МОСКОВСКИЙ АВИАЦИОННЫЙ ИНСТИТУТ
(НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ)

Кафедра вычислительной математики и программирования

Тюменцев Ю. В., Козлов Д. С.

ПРАКТИКУМ
ПО

НЕЙРОИНФОРМАТИКЕ

для специальности 010501 «Прикладная математика и информатика»
(2006–2010 гг. приема)

Под редакцией проф. Зайцева В. Е.

Москва, 2010

СОДЕРЖАНИЕ

Лабораторная работа № 1.
Персептроны. Процедура обучения Розенблатта

Целью работы является исследование свойств персептрона Розенблатта и его применение для решения задачи распознавания образов.

Основные этапы работы:

1. Для первой обучающей выборки построить и обучить сеть, которая будет правильно относить точки к двум классам. Отобразить дискриминантную линию и проверить качество обучения.

2. Изменить обучающее множество так, чтобы классы стали линейно неразделимыми. Проверить возможности обучения по правилу Розенблатта.

3. Для второй обучающей выборки построить и обучить сеть, которая будет правильно относить точки к четырем классам. Отобразить дискриминантную линию и проверить качество обучения.

Сценарий работы:

0. Для выполнения работы требуется использовать пакет прикладных программ Neural Network Toolbox системы MATLAB. Система MATLAB (Matrix Laboratory) стала стандартом де-факто в академической и научной среде. Система развивалась от специализированной программы для матричных расчетов в начале 1970-х гг. до универсальной системы компьютерной математики. Одним из достоинств системы является наличие большого числа поставляемых пакетов расширения по основным областям современной математики. Также к достоинствам можно отнести то, что множество пакетов расширений поставляется вместе с исходными текстами программ. MATLAB содержит интерпретируемый С-подобный язык программирования. Поэтому вычислительная среда MATLAB (рис. 1) может использоваться как в виде «суперкалькулятора» для выполнения огромного числа математических и научно-технических расчетов, так и для создания пользователями своих собственных программ и пакетов расширений. Система MATLAB изначально ориентирована на матричные вычисления, что должно учитываться при разработке алгоритмов. Описанные преимущества делают систему MATLAB особенно удобной для макетирования и отработки алгоритмов.

В пакете прикладных программ Neural Network Toolbox реализованы наиболее известные типы сетей, различные методы их обучения и использования. Модели сетей, реализованных в пакете, могут быть использованы для решения задач аппроксимации функции, распознавания образов, классификации, сжатия данных и оптимизации. Для сети каждого типа реализованы функции создания, инициализации, обучения, адаптации и демонстрационные примеры. Пакет может использоваться совместно с системой блочного имитационного моделирования Simulink.

Для выполнения заданий допускается использовать любые версии пакета Neural Network Toolbox. Но в сценарии работы указываются функции пакета Neural Network Toolbox версии 7.0 (MATLAB R2010b).

1. С помощью персептрона Розенблатта решить задачу классификации точек плоскости. Точки располагаются по осям в диапазоне [-5;5]. Для этого построить и обучить сеть, которая будет правильно классифицировать точки из заданного набора примеров. В сети должны быть нейроны, имеющие ненулевое смещение.

Для первой обучающей выборки построить и обучить сеть, которая будет правильно относить точки к двум классам.

