

ЛАБОРАТОРНАЯ РАБОТА №1

«Кластеризация с помощью нейронных сетей»

Цель лабораторной работы: освоить основные принципы решения задачи кластеризации с использованием нейронных сетей со слоем Кохонена и самоорганизующихся карт.

Задание: Используя встроенные функции пакета нейронных сетей математической среды Matlab, построить нейронную сеть со слоем Кохонена, которая множество входных данных разделит на кластеры и выявит их центры. На обученную сеть подать новый входной вектор и определить, к какому кластеру он относится.

Теоретические сведения

Самоорганизующиеся карты. Самоорганизующиеся карты (Self Organizing Maps - SOM) это одна из разновидностей нейросетевых алгоритмов. Основным отличием данной технологии от рассмотренных нами ранее нейросетей, обучаемых по алгоритму обратного распространения, является то, что при обучении используется метод обучения без учителя, то есть результат обучения зависит только от структуры входных данных. Нейронные сети ДЕННОГО типа часто применяются для решения самых различных задач, от восстановления пропусков в данных до анализа данных и поиска закономерностей, например, в финансовой задаче.

Основы самоорганизующихся карт. Алгоритм функционирования самообучающихся карт представляет собой один из вариантов кластеризации многомерных векторов. Примером таких алгоритмов может служить алгоритм ближайших средних (c-means). Важным отличием алгоритма SOM является то, что в нем все нейроны (узлы, центры классов) упорядочены в некоторую структуру (обычно двумерную сетку). При этом в ходе обучения модифицируется не только нейрон-победитель, но и его соседи, но в меньшей степени. За счет этого SOM можно считать одним из методов проецирования многомерного пространства в пространство с более низкой размерностью. При использовании этого алгоритма вектора, схожие в исходном пространстве, оказываются рядом и на полученной карте.

Структура самоорганизующихся карт. SOM подразумевает использование упорядоченной структуры нейронов. Обычно используются одно- и двумерные сетки. При этом каждый нейрон представляет собой n -мерный вектор-столбец $w \sim [w_1, w_2, \dots, w_n]$, где n определяется размерностью исходного пространства (размерностью входных векторов). Применение одно- и двумерных сеток связано с тем, что возникают проблемы при отображении пространственных структур большей размерности (при этом опять возникают проблемы с понижением размерности до двумерной, представимой на мониторе).

Обычно нейроны располагаются в узлах двумерной сетки с прямоугольными или шестиугольными ячейками. При этом, как было сказано выше, нейроны также взаимодействуют друг с другом. Величина этого взаимодействия определяется расстоянием между нейронами на карте.

Начальная инициализация карты. При реализации алгоритма SOM заранее задается конфигурация сетки (прямоугольная или шестиугольная), а также количество нейронов в сети. Некоторые источники рекомендуют использовать максимально возможное количество нейронов в карте. При этом начальный радиус обучения (neighborhood в англоязычной литературе) в значительной степени влияет на способность обобщения при помощи, полученной карты. В случае, когда количество узлов карты превышает количество примеров в обучающей выборке, то успех использования алгоритма в большой степени зависит от подходящего выбора начального радиуса обучения. Однако в случае, когда размер карты составляет десятки тысяч нейронов, время, требуемое на обучение карты, обычно бывает слишком велико для решения практических задач, таким образом, необходимо достигать допустимого компромисса при выборе количества узлов.

Перед началом обучения карты необходимо проинициализировать весовые коэффициенты нейронов. Удачно выбранный способ инициализации может существенно ускорить обучение и привести к получению более качественных результатов. Существуют три способа инициализации начальных весов:

1. инициализация случайными значениями, когда всем весам даются малые случайные величины;
2. инициализация примерами, когда в качестве начальных значений задаются значения случайно выбранных примеров из обучающей выборки;
3. линейная инициализация. В этом случае веса инициализируются значениями векторов, линейно упорядоченных вдоль линейного подпространства, проходящего между двумя главными собственными векторами исходного набора данных. Собственные векторы могут быть найдены, например, при помощи процедуры Грама-Шмидта.

