

# Лабораторная работа №0

## Задание

Применить нейронную сеть Кохонена с самообучение для задачи кластеризации.

На первом этапе сгенерировать случайные точки на плоскости вокруг 2 центров кластеризации (примерно по 20-30 точек).

Далее считать, что сеть имеет два входа (координаты точек) и два выхода – один из них равен 1, другой 0 (по тому, к какому кластеру принадлежит точка).

Подавая последовательно на вход (вразнобой) точки, настроить сеть путем применения описанной процедуры обучения так, чтобы она приобрела способность определять, к какому кластеру принадлежит точка.

Коэффициент  $\alpha$  выбрать, уменьшая его от шага к шагу по правилу  $\alpha = (50-i)/100$ , причем для каждого нейрона это будет свое значение  $\alpha$ , а подстраиваться на каждом шаге будут веса только одного (выигравшего) нейрона.

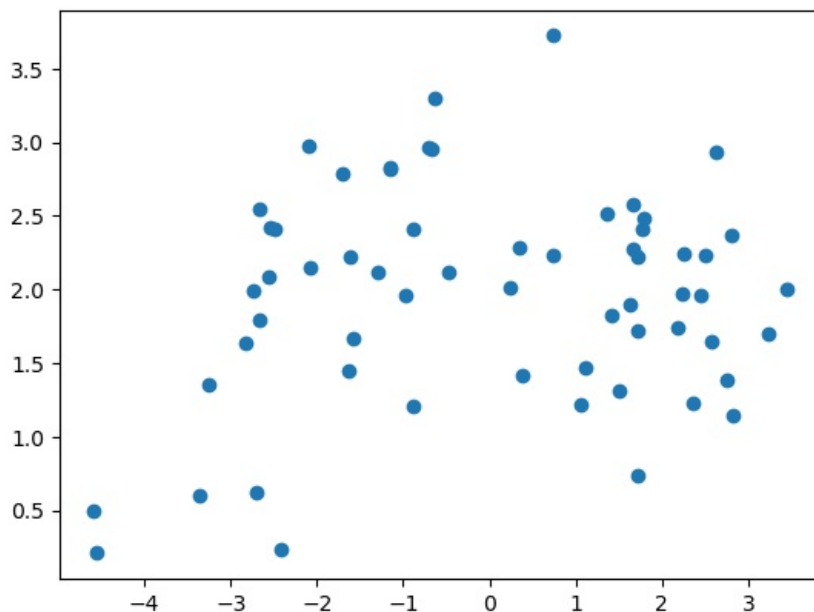
```
In [703] from sklearn.datasets import make_classification
```

```
X, y = make_classification(n_samples=60,  
                           n_features=2,  
                           n_redundant=0,  
                           n_informative=2,  
                           n_clusters_per_class=1,  
                           n_classes=2,  
                           random_state=56,  
                           class_sep=2)
```

```
In [704] import matplotlib.pyplot as plt
```

```
plt.scatter(X[:, 0], X[:, 1])
```

```
Out[704]: <matplotlib.collections.PathCollection at 0x147814150>
```



```
In [705] import numpy as np
```

```
def update_cluster_centers(X, c):  
    centers = np.zeros((2, 2))  
    for i in range(1, 3):  
        ix = np.where(c == i)  
        centers[i - 1, :] = np.mean(X[ix, :], axis=1)  
    return centers
```

```
In [706] from scipy.cluster.hierarchy import fcluster, linkage
```

