

## Лабораторная работа № 7.

### Автоассоциативные сети с узким горлом

*Целью работы* является исследование свойств автоассоциативных сетей с узким горлом, алгоритмов обучения, а также применение сетей для выполнения линейного и нелинейного анализа главных компонент набора данных.

#### Основные этапы работы:

1. Использовать автоассоциативную сеть с узким горлом для отображения набора данных, выделяя первую главную компоненту данных.
2. Использовать автоассоциативную сеть с узким горлом для аппроксимации кривой на плоскости, выделяя первую нелинейную главную компоненту данных.
3. Применить автоассоциативную сеть с узким горлом для аппроксимации пространственной кривой, выделяя старшие нелинейные главные компоненты данных.

#### Сценарий работы:

1. Задан обучающий набор  $\{x_i, y_i\}$ ,  $i = 1, \dots, N$ . Построить автоассоциативную сеть с узким горлом, реализующую метод главных компонент. С помощью сети восстановить набор данных, учитывая информацию только о первой главной компоненте.

1.1 В соответствии с вариантом задания сгенерировать обучающее множество. Выполнить преобразование множества с помощью функции *con2seq*. Не выделять из обучающего множества контрольное и тестовое подмножества.

1.2 Создать линейную многослойную сеть прямого распространения с помощью функции *feedforwardnet*. Число нейронов скрытого слоя задать равным 1. Использовать активационные функции *purelin* для скрытого и выходного слоев. Задать метод Левенберга-Марквардта в качестве алгоритма обучения. Сконфигурировать сеть (*configure*) под обучающее множество.

1.3 Инициализировать (*init*) весовые коэффициенты и смещения сети с помощью функции, заданной по умолчанию.

1.4 Задать параметры обучения: число эпох обучения (*net.trainParam.epochs*) равным 100, предельное значение критерия обучения (*net.trainParam.goal*) равным  $10^{-5}$ .

1.5 Выполнить обучение сети с помощью функции *train*. Для обучения использовать обучающую выборку. Занести в отчет содержимое окон Performance и Neural Network Training.

1.6 Отобразить структуру сети и проведенное обучение в отчете, заполнив таблицу 1.

1.7 Рассчитать выход сети для обучающего множества.

1.8 Отобразить обучающее множество и выход сети с параметрами  $-r$  и  $-b$  соответственно и *Linewidth* равным 2 в области  $[-1, 1] \times [-1, 1]$  (*axis*). График занести в отчет.

#### Варианты заданий:

Номер варианта соответствует номеру студента в списке группы. Обучающее множество представляет собой алгебраическую линию или геометрическую фигуру. Для генерации точек использовать параметрическое уравнение алгебраической линии в канонической системе координат.

$$\begin{aligned}t &= 0 : h : 2\pi \\x &= f(t) \\y &= g(t)\end{aligned}$$

Константы  $a$  и  $b$  задают большую и малую полуоси эллипса. Параметры преобразования прямоугольной системы координат на плоскости: угол поворота ( $\alpha$ ) и координаты параллельного переноса ( $x_0, y_0$ ). Константы  $d_1$  и  $d_2$  задают длины сторон прямоугольника. При построении прямоугольника считать, что точка пересечения диагоналей находится в начале системы координат.

