

Лабораторная работа №3.

Кластеризация векторов числовых данных с использованием слоя Кохонена в среде Matlab.

Цель: *изучить приемы работы с самообучающимися нейронными сетями и особенности их использования на примере слоя Кохонена.*

Исходные данные для работы:

1. Папка [“4m_file”](#). Содержит текст программы [cohonon.m](#) и файлы данных для его работы. В файле [all.spk](#) записаны 23 вектора, подлежащие автоматической классификации (кластеризации). Каждая колонка содержит 128 чисел и представляет собой один из векторов. Первые 8 векторов относятся к первому классу, следующие 4 – ко второму, следующие 5 – к третьему, последние 6 – к четвертому. В файле [all.ini](#) записаны векторы минимальных и максимальных значений для каждой из 128 переменных вектора входов. Минимальное значение везде принято равным 0, а максимальное – 1.

2. Папка [“4NNet”](#). Содержит исходные данные в формате, пригодном для импорта при помощи утилиты nntool.

3. Папка [“Source”](#). Содержит входные векторы для кластеризации, разделенные по классам. Они расположены по одному вектору в файле (всего – 23 вектора, по классам – 8, 4, 5, 6 векторов), и приведены для справки.

Порядок выполнения работы.

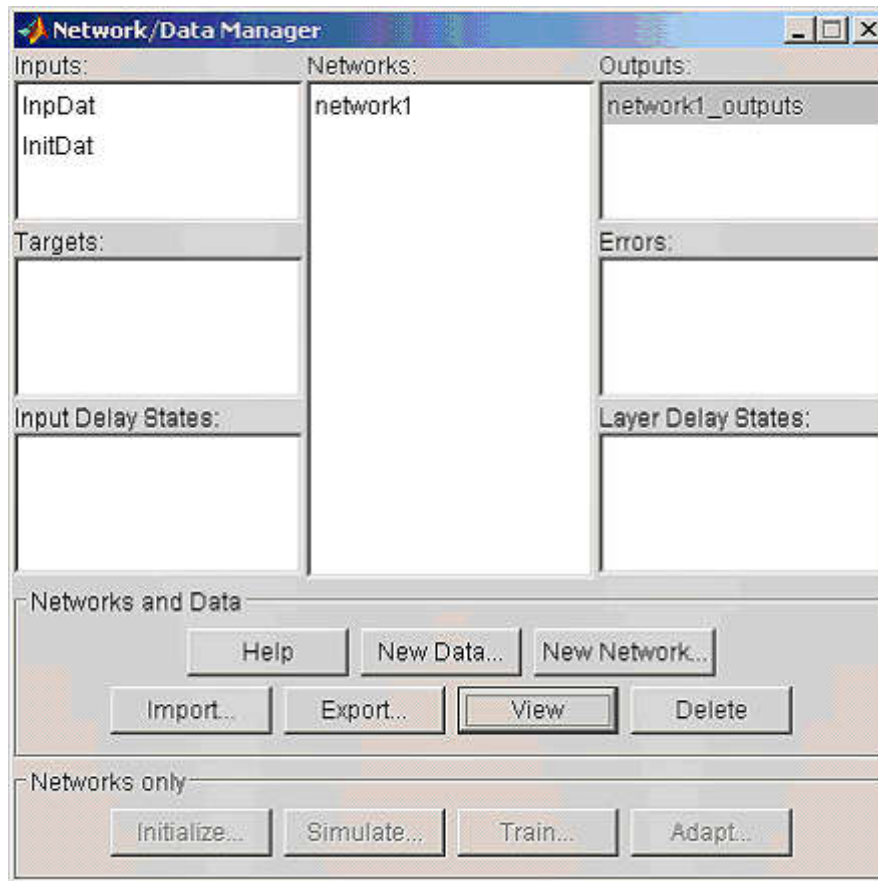
Часть первая.

1. Запустить среду Matlab.
 2. Переписать содержимое папки “4m_file” в какую либо папку, не содержащую длинных имен в своем пути, например, “Z:\NeuroNet\lab3”. Это необходимо для правильной работы Matlab.
 3. Открыть файл [cohonon.m](#), Разобраться в работе программы, исправить ее и запустить на выполнение. Просмотреть результаты работы можно в основном окне MatLab в подокне Command Window.
 4. Запустить программу повторно, сравнить результаты работы. Объяснить полученные результаты.
- Пример результатов выполнения cohonon.m

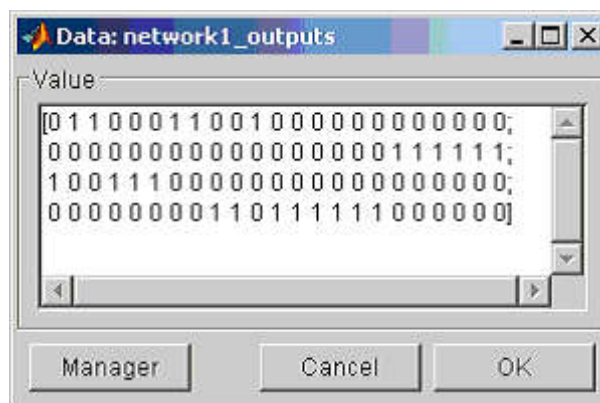
```
ac =
Columns 1 through 9
3 2 2 3 2 2 2 2 3
Columns 10 through 18
3 3 3 4 4 4 4 4 1
Columns 19 through 23
1 1 1 1 1
```

Часть вторая.

5. Переписать содержимое папки [“4NNet”](#) в какую либо папку, не содержащую длинных имен и имен со спецсимволами (пробелы, скобки и т.п.) в своем пути.
6. В главном окне Matlab выполнить команду “nntool” или в Launch Pad открыть Neural Network Toolbox -> NNTool. В результате откроется окно визуального инструмента для работы с нейронными сетями в Matlab.



7. При помощи команды Import импортировать в nntool данные из файлов [data.mat](#) (переменная InpDat) и [init.dat](#) (переменная InitDat) как входные данные (Inputs), кнопка Load.
8. При помощи команды New Network создать сеть типа Competitive с количеством нейронов, равным количеству различаемых классов. Диапазоны входных значений взять из переменной InitDat.
9. При помощи команды Train обучить полученную сеть на векторах, взятых из переменной InpDat. Количество эпох взять произвольно от 50 до 300.
10. При помощи команды Simulate смоделировать работу сети на векторах, взятых из переменной InpDat.
11. Просмотреть выходы сети network1_outputs при помощи команды View. Сравнить результаты с полученными в первой части работы. Объяснить полученные результаты.



Часть третья (факультативная).

12. Провести корреляционный анализ исходных данных и обосновать их разбиение на четыре различных класса.

Отчет должен содержать:

1. Исправленную версию программы cohonen.m.
2. Результаты обоих запусков программы.
3. Результаты работы сети, смоделированной при помощи утилиты nntool.

4. Анализ результатов, полученных в первой и второй частях задания.