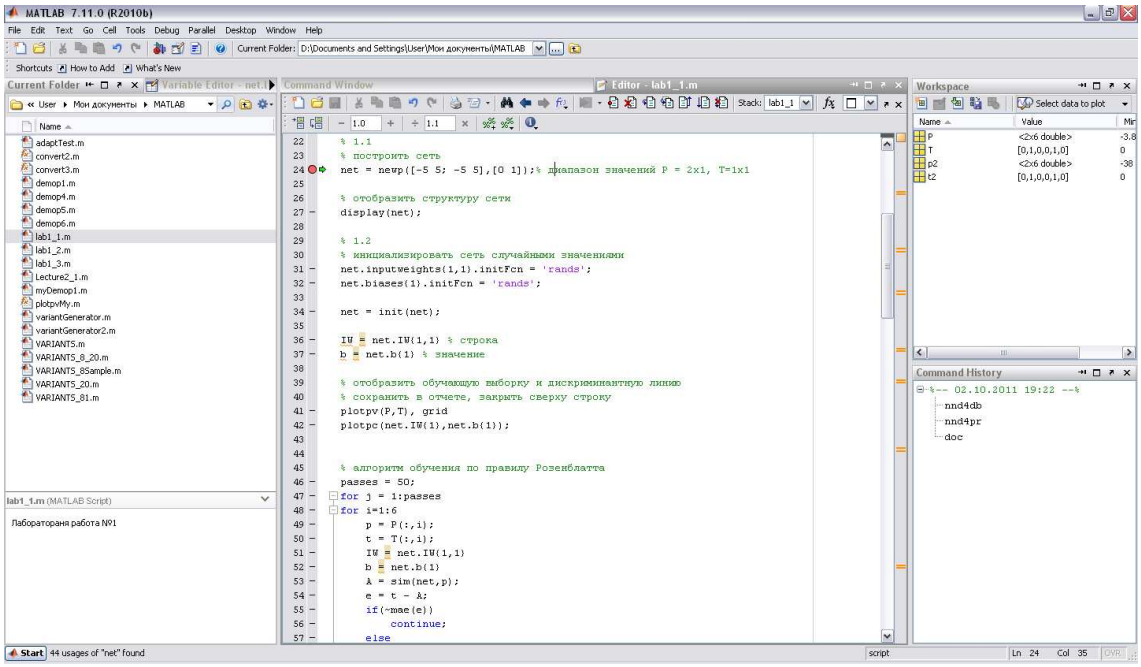

Рис. 1. Окно вычислительной среды MATLAB.

1.1 Обучающее множество занести в отчет.

1.2 Создать сеть. Сконфигурировать сеть под обучающее множество. Отобразить структуру сети с помощью функции *display* и результат занести в отчет.

1.3 Реализовать алгоритм обучения по правилу Розенблатта. Код алгоритма занести в отчет.

1.3.1 Инициализировать сеть случайными значениями. Для инициализации весов и смещений использовать функцию *rands*. Занести в отчет весовые коэффициенты и смещения.

1.3.2 Рассчитать два цикла обучения сети по правилу. Для расчета выходов сети использовать функцию *net*. В качестве показателя качества обучения использовать функцию *mae*. Занести в отчет весовые коэффициенты и смещения после расчета каждой эпохи (итерации). Также занести в отчет ошибку обучения сети по всей обучающей выборке (*mae(T - net(P))*)

1.3.3 После обучения отобразить обучающую выборку и дискриминантную линию. Для отображения использовать функции *plotpv* и *plotpc* соответственно. Также отобразить сетку с помощью функции *grid*.

1.4 Провести обучение сети с помощью встроенной функции *train* и проверить качество обучения. Занести в отчет окно Neural Network Training.

1.4.1 Инициализировать сеть случайными значениями.

1.4.2 Провести обучение сети с помощью функции *train* с числом эпох равным 50. Если необходимо, то произвести обучение несколько раз. Занести в отчет весовые коэффициенты и смещения.

1.4.3 Проверить качество обучения: случайным образом задать 3 точки и классифицировать их. Для генерации случайных чисел использовать функцию *rands*. Отобразить сетку, дополнительные точки, обучающую выборку, и дискриминантную линию. Результаты занести в отчет.

2. Изменить обучающее множество так, чтобы классы стали линейно неразделимыми. Проверить возможности обучения по правилу Розенблатта.

2.1 Изменить обучающее множество.

2.2 Инициализировать сеть случайными значениями.

2.3 Провести обучение сети с помощью функции *train* с числом эпох равным 50. Отобразить обучающую выборку и полученную дискриминантную линию. Результаты занести в отчет.

3. Для второй обучающей выборки построить и обучить сеть, которая будет правильно относить точки к четырем классам.

3.1 Обучающее множество занести в отчет.

3.2 Создать сеть.

3.3 Инициализировать сеть случайными значениями.

3.4 Провести обучение сети с помощью функции *train* с числом эпох равным 50. Если необходимо, то произвести обучение несколько раз. Занести в отчет весовые коэффициенты и смещения. Занести в отчет окно Neural Network Training.