№	Обучающее множество
1.	Эллипс: $a = 0.7, b = 0.4, \alpha = \pi/3, x_0 = 0.2, y_0 = -0.4$
2.	Эллипс: $a = 0.6, b = 0.2, \alpha = -\pi/3, x_0 = -0.1, y_0 = 0$
3.	Эллипс: $a = 0.6, b = 0.6, \alpha = -\pi/3, x_0 = -0.1, y_0 = 0$
4.	Прямоугольник: $d_1 = 0.1, d_2 = 0.8, \alpha = \pi/8, x_0 = -0.2, y_0 = 0.1$
5.	Эллипс: $a = 0.6, b = 0.6, \alpha = 0, x_0 = 0, y_0 = 0$
6.	Прямоугольник: $d_1 = 0.3, d_2 = 0.8, \alpha = -\pi/4, x_0 = 0.5, y_0 = -0.4$
7.	Эллипс: $a = 0.7, b = 0.2, \alpha = -\pi/6, x_0 = 0, y_0 = -0.1$
8.	Эллипс: $a = 0.7, b = 0.7, \alpha = -\pi/6, x_0 = 0, y_0 = -0.1$
9.	Прямоугольник: $d_1 = 0.6, d_2 = 0.2, \alpha = 0, x_0 = 0, y_0 = 0$
10.	Прямоугольник: $d_1 = 0.1, d_2 = 0.8, \alpha = \pi/4, x_0 = 0.4, y_0 = 0.4$
11.	Прямоугольник: $d_1 = 0.5, d_2 = 0.3, \alpha = 0, x_0 = -0.5, y_0 = 0.1$
12.	Эллипс: $a = 0.6, b = 0.9, \alpha = \pi/8, x_0 = 0.2, y_0 = -0.1$
13.	Эллипс: $a = 0.4, b = 0.3, \alpha = 0, x_0 = 0, y_0 = 0$
14.	Прямоугольник: $d_1 = 0.3, d_2 = 0.8, \alpha = -\pi/2, x_0 = 0.2, y_0 = -0.1$
15.	Эллипс: $a = 0.5, b = 0.4, \alpha = 0, x_0 = 0.3, y_0 = -0.1$
16.	Прямоугольник: $d_1 = 0.5, d_2 = 0.65, \alpha = -\frac{3\pi}{2}, x_0 = -0.25, y_0 = -0.55$
17.	Прямоугольник: $d_1 = 0.2, d_2 = 0.3, \alpha = \pi, x_0 = 0.5, y_0 = 0.5$

№	Обучающее множество
18.	Эллипс: $a = 0.1, b = 0.6, \alpha = 0, x_0 = 0, y_0 = 0.1$
19.	Прямоугольник: $d_1 = 0.7, d_2 = 0.75, \alpha = \pi/3, x_0 = -0.1, y_0 = -0.4$
20.	Прямоугольник: $d_1 = 0.3, d_2 = 0.5, \alpha = \pi/3, x_0 = 0.4, y_0 = -0.2$
21.	Эллипс: $a = 0.5, b = 0.5, \alpha = \pi/6, x_0 = 0.2, y_0 = -0.1$
22.	Эллипс: $a = 0.3, b = 0.7, \alpha = \pi/4, x_0 = 0, y_0 = 0$
23.	Прямоугольник: $d_1 = 0.7, d_2 = 0.4, \alpha = -\pi/6, x_0 = 0.5, y_0 = 0$
24.	Прямоугольник: $d_1 = 0.3, d_2 = 0.5, \alpha = -\pi/8, x_0 = 0.2, y_0 = -0.1$
25.	Прямоугольник: $d_1 = 0.4, d_2 = 0.5, \alpha = -\pi/3, x_0 = 0, y_0 = -0.5$
26.	Эллипс: $a = 0.1, b = 0.6, \alpha = \pi/3, x_0 = 0.2, y_0 = 0.1$
27.	Прямоугольник: $d_1 = 0.7, d_2 = 0.4, \alpha = \pi/3, x_0 = 0.3, y_0 = -0.25$
28.	Эллипс: $a = 0.6, b = 0.6, \alpha = 0, x_0 = -0.3, y_0 = 0.3$
29.	Эллипс: $a = 0.3, b = 0.8, \alpha = 0, x_0 = -0.5, y_0 = 0.4$
30.	Эллипс: $a = 0.5, b = 0.4, \alpha = \pi/8, x_0 = 0.3, y_0 = -0.1$

## Этап 2

2. Задан обучающий набор  $\{x_i, y_i\}$ ,  $i = 1, \dots, N$ . Точки набора лежат на плоской кривой. Построить автоассоциативную сеть с узким горлом, реализующую нелинейный метод главных компонент. С помощью сети выполнить аппроксимацию кривой, выделяя первую нелинейную главную компоненту.