Обучение самоорганизующихся карт. Обучение состоит из последовательности коррекций векторов, представляющих собой нейроны. На каждом шаге обучения из исходного набора данных случайно выбирается один из векторов, а затем производится поиск наиболее похожего на него вектора коэффициентов нейронов. При этом выбирается нейрон-победитель, который наиболее похож на вектор входов. Под похожестью в данной задаче понимается расстояние между векторами, обычно вычисляемое в евклидовом пространстве. После того, как найден нейрон-победитель, производится корректировка весов нейро-сети. При этом вектор, описывающий нейрон-победитель, и векторы, описывающие его соседей в сетке, перемещаются в направлении входного вектора.

Методика выполнения лабораторной работы

Задача. Используя встроенные функции пакета нейронных сетей математической среды Matlab, построить нейронную сеть со слоем Кохонена, которая множество входных данных разделит на кластеры и выявит их центры. На обученную сеть подать новый входной вектор и определить, к какому кластеру он относится.

Для создания нейронной сети со слоем Кохонена воспользуемся встроенной в среду Matlab функцией `newc`:

1.

```
X=[0 1; 0 1];
clusters=5;
points=5;      %Задание количества точек в кластере
std_dev=0.01;
p=nnngenc(X,clusters,points,std_dev);%Моделирование входных данных
h=newc([0 1;0 1],5,.1);      % создание слоя Кохонена
h.trainParam.epochs=50; %Задание количества циклов обучения
h=init(h);
h=train(h,p);
w=h.IW{1}; % вывод графиков исходных данных и выявленных центров кластеров
plot(p(1,:),p(2,:),'^r'),grid;
hold on;
plot(w(:,1),w(:,2),'ob');
xlabel('p(1)');
ylabel('p(2)');
A=0.6
B=0.5
p=[A;B];
plot(A,B,'^k')
y=sim(h,p) % Опрос сети
A =0.6000
```

$B = 0.5000$

$y = (2, 1)$

Результат работы программы представлен на **рис. 1**. Кроме того, его можно увидеть в командном окне: $Y = (2, 1)$

Предъявленный вектор отнесен ко второму кластеру. Теперь данный алгоритм применим к реальной (хоть и элементарной) задаче кластеризации. На вход нейронной сети будем подавать данные весоростовых показателей людей и попробуем выявить три класса (кластера):

1) нормальный весоростовой показатель;

2) избыток веса;

3) недостаток веса.

2. %входные данные (первая строка матрицы - рост; вторая - вес)

```
p=[175 180 182 175 183 176 183 176 183 176 175 180 178 180 178 182 178 182 179 174 172 179;
```

```
70 75 100 99 42 48 76 72 40 45 92 96 70 69 95 90 79 82 80 50 96 91] %создаем НС Кохонена с 3 кластерами (нормальный весоростовой показатель, избыток веса и недостаток веса)
```

```
h=newc([0 200;0 100],3,.1)
```

```
h.trainParam.epochs=500; %Задание количества циклов обучения
```

```
h=train(h,p)
```

```
w=h.IW{1};
```

```
plot(p(1,:),p(2:,:),'^r');
```

```
hold on;
```

```
plot(w(:,1),w(:,2),'ob');
```

```
xlabel('Rost');
```

```
ylabel('Ves');
```

```
% Задание нового входного вектора
```

```
%Опрос сети
```

```
A=181
```

```
B=65
```

```
p=[A;B];
```

```
plot(A,B,'+r')
```

```
y=sim(h,p)
```

```
A=181
```

```
B=65
```

```
y=(2,1)
```

Результат работы программы представлен на **рис. 2**. Кроме того, его можно увидеть в командном окне: $y = (2, 1)$

Предъявленный вектор отнесен ко второму кластеру.

Теперь рассмотрим использование самоорганизующей карты на примере двумерных векторов. Используя самоорганизующиеся карты, двумерные векторы разбить на кластеры и выявить их центры, затем подать на вход самоорганизующей карты новый вектор и определить кластер, к которому он относится.