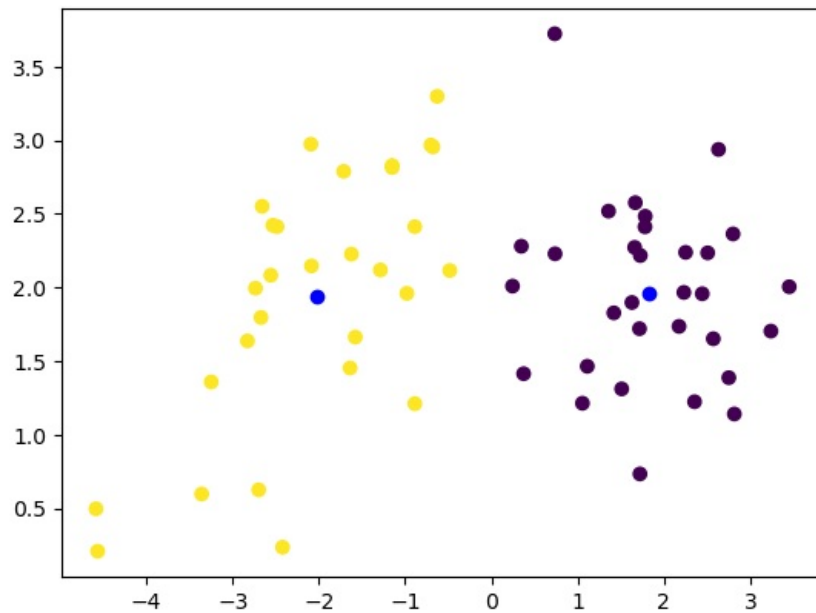
```
mergings = linkage(X, method='ward')  
T = fcluster(mergings, 2, criterion='maxclust')
```

```
clusters = update_cluster_centers(X, T)
clusters
```

```
Out[706]: array([[ 1.83254732,  1.96280369],
        [-2.03146912,  1.94312656]])
```

```
In [707]: plt.scatter(X[:, 0], X[:, 1], c=T)
plt.scatter(clusters[:, 0], clusters[:, 1], c='blue')
```

```
Out[707]: <matplotlib.collections.PathCollection at 0x147843b50>
```



```
In [708]: import math
```

```
class SOM:
    def __init__(self, n, c):
        """
        n - количество атрибутов
        C - количество кластеров
        """
        self.n = n
        self.c = c
        self.a = [0 for _ in range(n)]

    def calculate_a(self, i):
        """
        Вычисление значение шага относительного текущего выбора
        """
        return (50 - i) / 100

    def winner(self, weights, sample):
        """
        Вычисляем выигравший нейрон (вектор) по Евклидову расстоянию
        """
        d0 = 0
        d1 = 0
        for i in range(len(sample)):
            d0 += math.pow((sample[i] - weights[0][i]), 2)
            d1 += math.pow((sample[i] - weights[1][i]), 2)

        if d0 > d1:
            return 0
        else:
            return 1

    def update(self, weights, sample, j):
        """
        Обновляем значение для выигравшего нейрона
        """
        for i in range(len(weights)):
            weights[j][i] = weights[j][i] + self.calculate_a(self.a[j]) * (sample[i] - weights[j][i])

        print(f'\nШаг для {j} кластера = {self.calculate_a(self.a[j])}')
        self.a[j] += 1
        print(f'Веса после обновления:')
        print(weights)

        return weights
```

```
In [709]: # Обучающая выборка (m, n)
# m - объем выборки
# n - количество атрибутов в записи
```

```

np.random.shuffle(X)
T = X
m, n = len(T), len(T[0])

# Обучающие веса (n, C)
# n - количество атрибутов в записи
# C - количество кластеров
C = 2

weights = np.random.normal(100, 10, size=(n, C)) / 100
weights

```

```

Out[709]: array([[1.09909639, 0.86606284],
                [0.92866499, 0.98073679]])

```

```

In [710]: som = SOM(n, C)
          som

```

```

Out[710]: <__main__.SOM at 0x147446d90>

```

```

In [711]: for i in range(m):
          sample = T[i]
          J = som.winner(weights, sample)
          weights = som.update(weights, sample, J)

```

Шаг для 0 кластера = 0.5

Веса после обновления:

```

[[0.10539675  1.63957672]
 [0.92866499  0.98073679]]

```

Шаг для 1 кластера = 0.5

Веса после обновления:

```

[[ 0.10539675  1.63957672]
 [-0.74608651  0.60786591]]

```

Шаг для 1 кластера = 0.49

Веса после обновления:

```

[[0.10539675  1.63957672]
 [0.46070016  1.1527871 ]]

```

Шаг для 1 кластера = 0.48

Веса после обновления:

```

[[ 0.10539675  1.63957672]
 [-0.18657318  1.1805079 ]]

```

Шаг для 1 кластера = 0.47

Веса после обновления:

```

[[0.10539675  1.63957672]
 [0.95939365  1.67774359]]

```

Шаг для 0 кластера = 0.49

Веса после обновления:

```

[[0.59831767  1.55378581]
 [0.95939365  1.67774359]]

```

Шаг для 1 кластера = 0.46

Веса после обновления:

```

[[ 0.59831767  1.55378581]
 [-0.72308084  1.19341266]]

```

Шаг для 0 кластера = 0.48

Веса после обновления:

```

[[-0.01426921  2.22633064]
 [-0.72308084  1.19341266]]

```

Шаг для 1 кластера = 0.45

Веса после обновления:

```

[[-0.01426921  2.22633064]
 [-0.61475555  1.60819071]]

```

Шаг для 1 кластера = 0.44

Веса после обновления:

```

[[-0.01426921  2.22633064]
 [ 0.38978472  2.0340259 ]]