2.1 В соответствии с вариантом задания сгенерировать обучающее множество. Выполнить преобразование множества с помощью функции *con2seq*. Не выделять из обучающего множества контрольное и тестовое подмножества.

2.2 Создать многослойную сеть прямого распространения с помощью функции *feedforwardnet*. Число нейронов скрытого слоя задать равным  $[10, 1, 10]$ . Использовать активационные функции *tansig* для трех скрытых слоев и *purelin* для выходного слоя. Задать метод Левенберга-Марквардта в качестве алгоритма обучения. Сконфигурировать сеть (*configure*) под обучающее множество.

2.3 Инициализировать (*init*) весовые коэффициенты и смещения сети с помощью функции, заданной по умолчанию.

2.4 Задать параметры обучения: число эпох обучения (*net.trainParam.epochs*) равным 2000, предельное значение критерия обучения (*net.trainParam.goal*) равным  $10^{-5}$ .

2.5 Выполнить обучение сети с помощью функции *train*. Для обучения использовать обучающую выборку. Созданная сеть чувствительна к инициализации, поэтому иногда необходимо провести обучение несколько раз. Если результаты аппроксимации неудовлетворительные, то следует увеличить число эпох обучения или число нейронов в скрытых слоях, но при этом число нейронов в узком горле сети должно остаться равным 1. Занести в отчет содержимое окон Performance и Neural Network Training.

2.6 Отразить структуру сети и проведенное обучение в отчете, заполнив таблицу 1.

2.7 Рассчитать выход сети для обучающего множества.

2.8 Отобразить обучающее множество и выход сети с параметрами  $-r$  и  $-b$  соответственно и *Linewidth* равным 2. График занести в отчет.

### Варианты заданий:

Номер варианта соответствует номеру студента в списке группы. Кривая задается полярным уравнением. Генерацию точек точек кривой проводить по формулам:

$$\varphi \in [0, 2\pi] \text{ с шагом } h = 0.025$$

$$x = r \cos(\varphi)$$

$$y = r \sin(\varphi)$$

$\varphi \in [0, 2\pi]$  если в задании специально не указан отрезок.

№	Полярные уравнения
1.	$r = \frac{\varphi}{2\pi}$
2.	$r = e^{\varphi}$
3.	$r = -\varphi + 1$
4.	$r = \frac{2}{\varphi} + 4, \quad \varphi \in [0.01, 2\pi]$
5.	$r = \varphi + 3$

№	Полярные уравнения
6.	$r = e^{\varphi} + 2\varphi$
7.	$r = \varphi^2;$
8.	$r = 2\varphi$
9.	$r = 5 \operatorname{ctg} \varphi, \quad \varphi \in [0.01, \pi]$
10.	$r = \frac{2 \sin \varphi}{\varphi}, \quad \varphi \in [0.01, \pi]$
11.	$r = 5, \quad \varphi \in [0.01, \frac{11\pi}{6}]$
12.	$r = \sqrt{\varphi}$
13.	$r = \frac{\pi}{\frac{\varphi}{\cos \frac{\varphi}{3}}}, \quad \varphi \in [0.01, \pi]$
14.	$r = 2\varphi^2$
15.	$r = -3\varphi^2 + 1$
16.	$r = \left(\frac{1}{2}\right)^{\varphi}$
17.	$r = 2 \cos \varphi, \quad \varphi \in [0.01, \frac{11\pi}{6}]$
18.	$r = \frac{2(\pi - 2\varphi)}{\pi \cos \varphi}, \quad \varphi \in [0.01, \pi]$
19.	$r = \operatorname{ctg} \frac{47}{\varphi}, \quad \varphi \in [0.01, \pi]$
20.	$8 \sin \varphi$