3.

```
P=rand(2,100) %Задание случайных двумерных входных векторов
```

```
figure(1)
```

```
hold on
```

```
plot(P(1,:),P(2,:),'+r') %визуальное изображение входных векторов
```

```
%Создание НС с 3*4 нейронами
```

```
%По умолчанию функция TFCN = 'hextop', то есть нейроны располагаются в узлах двумерной сетки с шестиугольными ячейками
```

```
net=newsom([0 1;0 1],[3 4]);
```

```
net.trainParam.epoch=1 %Задание числа циклов настройки
```

```

net=train(net,P)      % настройка сети
A=0.5
B=0.3
p=[A;B];           % Задание нового входного вектора
plot(A,B,'^k') % прорисовка на рисунке входного вектора (черный треугольник)
figure(2)
plotsom(net.iw{1,1},net.layers{1}.distances)
a=sim(net,p) % опрос сети
A = 0.5000
B = 0.3000
a = (5,1) 1

```

Результат работы программы представлен на **рис. 3 - 4.**

Результат работы программы можно увидеть и в командном окне: $a = (5,1) 1$

Предъявленный вектор отнесен к пятому кластеру.

Контрольные вопросы

1. Что понимается под кластеризацией? Задача кластеризации (категоризации, классификации "без учителя") - задача размещения входных векторов (образов) по категориям (кластерам) так, чтобы близкие векторы (схожие образы) оказались в одной категории. Отличие задачи кластеризации от похожей на нее задачи классификации заключается в том, что набор категорий изначально не задан и определяется в процессе обучения нейронной сети. Примером задачи кластеризации служит задача сжатия информации путем уменьшения разнообразия данных.

Кластеризация может быть использована для решения следующих задач:

- обработки изображения;
- классификации;
- тематического анализа коллекций документов;
- построения репрезентативной выборки.

Методы кластеризации при помощи нейронных сетей являются развитием классических методов кластеризации. Например, метод кластеризации векторов с помощью сети Кохонена содержит в своей основе метод К средних. В то же время нейронные сети являются гораздо более гибким инструментом в применении к данным, имеющим большой объем и избыточную размерность.

Закон обучения Кохонена

На рис. 1 приведена базовая структура слоя Кохонена. Слой состоит из N обрабатывающих элементов, каждый из которых получает p входных сигналов X_1, X_2, \dots, X_p , из нижестоящего слоя, который является прямым передатчиком сигналов. Входу x_i и связи (ij) припишем вес w_{ij} .

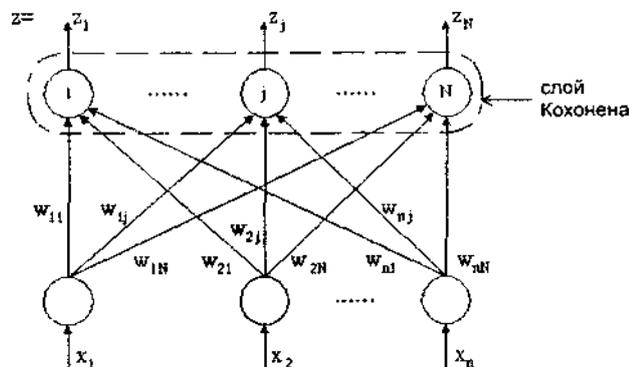


Рис. 1. Нейронная сеть со слоем Кохонена

2. Опишите встроенные операторы Matlab для кластеризации.

`newsc` – создание конкурентного слоя.

$Net=newc(PR,S,KLR,CLR)$ – функция создания слоя Кохонена.

Аргументы функции:

PR – $R \times 2$ матрица минимальных и максимальных значений для R входных элементов,

S – число нейронов,

KLR – коэффициент обучения Кохонена (по умолчанию 0,01)

CLR – Коэффициент «справедливости»(по умолчанию 0,001)

Опишите сеть Кохонена.

3. Зачем используются самоорганизующиеся карты? Самоорганизующиеся карты (Self Organizing Maps - SOM) это одна из разновидностей нейросетевых алгоритмов.

