        self.output_neurons = neuron_nums[2]
        self.W1 = np.random.randn(self.hidden_neurons, self.input_neurons) * 0.01
        self.b1 = np.zeros((self.hidden_neurons, 1))
        self.W2 = np.random.randn(self.output_neurons, self.hidden_neurons) * 0.01
        self.b2 = np.zeros((self.output_neurons, 1))

    def forward_propagation(self, X):
        self.Z1 = np.dot(self.W1, X) + self.b1
        self.A1 = np.tanh(self.Z1)
        self.Z2 = np.dot(self.W2, self.A1) + self.b2
        self.A2 = sigmoid(self.Z2)
        return self.A2

    def backward_propagation(self, X, Y):
        m = Y.shape[1]
        dZ2 = self.A2 - Y
        self.dW2 = (1 / m) * np.dot(dZ2, self.A1.T)
        self.db2 = (1 / m) * np.sum(dZ2, axis=1, keepdims=True)

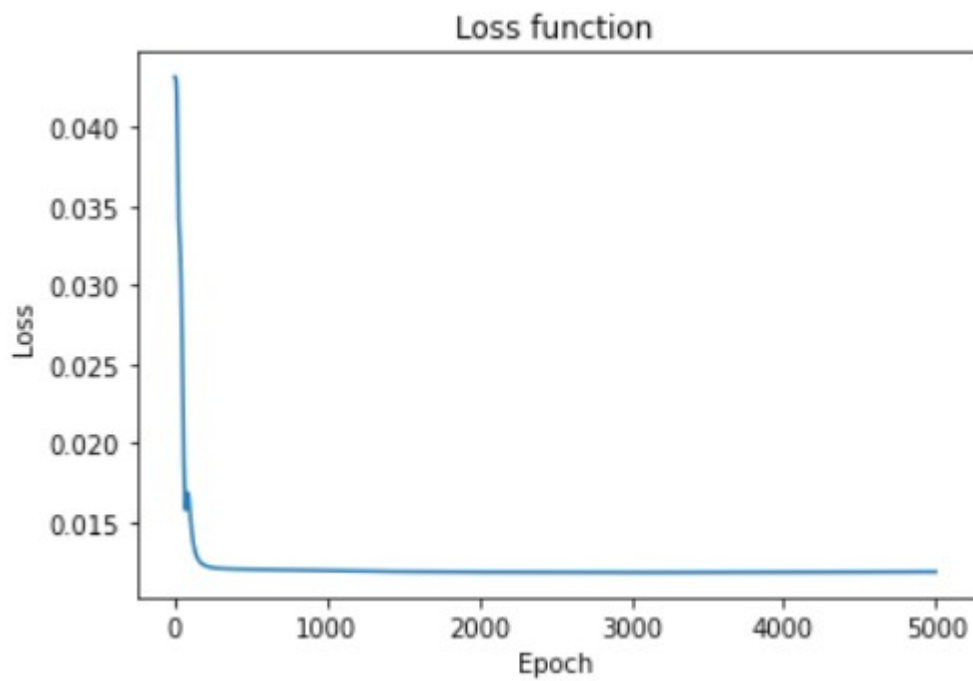
        dZ1 = np.dot(self.W2.T, dZ2) * tanh_derivative(self.Z1)
        self.dW1 = 1/m * np.dot(dZ1, X.T)
        self.db1 = 1/m * np.sum(dZ1, axis = 1, keepdims = True)

    def update_weights(self):
        self.W1 -= self.dW1
        self.W2 -= self.dW2