```

Шаг для 0 кластера = 0.47

Веса после обновления:

```

[[1.61533939  2.122175  ]
 [0.38978472  2.0340259 ]]

```

Шаг для 0 кластера = 0.46

Веса после обновления:

```

[[-0.35653438  1.97235313]
 [ 0.38978472  2.0340259 ]]

```

Шаг для 1 кластера = 0.43

Веса после обновления:

```

[[-0.35653438  1.97235313]
 [-0.67498333  2.08236136]]

```

Шаг для 1 кластера = 0.42  
Веса после обновления:  
[[-0.35653438 1.97235313]  
[-0.28873331 2.05165358]]

Шаг для 1 кластера = 0.41  
Веса после обновления:  
[[-0.35653438 1.97235313]  
[-0.84222374 1.80612117]]

Шаг для 1 кластера = 0.4  
Веса после обновления:  
[[-0.35653438 1.97235313]  
[ 0.61516974 2.02912959]]

Шаг для 0 кластера = 0.45  
Веса после обновления:  
[[0.55106216 2.10730485]  
[0.61516974 2.02912959]]

Шаг для 0 кластера = 0.44  
Веса после обновления:  
[[0.77291138 1.71393177]  
[0.61516974 2.02912959]]

Шаг для 1 кластера = 0.39  
Веса после обновления:  
[[0.77291138 1.71393177]  
[1.35303761 2.10944821]]

Шаг для 0 кластера = 0.43  
Веса после обновления:  
[[1.54672194 1.68707408]  
[1.35303761 2.10944821]]

Шаг для 0 кластера = 0.42  
Веса после обновления:  
[[0.1777637 2.14993836]  
[1.35303761 2.10944821]]

Шаг для 1 кластера = 0.38  
Веса после обновления:  
[[ 0.1777637 2.14993836]  
[-0.2005104 2.06596913]]

Шаг для 1 кластера = 0.37  
Веса после обновления:  
[[0.1777637 2.14993836]  
[0.01191397 1.82473151]]

Шаг для 1 кластера = 0.36  
Веса после обновления:  
[[0.1777637 2.14993836]  
[0.59408339 1.85103482]]

Шаг для 1 кластера = 0.35  
Веса после обновления:  
[[ 0.1777637 2.14993836]  
[-0.34636722 2.24388926]]

Шаг для 1 кластера = 0.34  
Веса после обновления:  
[[0.1777637 2.14993836]  
[0.72885332 1.86865359]]

Шаг для 1 кластера = 0.33  
Веса после обновления:  
[[0.1777637 2.14993836]  
[0.73057478 2.4808189 ]]

Шаг для 0 кластера = 0.41  
Веса после обновления:  
[[0.81095714 1.56900662]  
[0.73057478 2.4808189 ]]

Шаг для 1 кластера = 0.32  
Веса после обновления:  
[[0.81095714 1.56900662]  
[1.37750858 2.13080659]]

Шаг для 1 кластера = 0.31  
Веса после обновления:  
[[0.81095714 1.56900662]  
[0.15752724 2.11598055]]

Шаг для 1 кластера = 0.3  
Веса после обновления:  
[[0.81095714 1.56900662]

[0.84350098 2.06834585]]

Шаг для 0 кластера = 0.4  
Веса после обновления:  
[[-0.02830133 1.78896455]  
[ 0.84350098 2.06834585]]

Шаг для 1 кластера = 0.29  
Веса после обновления:  
[[-0.02830133 1.78896455]  
[-0.34304717 1.86238968]]

Шаг для 1 кластера = 0.28  
Веса после обновления:  
[[-0.02830133 1.78896455]  
[ 0.25146797 2.01633314]]

Шаг для 0 кластера = 0.39  
Веса после обновления:  
[[0.11802973 1.98051745]  
[0.25146797 2.01633314]]

Шаг для 0 кластера = 0.38  
Веса после обновления:  
[[1.30452069 1.87524443]  
[0.25146797 2.01633314]]

Шаг для 0 кластера = 0.37  
Веса после обновления:  
[[-0.87440754 1.36501399]  
[ 0.25146797 2.01633314]]

Шаг для 1 кластера = 0.27  
Веса после обновления:  
[[-0.87440754 1.36501399]  
[-0.48757866 2.12353742]]

Шаг для 1 кластера = 0.26  
Веса после обновления:  
[[-0.87440754 1.36501399]  
[-1.01955364 2.20128602]]

Шаг для 0 кластера = 0.36  
Веса после обновления:  
[[ 0.38823946 1.93122379]  
[-1.01955364 2.20128602]]

Шаг для 1 кластера = 0.25  
Веса после обновления:  
[[ 0.38823946 1.93122379]  
[-0.17591526 1.95684231]]