Нейронные сети ДЕННОГО типа часто применяются для решения самых различных задач, от восстановления пропусков в данных до анализа данных и поиска закономерностей, например, в финансовой задаче.

4. Опишите отличие сети Кохонена от SOM. Основным отличием данной технологии от рассмотренных нами ранее нейросетей, обучаемых по алгоритму обратного распространения, является то, что при обучении используется метод обучения без учителя, то есть результат обучения зависит только от структуры входных данных.

ЛАБОРАТОРНАЯ РАБОТА №2 **«Нейронные сети для распознавания образов»**

Цель работы: реализация алгоритмов распознавания образов с помощью нейронных сетей на языке программирования и изучение их свойств.

Последовательность проведения лабораторной работы:

1. Изучить устройство и работу однослойной, двухслойной нейронных сетей, сети встречного распространения.
2. Выбрать исходный алфавит классов (цифры, буквы русского алфавита, буквы английского алфавита, буквы греческого алфавита, знаки препинания и арифметические символы, радиоэлементы, геометрические фигуры, узоры, дорожные знаки, топологические элементы, картографические обозначения) по согласованию с преподавателем.
3. Составить программу, реализующую однослойную, двухслойную нейронные сети и сеть встречного распространения для задачи распознавания. Обучить нейронные сети.
4. Программное обеспечение должно позволять просматривать эталоны (классы) изображений, а также распознаваемые изображения; записывать эталоны изображений в библиотеку на диск, записывать входной образ на диск; позволять редактировать входной образ, сохранять веса нейронов при обучении сети.
5. Испытать программное обеспечение для различных входных данных.
6. Произвести оценку качества распознавания для различных случаев, систематизировав полученные результаты в таблицы. Построить графики выявленных зависимостей, сделать выводы.
7. Результаты работы оформить в виде отчета в текстовом редакторе.

Рекомендации по созданию программного обеспечения

После изучения теоретического материала, и выбора исходных изображений, системы признаков, необходимо определить количество нейронов в слоях. Максимальное количество нейронов первого слоя может совпадать с количеством признаков, хотя теоретически избыточность допускается, но не является рациональным подходом. Желательно, при создании программного обеспечения использовать объектно-ориентированный язык, чтобы была возможность динамически варьировать количество нейронов в слоях – порождать соответствующее количество экземпляров класса и количество слоев.

Рекомендации по исследованиям

При исследовании нейронных сетей необходимо проделать следующие эксперименты по выявлению качества распознавания с помощью созданного программного обеспечения.

Эксперимент №1. Исследование влияния отклонения изображения от эталона (в разных точках изображения) на качество распознавания. В качестве исходного распознаваемого образа берется один из эталонов и подправляется таким образом, чтобы отсутствовал один, два и т. д. пикселей в изображении эталона. Эксперимент проводится для нескольких случаев (отсутствие пикселей в разных участках изображения) для всех эталонов. При этом сравниваются различные нейронные сети.

Эксперимент №2. Исследование влияния отклонений в виде шума одного, двух, трех и т. д. пикселей в изображении на качество распознавания. В качестве исходного распознаваемого образа берется один из эталонов и добавляется один или несколько пикселей шума.

Эксперимент проводится для различного расположения шума и для различных эталонов. В ходе эксперимента также сравниваются различные нейронные сети.

Эксперимент №3. Исследование влияния наличия шума и отклонений в изображении в виде одного, двух, трех и т. д. пикселей в изображении на качество распознавания. В качестве исходного распознаваемого образа берется один из эталонов. В изображение распознаваемого образа вносится шум в виде нескольких пикселей и удаляется несколько пикселей в изображении символа. Эксперимент повторяется для различного расположения шума и отклонений, и для разных эталонов на различных типах нейронных сетей. Данный эксперимент является комбинацией первых двух.

Эксперимент №4. Исследование влияния наличия черной строки или столбца в изображении (как помеха в образе) на качество распознавания. В качестве исходного распознаваемого образа берется один из эталонов. В изображение вносится черная строка или столбец. Эксперимент повторяется для различного положения строки или столбца в изображении и для различных эталонов. В ходе эксперимента сравниваются различные нейронные сети.