3.5 Проверить качество обучения: случайным образом задать 5 точек и классифицировать их. Отобразить сетку, дополнительные точки, обучающую выборку, и дискриминантную линию. Результаты занести в отчет.

Варианты заданий:

Номер варианта соответствует номеру студента в списке группы.

№	Входные образы	Распределение по классам
1.	$\begin{bmatrix} 1.1 & -1.5 & 0.8 & 4.1 & 2.5 & -1.2 \\ -0.3 & 3.3 & 0.4 & -2.2 & 2.5 & 0.6 \end{bmatrix}$	$\begin{bmatrix} 1 & 0 & 1 & 1 & 0 & 1 \end{bmatrix}$
	$\begin{bmatrix} 3.6 & -1.5 & -2.8 & 1 & -3.6 & -0.8 & 2.2 & 3.4 \\ 1.3 & 4.9 & 1.5 & -1.2 & -4.8 & -3.2 & -1.3 & 2.3 \end{bmatrix}$	$\begin{bmatrix} 0 & 0 & 1 & 0 & 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$
2.	$\begin{bmatrix} 2.6 & 3.6 & 0.1 & 0.8 & -3.1 & 2.4 \\ -3.4 & 4.8 & 3.8 & -3.5 & -1 & 3.2 \end{bmatrix}$	$\begin{bmatrix} 1 & 1 & 0 & 0 & 0 & 1 \end{bmatrix}$
	$\begin{bmatrix} -1.6 & 2.9 & 1.8 & -4.5 & -4.6 & 2.2 & 3.7 & -4.3 \\ 2.3 & 0.4 & 3.9 & -2 & -3.1 & 2.2 & 0.8 & 4.2 \end{bmatrix}$	$\begin{bmatrix} 0 & 1 & 1 & 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 1 & 1 & 1 & 1 & 0 \end{bmatrix}$

№	P	T
3.	$\begin{bmatrix} -2.8 & 4 & 3.4 & 0.8 & 1.6 & 1.2 \\ -0.1 & 0.7 & 2.3 & -2.6 & -4.2 & 1.6 \end{bmatrix}$	$\begin{bmatrix} 1 & 0 & 0 & 1 & 1 & 1 \end{bmatrix}$
	$\begin{bmatrix} -1.8 & 2.1 & 2.2 & 1.7 & -0.7 & 3.1 & -2.6 & -1.3 \\ -0.5 & 3.8 & -4.9 & -0.7 & -3.9 & -1.8 & -1.6 & 0.4 \end{bmatrix}$	$\begin{bmatrix} 0 & 1 & 1 & 1 & 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 & 1 & 0 & 1 & 0 \end{bmatrix}$
4.	$\begin{bmatrix} -4 & -3.4 & 0.7 & 4.3 & 2.3 & 3.6 \\ -3.6 & 1.2 & -4.5 & 2.2 & -4.4 & 4.3 \end{bmatrix}$	$\begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 1 \end{bmatrix}$
	$\begin{bmatrix} 4.3 & -2.5 & 0.9 & 1.1 & 0.3 & -0.5 & 4.6 & 1.9 \\ -3.1 & 3.9 & 0 & 3.1 & -3 & -0.8 & 1.2 & 2.2 \end{bmatrix}$	$\begin{bmatrix} 0 & 1 & 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 & 1 & 0 & 1 \end{bmatrix}$
5.	$\begin{bmatrix} 0.3 & 4.9 & -0.9 & 2.6 & -4 & -1.5 \\ 2 & -2.2 & -0.4 & 3.1 & -3.3 & -4.5 \end{bmatrix}$	$\begin{bmatrix} 0 & 0 & 1 & 0 & 1 & 1 \end{bmatrix}$
	$\begin{bmatrix} -3.5 & -0.7 & 1.1 & 3.6 & 4 & 0.1 & 0.5 & 2.6 \\ -4.9 & 3.3 & 0.2 & -4.1 & -4 & -3.6 & -5 & 3.4 \end{bmatrix}$	$\begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 \end{bmatrix}$
6.	$\begin{bmatrix} -0.5 & 4.9 & -2.1 & -2.1 & 0 & 1.3 \\ -4 & -1.7 & -4.4 & -4.6 & 2.6 & -4.2 \end{bmatrix}$	$\begin{bmatrix} 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix}$
	$\begin{bmatrix} -0.8 & -2.1 & -3.9 & 2 & 2.8 & -1.1 & -2.8 & -3.2 \\ -2.5 & -0.8 & -0.1 & -2.6 & -4.3 & -5 & -5 & -3.6 \end{bmatrix}$	$\begin{bmatrix} 1 & 0 & 0 & 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 \end{bmatrix}$
7.	$\begin{bmatrix} -3.8 & -0.2 & 2.9 & -4.5 & -4.2 & 4.4 \\ 0.4 & 3.9 & 2.3 & -4.3 & 2.9 & 1.8 \end{bmatrix}$	$\begin{bmatrix} 1 & 1 & 0 & 0 & 1 & 0 \end{bmatrix}$
	$\begin{bmatrix} -0.6 & -4.7 & 2.1 & -1.7 & -1.8 & 0.4 & 0.5 & -2.6 \\ 4 & 0.3 & -3.3 & -3.2 & -1 & -4.6 & -2.3 & -2.6 \end{bmatrix}$	$\begin{bmatrix} 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 1 & 0 & 0 & 0 \end{bmatrix}$