№	Полярные уравнения
21.	$r = \frac{1}{\sqrt{\varphi}}, \quad \varphi \in [0.01, 2\pi]$
22.	$r = 2^\varphi$
23.	$r = \frac{1}{\cos \frac{\varphi}{3}}, \quad \varphi \in [0.01, \pi]$
24.	$r = \cos^2 \frac{\varphi}{2}, \quad \varphi \in [0.01, \pi]$
25.	$r = \frac{1 - \sin \varphi}{\cos \varphi}, \quad \varphi \in [0.01, \frac{11\pi}{12}]$
26.	$r = \sqrt{\varphi} + 5\varphi$
27.	$r = \varphi^2 + 3\varphi$
28.	$r = \frac{2(1 - \sin \varphi)}{\cos \varphi}, \quad \varphi \in [0.01, \frac{11\pi}{12}]$
29.	$r = 2 \cos^2 \frac{\varphi}{2}, \quad \varphi \in [0.01, \pi]$
30.	$r = 2, \quad \varphi \in [0.01, \frac{11\pi}{6}]$

### Этап 3

3. Задан обучающий набор  $\{x_i, y_i\}$ ,  $i = 1, \dots, N$ . Точки набора лежат на пространственной кривой. Построить автоассоциативную сеть с узким горлом, реализующую нелинейный метод главных компонент. С помощью сети выполнить аппроксимацию кривой, выделяя две старшие нелинейные главные компоненты.

3.1 Модифицировать обучающее множество из задания 2, добавив в каждой точке третью координату по формуле

$$z = \varphi$$

Выполнить преобразование множества с помощью функции *con2seq*. Не выделять из обучающего множества контрольное и тестовое подмножества.

3.2 Создать многослойную сеть прямого распространения с помощью функции *feedforwardnet*. Число нейронов скрытого слоя задать равным [10, 2, 10]. Использовать активационные функции *tansig* для трех скрытых слоев и *purelin* для выходного слоя. Задать метод Левенберга-Марквардта в качестве алгоритма обучения. Сконфигурировать сеть (*configure*) под обучающее множество.

3.3 Инициализировать (*init*) весовые коэффициенты и смещения сети с помощью функции, заданной по умолчанию.

3.4 Задать параметры обучения: число эпох обучения (*net.trainParam.epochs*) равным 1000, предельное значение критерия обучения (*net.trainParam.goal*) равным  $10^{-5}$ .

3.5 Выполнить обучение сети с помощью функции *train*. Для обучения использовать обучающую выборку. Созданная сеть чувствительна к инициализации, поэтому иногда необходимо провести обучение несколько раз. Если результаты аппроксимации неудовлетворительные, то следует увеличить число эпох обучения или число нейронов в скрытых слоях, но при этом число нейронов в узком горле сети должно остаться равным 2. Занести в отчет содержимое окон Performance и Neural Network Training.

3.6 Отобразить структуру сети и проведенное обучение в отчете, заполнив таблицу 1.

3.7 Рассчитать выход сети для обучающего множества.

3.8 Отобразить обучающее множество и выход сети с параметрами  $-r$  и  $-b$  соответственно и *Linewidth* равным 2 с помощью функции *plot3*. График занести в отчет.

## Литература

1. Ежов А. А., Шумский С. А. Нейрокомпьютинг и его применения в экономике и бизнесе. – М.: МИФИ, 1998. – с. 70–79. – (серия «Учебники экономико-аналитического института МИФИ» под ред. проф. В. В. Харитонов)
2. Хайкин С. Нейронные сети: Полный курс: Пер. с англ. Н. Н. Куссуль и А. Ю. Шелестова под ред. Н. Н. Куссуль – М.: Вильямс, 2006. – с. 509–523.
3. Фихтенгольц Г. М. Курс дифференциального и интегрального исчисления. В 3 т. – 8-е изд. – М.: ФИЗМАТЛИТ, 2003. – т. 1. – с. 572–585.