Эксперимент №5. Исследование влияния наличия белой строки или столбца в изображении (как помеха в образе) на качество распознавания. В качестве исходного распознаваемого образа берется один из эталонов. В изображение вносится белая строка или столбец. Эксперимент повторяется для различного положения строки или столбца в изображении и для различных эталонов на различных нейронных сетях.

Эксперимент №6. Исследование влияния количества нейронов в слоях на качество распознавания. В качестве исходного распознаваемого образа берется один из эталонов. Количество нейронов в слое варьируется от двух до равного числу признаков (для двухслойных сетей слои варьируются последовательно). Эксперимент повторяется на различных нейронных сетях.

Эксперимент №7. Исследование влияния количества нейронов в слоях и количества эталонов на скорость обучения сети. В ходе проведения эксперимента количество нейронов в слое варьируется от двух до равного числу признаков (для двухслойных сетей слои варьируются последовательно), также варьируется количество эталонов (в пределах выбранных по согласованию с преподавателем). В каждом случае фиксируется число итераций и время обучения. Эксперимент повторяется на различных нейронных сетях.

Эксперимент №8. Исследование влияния начертания входных образов на качество распознавания. В качестве исходного распознаваемого образа берется один из эталонов. Эталон модифицируется (делается жирное или наклонное, или подчеркнутое начертание). Эксперимент повторяется на различных нейронных сетях для различных видоизменений.

Примечание. При использовании цвета в изображении эксперименты №4 и №5 следует проводить для цвета фона и цвета образа. В эксперименте №4 вместо черной строки или столбца берется столбец или строка цвета образа. В эксперименте №5 вместо белой строки или столбца берется строка или столбец фоновой цвета.

Рекомендации по оформлению отчета

Результаты экспериментов следует оформить в виде таблиц и графиков с пояснениями и подписями. На графиках следует показать зависимость ошибки распознавания для различных нейронных сетей от количества и вида помех, вносимых в изображение, а также от варьируемых параметров.

В качестве таблицы значений можно использовать таблицу следующего вида

Таблица 3.

Численные значения экспериментов

Изображение входного образа	Значение СКО	Нейронная сеть		
		Односло йная	Двухсло йная	Кохонена - Гроссбер га

--	--	--	--	--

Заполняйте две копии этой таблицы – одну для эталонов, другую для предъявляемых изображений, записывая в колонки значения выхода нейронной сети.

В отчете также следует привести данные об обучении нейронной сети: количество итераций, первоначальные и конечные значения весов.

Контрольные вопросы к лабораторной работе

1. Каковы основные понятия теории распознавания?
2. Дайте определение класса образов.
3. Что такое алфавит классов?
4. Дайте определение объекта класса образов.
5. Дайте определение признака класса образов.
6. Какие типы признаков вы знаете? Приведите примеры.
7. Что такое нейронная сеть?
8. Что такое синапс?
9. Что такое аксон?
10. Что определяет уровень активации нейрона?
11. Дайте определение активационной функции.
12. Какие типы активационных функций Вам известны?
13. Что такое персептрон?
14. В чем преимущество сигмоидальной функции?
15. В чем заключается проблема функции «исключающее или»?
16. В чем заключается цель обучения нейронной сети?
17. Что такое обучающая пара?
18. Что такое обучающее множество?
19. Расскажите алгоритм обучения персептрона.
20. Что такое дельта-правило?
21. Перечислите шаги процедуры обратного распространения.
22. Какие действия выполняются при проходе вперед?
23. Какие действия выполняются при обратном проходе?
24. Какие недостатки есть у процедуры обратного распространения?
25. Опишите устройство сети встречного распространения.
26. Как устроен и работает слой Кохонена?
27. Как устроен и работает слой Гроссберга?
28. В чем заключается проблема выбора начальных значений весовых векторов?
29. Как решают проблему выбора начальных значений весовых векторов?