№	P	T
8.	$\begin{bmatrix} -2.8 & -0.2 & 2.8 & -2.1 & 0.3 & -1 \\ 1.4 & -3.5 & -4 & -2.7 & -4.1 & -4 \end{bmatrix}$	$\begin{bmatrix} 0 & 1 & 1 & 0 & 1 & 0 \end{bmatrix}$
	$\begin{bmatrix} 1.7 & 4.7 & -0.5 & 1.8 & 1.5 & -1.3 & -3.9 & 4.7 \\ 3.3 & -4.5 & 0.8 & 2.1 & 2.2 & 0.8 & -4.5 & -2.2 \end{bmatrix}$	$\begin{bmatrix} 1 & 0 & 1 & 1 & 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 1 & 1 & 0 & 0 & 1 \end{bmatrix}$
9.	$\begin{bmatrix} -1.1 & 1.8 & 4.8 & 1.2 & -1.2 & 2.5 \\ -4.3 & -1 & -1 & -3.5 & -3.4 & 3.7 \end{bmatrix}$	$\begin{bmatrix} 0 & 1 & 1 & 1 & 0 & 1 \end{bmatrix}$
	$\begin{bmatrix} 4.6 & -1 & -0.3 & -1.1 & 0.5 & 4.9 & 0.3 & -3.9 \\ 1.7 & 4.3 & -2.7 & 2 & 2.5 & 4.6 & 4.6 & -4.5 \end{bmatrix}$	$\begin{bmatrix} 1 & 0 & 1 & 1 & 1 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$
10.	$\begin{bmatrix} 3 & -3.8 & -1.8 & -1.1 & -3.2 & -4.8 \\ 2.4 & 0.2 & 0.4 & -0.9 & -2.5 & 4.2 \end{bmatrix}$	$\begin{bmatrix} 0 & 1 & 1 & 1 & 1 & 0 \end{bmatrix}$
	$\begin{bmatrix} 2 & 2.3 & 0.4 & -1.9 & -3.2 & -0.4 & 4.1 & -5 \\ -1.3 & 4.5 & 0.4 & -4.3 & -4.1 & -5 & 1.4 & -4.7 \end{bmatrix}$	$\begin{bmatrix} 0 & 0 & 0 & 1 & 1 & 1 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 & 1 & 1 & 0 \end{bmatrix}$
11.	$\begin{bmatrix} -3.9 & 4.5 & 0.8 & 2.5 & 0 & 3.9 \\ -0.1 & -1.6 & -2.8 & -2.5 & 1.9 & 4.5 \end{bmatrix}$	$\begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 1 \end{bmatrix}$
	$\begin{bmatrix} 3.9 & -4.6 & 2.7 & -3.3 & -2.9 & 4 & -4 & -4.5 \\ -4.1 & 0.5 & -1.9 & -1.7 & 0.1 & 1.2 & -1.1 & 0 \end{bmatrix}$	$\begin{bmatrix} 0 & 1 & 0 & 0 & 1 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 & 0 & 1 & 0 & 0 \end{bmatrix}$
12.	$\begin{bmatrix} 2.7 & -3.8 & -0.4 & -1.7 & 2.9 & 0.2 \\ 4.3 & 0.6 & -4.9 & -3.4 & -1.9 & -3.4 \end{bmatrix}$	$\begin{bmatrix} 0 & 0 & 1 & 1 & 1 & 1 \end{bmatrix}$
	$\begin{bmatrix} -1.5 & 4.6 & 4.7 & 1.6 & 1.7 & 1.2 & -4.9 & 4.7 \\ -0.6 & -4.6 & -3.2 & 0.8 & -1.4 & 3.1 & -4.2 & 1.5 \end{bmatrix}$	$\begin{bmatrix} 0 & 0 & 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 1 & 1 & 0 & 0 & 0 & 1 & 1 \end{bmatrix}$

