

Липецкий государственный технический университет

Факультет автоматизации и информатики

Кафедра автоматизированных систем управления

ЛАБОРАТОРНАЯ РАБОТА №6

по дисциплине

«Прикладные интеллектуальные системы и экспертные системы»

Нейронные сети. Обучение без учителя

Студент

Мамедов Р. В.

Группа М-ИАП-23-1

Руководитель

Кургасов В. В.

доцент, канд. пед. наук

Липецк 2023 г.

Задание кафедры

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

Ход работы

Для генерации данных воспользуемся функцией `make_blobs` из пакета `sklearn.datasets`. После генерации данных визуализируем их с использованием библиотеки `matplotlib.pyplot`. Визуализация полученных данных представлена на рисунке 1.

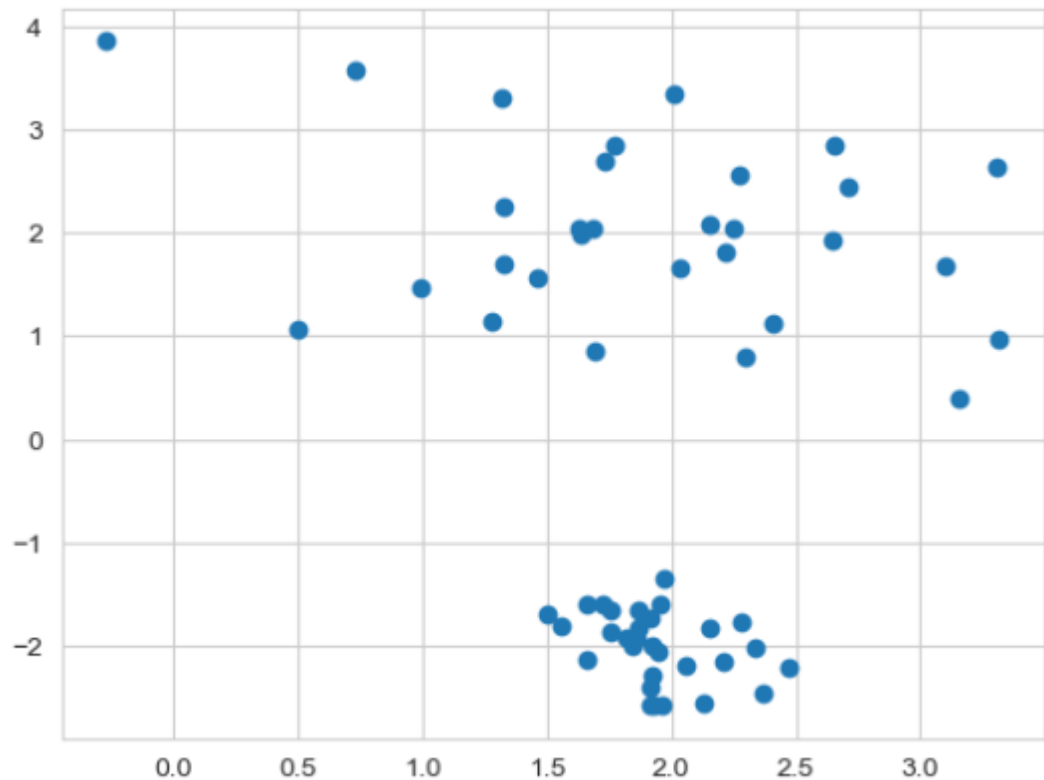


Рисунок 1 – График полученных данных

Выделим два кластера и обозначим их центры, полученный график представлен на рисунке 2.

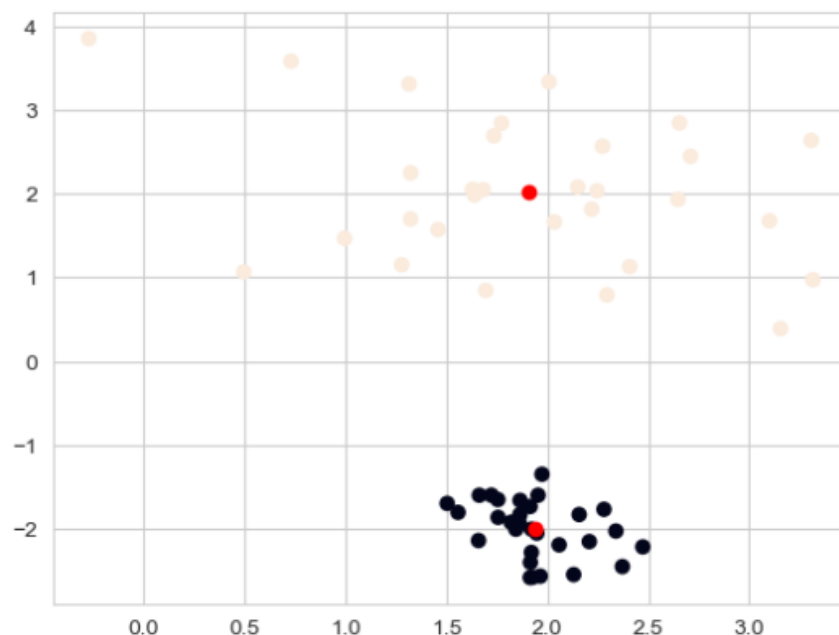


Рисунок 2 – График кластеров данных и их центры

Перемешиваем данные и последовательно будем подавать их на вход в нейронную сеть Кохонена. Для её работы необходимы веса, которые представлены на рисунке 3.

÷	0 ÷	1 ÷
0	1.708058	0.700536
1	2.025349	-0.398819

Рисунок 3 – Веса для работы нейронной сети

Далее будем обновлять эти веса после при выборе выигрышного нейрона. Веса после выбора нейрона представлены на рисунке 4. Итоговые веса после обучения представлены на рисунке 5.

```
Шаг для 1 кластера = 0.5
Веса после обновления:
[[ 1.10261378  1.07540902]
 [ 1.39010399 -0.7770354 ]]
```

```
Шаг для 0 кластера = 0.49
Веса после обновления:
[[ 1.70866206 -0.44326412]
 [ 1.39010399 -0.7770354 ]]
```

```
Шаг для 1 кластера = 0.48
Веса после обновления:
[[ 1.70866206 -0.44326412]
 [ 1.82442936 -0.02292649]]
```

```
Шаг для 0 кластера = 0.47
Веса после обновления:
[[ 2.46010455  1.00456066]
```

Рисунок 4 – Веса при начальных итерациях нейронной сети

```
Обученные веса:
[[ 1.70805786  0.7005361 ]
 [ 2.02534925 -0.39881894]]
```

Рисунок 5 - Итоговые обученные веса

Посмотрим на итоговую точность нейронной сети. Полученный результат представлен на рисунке 6.

```
1 from sklearn.metrics import accuracy_score
2
3 print(f'Точность нейронной сети: {accuracy_score(y, predicted) * 100}%')
Executed at 2023.12.17 23:26:17 in 19ms
```

Точность нейронной сети: 61.66666666666667%

Рисунок 6 – Итоговая точность нейронной сети

Вывод

В ходе выполнения данной лабораторной работы мною были получены навыки построения нейронной сети Кохонена с самообучения для решения задачи кластеризации.

После успешного построения и обучения модели была рассчитана характеристика точности кластеризации точек к их кластерам, которая составила 61,67%.