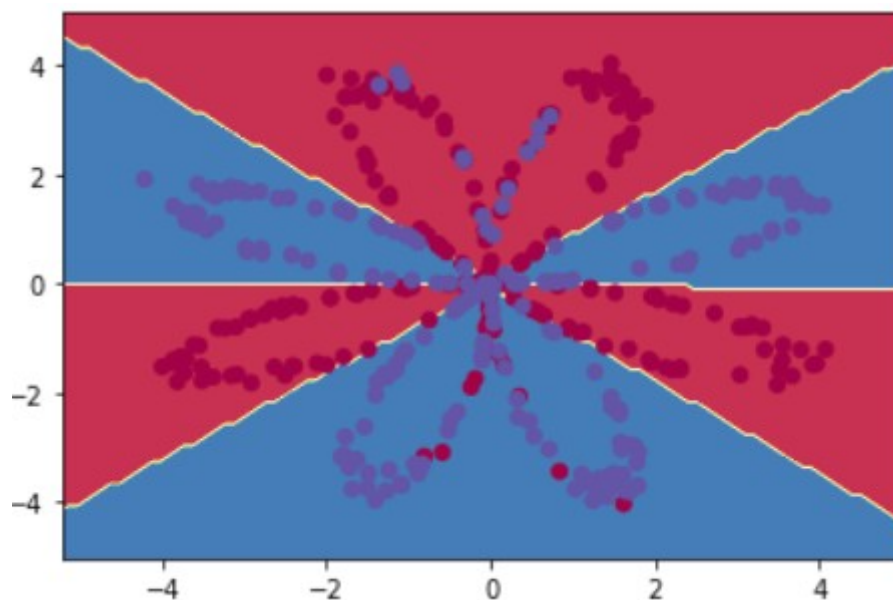
        self.b1 -= self.db1
        self.b2 -= self.db2
```

Модель нейронної мережі з одним прихованим шаром

2. Візуалізувати функцію витрат і результат нейронної мережі.



Функція витрат



Результат нейронної мережі

3. Проаналізувати нейронну мережу

Аналіз було проведено на 9 нейронних мережах з 3 шарами, з різною кількістю нейронів у прихованому шарі та з різною кількістю епох (1000, 5000, 10000). Характеристикою аналізу я вибрав ROC_AUC – площа під кривою ROC яка показує міру якості бінарної класифікації.

```
hidden neurons num: 1 epochs num 1000 roc_auc score: 0.68
hidden neurons num: 1 epochs num 5000 roc_auc score: 0.67
hidden neurons num: 1 epochs num 10000 roc_auc score: 0.68
hidden neurons num: 2 epochs num 1000 roc_auc score: 0.65
hidden neurons num: 2 epochs num 5000 roc_auc score: 0.67
hidden neurons num: 2 epochs num 10000 roc_auc score: 0.67
hidden neurons num: 3 epochs num 1000 roc_auc score: 0.90
hidden neurons num: 3 epochs num 5000 roc_auc score: 0.91
hidden neurons num: 3 epochs num 10000 roc_auc score: 0.91
hidden neurons num: 4 epochs num 1000 roc_auc score: 0.88
hidden neurons num: 4 epochs num 5000 roc_auc score: 0.91
hidden neurons num: 4 epochs num 10000 roc_auc score: 0.92
hidden neurons num: 5 epochs num 1000 roc_auc score: 0.88
hidden neurons num: 5 epochs num 5000 roc_auc score: 0.90
hidden neurons num: 5 epochs num 10000 roc_auc score: 0.91
hidden neurons num: 6 epochs num 1000 roc_auc score: 0.89
hidden neurons num: 6 epochs num 5000 roc_auc score: 0.90
hidden neurons num: 6 epochs num 10000 roc_auc score: 0.91
hidden neurons num: 7 epochs num 1000 roc_auc score: 0.88
hidden neurons num: 7 epochs num 5000 roc_auc score: 0.92
hidden neurons num: 7 epochs num 10000 roc_auc score: 0.91
hidden neurons num: 8 epochs num 1000 roc_auc score: 0.88
hidden neurons num: 8 epochs num 5000 roc_auc score: 0.91
hidden neurons num: 8 epochs num 10000 roc_auc score: 0.91
hidden neurons num: 9 epochs num 1000 roc_auc score: 0.88
hidden neurons num: 9 epochs num 5000 roc_auc score: 0.91
hidden neurons num: 9 epochs num 10000 roc_auc score: 0.92
```

Результати аналізу нейронної мережі

Як видно з рисунку вище — зі збільшенням кількості епох та кількості нейронів у прихованому шарі зростає також значення ROC_AUC, проте після 3 нейронів зріст не вагомий

ВИСНОВОК

Під час виконання лабораторної роботи я реалізував класифікатор даних на основі нейронної мережі з одним прихованим шаром. Код та звіт завантажив на свій github: https://github.com/newbeepi/LNU_ML.