№	P	T
13.	$\begin{bmatrix} 0.7 & -2.7 & 3.2 & -4.6 & 1.4 & 1.4 \\ -4.5 & -1.5 & -4.9 & -3.4 & 2.3 & -0.5 \end{bmatrix}$	$\begin{bmatrix} 0 & 1 & 0 & 1 & 1 & 0 \end{bmatrix}$
	$\begin{bmatrix} 0.6 & 2.4 & 1.4 & -3.7 & -1.4 & 2.8 & -3.7 & 0.5 \\ -2.4 & 0 & -2 & -0.3 & 2.8 & 1.6 & -4.8 & -2 \end{bmatrix}$	$\begin{bmatrix} 1 & 1 & 1 & 0 & 0 & 1 & 1 & 1 \\ 0 & 1 & 0 & 0 & 1 & 1 & 0 & 0 \end{bmatrix}$
14.	$\begin{bmatrix} 4.7 & -3.9 & -1 & -2.4 & 2.1 & -3.9 \\ -0.7 & -2.5 & 0.9 & 1 & -2.8 & -2.1 \end{bmatrix}$	$\begin{bmatrix} 0 & 1 & 0 & 0 & 1 & 1 \end{bmatrix}$
	$\begin{bmatrix} -0.5 & 2.8 & 4.1 & 0.9 & 3.9 & -3 & 2.6 & -2.2 \\ -2.6 & 3.8 & 0.5 & -3.6 & -0.5 & 3.9 & 3.8 & 1.7 \end{bmatrix}$	$\begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 1 & 1 & 1 \\ 0 & 1 & 1 & 0 & 1 & 0 & 1 & 0 \end{bmatrix}$
15.	$\begin{bmatrix} -4.1 & -1.7 & -3.7 & -4 & -0.1 & 2.1 \\ -2.4 & 1.7 & 2.2 & 1.5 & 2.7 & 4 \end{bmatrix}$	$\begin{bmatrix} 1 & 1 & 0 & 0 & 1 & 1 \end{bmatrix}$
	$\begin{bmatrix} 2 & -2.3 & -4.1 & 1.9 & 4.5 & -0.7 & 2.6 & -3.2 \\ -4.7 & -4.6 & 3.2 & -1.9 & -4.7 & -1.2 & 2.9 & -0.2 \end{bmatrix}$	$\begin{bmatrix} 1 & 1 & 0 & 1 & 1 & 1 & 0 & 1 \\ 0 & 1 & 1 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$
16.	$\begin{bmatrix} -5 & -1.2 & -5 & -0.8 & 2.7 & 2.8 \\ 1 & 4.1 & -0.4 & -0.4 & -1.8 & -0.3 \end{bmatrix}$	$\begin{bmatrix} 0 & 0 & 0 & 1 & 1 & 1 \end{bmatrix}$
	$\begin{bmatrix} 4.1 & 2.5 & -1.2 & -4.3 & 0.3 & 4.3 & 0.6 & -4.9 \\ -2.2 & 2.5 & 0.6 & -4.5 & 2.7 & -3.8 & -0.4 & -1.7 \end{bmatrix}$	$\begin{bmatrix} 1 & 0 & 0 & 1 & 0 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 & 1 & 1 & 1 & 0 \end{bmatrix}$
17.	$\begin{bmatrix} 1.4 & 1.3 & 1.4 & 2.2 & 4.9 & -4 \\ -0.3 & 0.4 & 0.4 & 0.2 & -2.9 & -4 \end{bmatrix}$	$\begin{bmatrix} 1 & 1 & 1 & 1 & 0 & 0 \end{bmatrix}$
	$\begin{bmatrix} -2.6 & -4.1 & 4.4 & 0.7 & -2.7 & 3.2 & -4.6 & 1.4 \\ -1 & -3.7 & 4.5 & -4.5 & -1.5 & -4.9 & -3.4 & 2.3 \end{bmatrix}$	$\begin{bmatrix} 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 & 0 & 1 \end{bmatrix}$

