

Лабораторна робота №1
“Навчання персептрона”

Група ТК-41
Соловей Євгеній

Генерація точок

Генеруємо точки в обмежених координатах. Першу групу точок обмежимо $[0, 0]$ та $[0.5, 0.5]$ другу групу обмежимо $[0.5, 0.5]$ та $[1, 1]$.

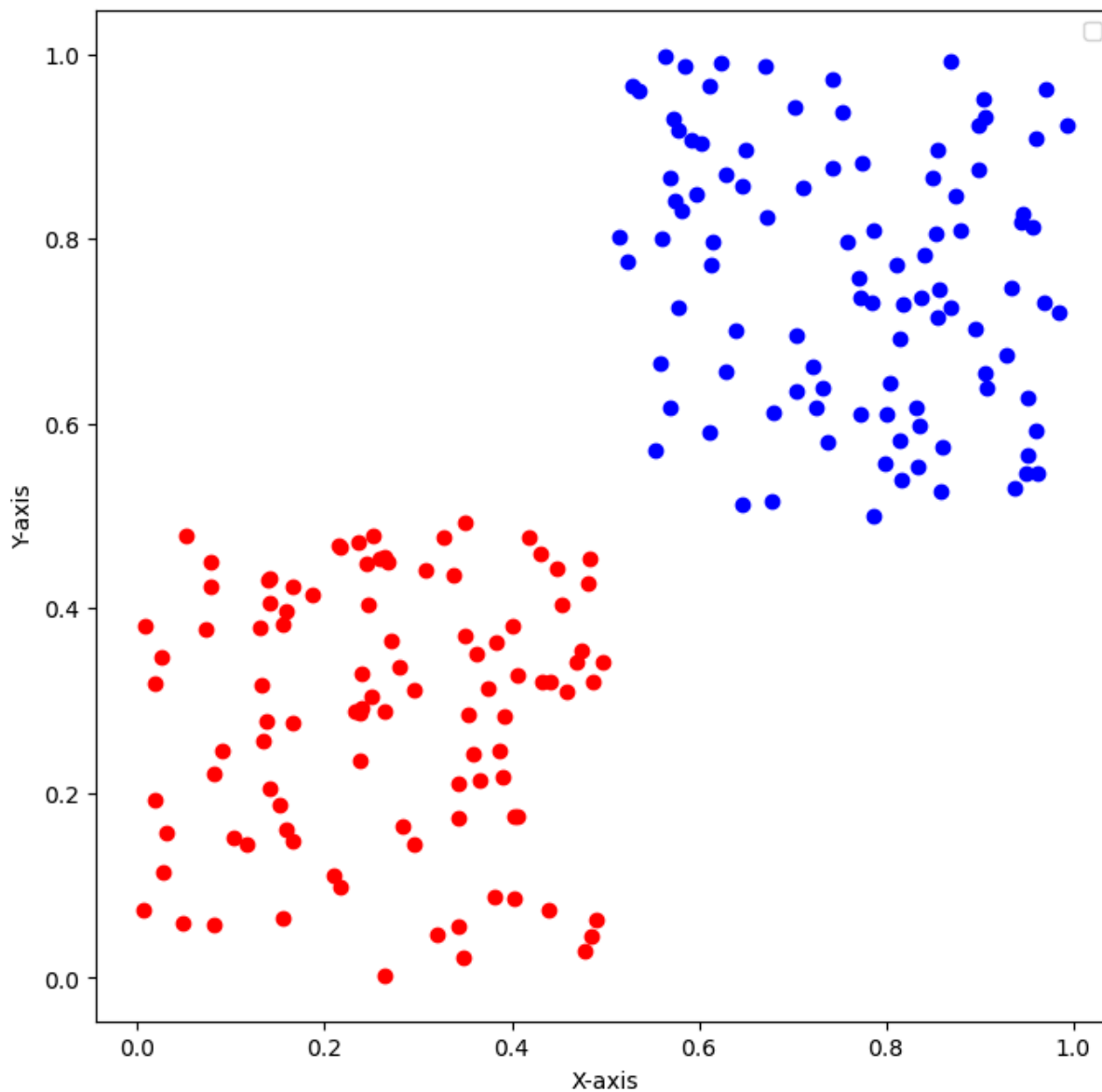


Рис. 1. Візуалізація 100 згенерованих точок для кожної з груп

Навчання перцептрона

Функція генерації 150000 точок написана за допомогою random:

```
def generate_points():  
    X = []  
    target = []  
  
    for i in range(150000):  
        x = np.random.uniform(0, 0.5)  
        y = np.random.uniform(0, 0.5)  
        z = np.random.uniform(0, 0.5)  
        X.append(np.array((x, y, z)))  
        target.append(-1)  
  
    for i in range(150000):  
        x = np.random.uniform(0.5, 1)  
        y = np.random.uniform(0.5, 1)  
        z = np.random.uniform(0.5, 1)  
        X.append(np.array((x, y, z)))  
        target.append(1)  
  
    X = np.array(X)  
    target = np.array(target)  
    return X, target
```

Рис. 2. Функція для генерації точок

Алгоритм навчання перцептрона:

1. Покладемо вектор ваг w
2. Будемо повторювати N разів наступні кроки
3. Для кожного тестового набору
4. Обчислити $y = \text{sing}[(x, w)]$
5. Якщо $y \cdot d < 0$, то коригуємо ваги: $w = w + a \cdot d \cdot x$, де a – деяке число (швидкість навчання) з інтервалу $(0; 1)$

```
X, y = generate_points()

w1 = -0.01
w2 = 0.02
w3 = 0.03

a = 0.15
errors = []

for i in range(10):
    correct = 0
    train = list(zip(X, y))
    random.shuffle(train)
    for x, target in train:
        pred = (w1 * x[0] + w2 * x[1])
        # pred = (w1 * x[0] + w2 * x[1] + w3 * x[2])
        activation = pred if abs(pred) > 0.01 else -1 * pred
        if activation > 0:
            pred = 1
        else:
            pred = -1
        if target * pred <= 0:
            w1 += a * target * x[0]
            w2 += a * target * x[1]
            # w3 += a * target * x[2]
        else:
            correct += 1
    errors.append(correct)

print(errors)
print(w1, w2)
```

Рис. 3. Двовимірний випадок

Загальна кількість згенерованих даних була 300000, та було 10 епох. На наступному зображенні можемо побачити як змінюється передбачання значенб протягом навчання:

```
[278198, 288691, 283805, 279689, 300000, 300000, 300000, 300000, 300000, 300000]  
0.01695074557746047 0.003049044520187602
```

Бачимо що отримані ваги - це $w1 = 0.016$ та $w2 = 0.003$.

Для 3-х вимірнього випадку трохи змінимо код на наступний:

```
pred = (w1 * x[0] + w2 * x[1] + w3 * x[2])  
  
w3 += a * target * x[2]  
  
print(w1, w2, w3),
```

та отримаємо результат:

```
[162199, 270721, 300000, 300000, 300000, 300000, 300000, 300000, 300000, 300000]  
0.006264519271481014 0.0019920614606057964 0.011814891923571255
```

Бачимо що отримані ваги - це $w1 = 0.0062$, $w2 = 0.0019$ та $w3 = 0.0118$.