Шаг для 1 кластера = 0.24  
Веса после обновления:  
[[0.38823946 1.93122379]  
[0.19214984 2.09163497]]

Шаг для 1 кластера = 0.23  
Веса после обновления:  
[[0.38823946 1.93122379]  
[0.64820366 2.0097532 ]]

Шаг для 1 кластера = 0.22  
Веса после обновления:  
[[ 0.38823946 1.93122379]  
[-0.23336537 1.69883582]]

Шаг для 0 кластера = 0.35  
Веса после обновления:  
[[ 0.00685161 2.29402551]  
[-0.23336537 1.69883582]]

Шаг для 0 кластера = 0.34  
Веса после обновления:  
[[-1.54748175 1.58479381]  
[-0.23336537 1.69883582]]

Шаг для 1 кластера = 0.21  
Веса после обновления:  
[[-1.54748175 1.58479381]  
[-0.74248 1.87788189]]

Шаг для 1 кластера = 0.2  
Веса после обновления:  
[[-1.54748175 1.58479381]  
[-0.91857074 1.94764795]]

Шаг для 0 кластера = 0.33  
Веса после обновления:

```
[[-1.4166548  1.99061384]
 [-0.91857074  1.94764795]]
```

Шаг для 0 кластера = 0.32  
Веса после обновления:  
[[-0.72731913 2.06684412]  
 [-0.91857074 1.94764795]]

Шаг для 1 кластера = 0.19  
Веса после обновления:  
[[-0.72731913 2.06684412]  
 [-0.9627162 2.11504338]]

Шаг для 0 кластера = 0.31  
Веса после обновления:  
[[-1.37871582 1.93349344]  
 [-0.9627162 2.11504338]]

Шаг для 0 кластера = 0.3  
Веса после обновления:  
[[-1.15364551 2.3429773 ]  
 [-0.9627162 2.11504338]]

Шаг для 0 кластера = 0.29  
Веса после обновления:  
[[-1.2767011 2.14578477]  
 [-0.9627162 2.11504338]]

Шаг для 0 кластера = 0.28  
Веса после обновления:  
[[-0.41999837 2.24041713]  
 [-0.9627162 2.11504338]]

Шаг для 1 кластера = 0.18  
Веса после обновления:  
[[-0.41999837 2.24041713]  
 [-0.53400463 2.06324554]]

Шаг для 0 кластера = 0.27  
Веса после обновления:  
[[-0.57124384 2.16462782]  
 [-0.53400463 2.06324554]]

Шаг для 0 кластера = 0.26  
Веса после обновления:  
[[-0.03004055 1.94265381]  
 [-0.53400463 2.06324554]]

Шаг для 1 кластера = 0.17  
Веса после обновления:  
[[-0.03004055 1.94265381]  
 [-0.14975001 2.08940544]]

Шаг для 1 кластера = 0.16  
Веса после обновления:  
[[-0.03004055 1.94265381]  
 [ 0.23121714 2.0696989 ]]

```
In [712]: s = X[0]
J = som.winner(weights, s)

print(f"Элемент принадлежит к {J} кластеру, на самом деле к {y[0]} кластеру")
print("Обученные веса: ")
print(weights)

Элемент принадлежит к 1 кластеру, на самом деле к 1 кластеру
Обученные веса:
[[-0.03004055  1.94265381]
 [ 0.23121714  2.0696989 ]]
```

```
In [713]: predicted = np.array([som.winner(weights, s) for s in X])
predicted
```

```
Out[713]: array([1, 1, 0, 1, 0, 0, 1, 1, 1, 0, 0, 1, 1, 0, 1, 0, 0, 0, 0, 1, 1,
        1, 0, 1, 0, 0, 0, 0, 1, 0, 1, 1, 0, 0, 0, 1, 1, 1, 0, 0, 0, 0, 1,
        1, 1, 1, 1, 1, 0, 1, 1, 1, 1, 0, 0, 1, 0, 0, 0])
```

```
In [714]: y == predicted
```

```
Out[714]: array([ True, False,  True, False, False,  True, False,  True,  True,
        False,  True,  True,  True, False,  True, False, False, False,
        True, False, False, False, False,  True,  True,  True, False,
        True,  True,  True, False,  True,  True,  True,  True, False,
        True,  True, False, False, False, False,  True,  True,  True,
        True,  True, False,  True,  True,  True,  True,  True, False,
        True,  True, False,  True,  True])
```

```
In [715]: from sklearn.metrics import accuracy_score
```

```
print(f'Точность кластеризации: {accuracy_score(y, predicted) * 100}%')
```

Точность кластеризации: 63.33333333333333%