№	P	T
18.	$\begin{bmatrix} 2.6 & 3 & -1.3 & 2.9 & -1.8 & -0.7 \\ -1.1 & 2.5 & -2.9 & 4.4 & 1.7 & 3.3 \end{bmatrix}$	$\begin{bmatrix} 1 & 1 & 1 & 1 & 0 & 0 \end{bmatrix}$
	$\begin{bmatrix} -2.8 & -2.8 & -1.9 & -0.7 & 4 & -0.7 & -2.5 & 0.9 \\ -3.3 & -0.7 & 4.2 & -3.2 & 4.7 & -3.9 & -1 & -2.4 \end{bmatrix}$	$\begin{bmatrix} 0 & 0 & 1 & 0 & 1 & 0 & 0 & 1 \\ 1 & 1 & 1 & 0 & 0 & 0 & 1 & 0 \end{bmatrix}$
19.	$\begin{bmatrix} 1.9 & 3.1 & 4.8 & 3.6 & 4.8 & -0.3 \\ 0.8 & 3.7 & -5 & 1.1 & 0.2 & 3 \end{bmatrix}$	$\begin{bmatrix} 1 & 1 & 0 & 0 & 0 & 1 \end{bmatrix}$
	$\begin{bmatrix} -4 & -0.1 & 2.1 & 3.9 & 1.9 & -4.7 & 0 & 4 \\ 1.5 & 2.7 & 4 & -1.7 & -3.1 & 2.4 & -0.3 & 1 \end{bmatrix}$	$\begin{bmatrix} 0 & 0 & 0 & 1 & 1 & 0 & 0 & 1 \\ 0 & 1 & 1 & 1 & 1 & 0 & 1 & 1 \end{bmatrix}$
20.	$\begin{bmatrix} 4.3 & 2.3 & 3.6 & 4.8 & 2.8 & -3.3 \\ 2.2 & -4.4 & 4.3 & 3.5 & 0.1 & -1.1 \end{bmatrix}$	$\begin{bmatrix} 1 & 0 & 1 & 1 & 1 & 0 \end{bmatrix}$
	$\begin{bmatrix} -4.4 & 0.2 & 1.5 & -2.1 & -4.9 & -3.4 & -1.3 & -0.2 \\ -1.1 & -0.9 & 1.2 & -0.7 & 4.8 & -4 & -3.1 & -1.7 \end{bmatrix}$	$\begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 \end{bmatrix}$
21.	$\begin{bmatrix} 2.6 & -4 & -1.5 & 0.2 & -3.3 & 4 \\ 3.1 & -3.3 & -4.5 & -1.7 & -3 & 1.7 \end{bmatrix}$	$\begin{bmatrix} 0 & 1 & 1 & 0 & 1 & 0 \end{bmatrix}$
	$\begin{bmatrix} -1.2 & -2.5 & 1.1 & 3.2 & 2.3 & 0.8 & 4 & 3.1 \\ 0.8 & -2.1 & -2.4 & 4.8 & -1.6 & -4 & -3.7 & -2.4 \end{bmatrix}$	$\begin{bmatrix} 1 & 1 & 0 & 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 1 & 0 & 0 \end{bmatrix}$
22.	$\begin{bmatrix} -2.1 & 0 & 1.3 & -4.2 & 4 & -4 \\ -4.6 & 2.6 & -4.2 & 2.7 & 0.3 & 3.2 \end{bmatrix}$	$\begin{bmatrix} 0 & 0 & 0 & 1 & 0 & 1 \end{bmatrix}$
	$\begin{bmatrix} 1.3 & -2.6 & -2.2 & 1.9 & -2.5 & 1.6 & -1.6 & 1.7 \\ 4.5 & 1.7 & 1.7 & -4.4 & -2.8 & 3.4 & 2.8 & -5 \end{bmatrix}$	$\begin{bmatrix} 1 & 1 & 1 & 0 & 0 & 1 & 1 & 0 \\ 1 & 1 & 1 & 0 & 1 & 0 & 1 & 0 \end{bmatrix}$

№	P	T
23.	$\begin{bmatrix} 4.7 & -4.2 & -1.4 & 0.9 & -1.4 & -4.2 \\ 3.6 & -1.4 & 1.8 & 2.8 & -3 & 2.7 \end{bmatrix}$	$\begin{bmatrix} 0 & 1 & 1 & 1 & 1 & 1 \end{bmatrix}$
	$\begin{bmatrix} -2.7 & 1 & -0.5 & 2.7 & 1.6 & 3.4 & -2.5 & 0.8 \\ -3.9 & -0.5 & 1.6 & -1.5 & -0.9 & 3.3 & 1.1 & 0.4 \end{bmatrix}$	$\begin{bmatrix} 1 & 0 & 1 & 0 & 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix}$
24.	$\begin{bmatrix} -3.2 & 2.9 & 2.6 & -2.3 & 1.7 & -0.5 \\ -2.4 & -0.2 & -1.1 & -4.7 & -0.8 & 1 \end{bmatrix}$	$\begin{bmatrix} 1 & 0 & 0 & 1 & 0 & 1 \end{bmatrix}$
	$\begin{bmatrix} -1.6 & 0.8 & -4.6 & -2.6 & 1.8 & 2.3 & 1.8 & -0.6 \\ -3.6 & -2.4 & 2.5 & -0.6 & -1.5 & -1.1 & 2 & -4.9 \end{bmatrix}$	$\begin{bmatrix} 1 & 1 & 0 & 0 & 1 & 1 & 0 & 1 \\ 1 & 0 & 1 & 1 & 0 & 0 & 0 & 1 \end{bmatrix}$
25.	$\begin{bmatrix} -3.2 & -4.3 & 2 & -1.9 & 1.2 & -3.3 \\ -2.9 & 4.1 & 0.5 & -3.4 & 4.8 & -2.5 \end{bmatrix}$	$\begin{bmatrix} 0 & 1 & 1 & 0 & 1 & 0 \end{bmatrix}$
	$\begin{bmatrix} 0.7 & 2.3 & -2.6 & -4.2 & 1.6 & 3.9 & 2.6 & 4.2 \\ 3.4 & 0.8 & 1.6 & 1.2 & 2.2 & 4.8 & 0.8 & 0.8 \end{bmatrix}$	$\begin{bmatrix} 1 & 0 & 1 & 1 & 0 & 1 & 0 & 0 \\ 1 & 1 & 1 & 1 & 1 & 0 & 1 & 0 \end{bmatrix}$
26.	$\begin{bmatrix} -3.8 & -2.1 & 1.5 & 4.3 & -2.6 & 2.5 \\ -4.8 & -1.9 & 4.5 & -0.5 & 2.6 & 2.4 \end{bmatrix}$	$\begin{bmatrix} 1 & 1 & 1 & 0 & 1 & 0 \end{bmatrix}$
	$\begin{bmatrix} -4 & -3.4 & 0.7 & 4.3 & 2.3 & 3.6 & 4.8 & 2.8 \\ -3.6 & 1.2 & -4.5 & 2.2 & -4.4 & 4.3 & 3.5 & 0.1 \end{bmatrix}$	$\begin{bmatrix} 0 & 0 & 0 & 1 & 0 & 1 & 1 & 0 \\ 1 & 1 & 0 & 0 & 0 & 1 & 0 & 0 \end{bmatrix}$
27.	$\begin{bmatrix} -2.5 & 4.2 & 4.3 & 4.2 & 0.7 & -2.5 \\ -4.8 & 1.5 & -3.4 & 2.9 & -0.6 & 2.5 \end{bmatrix}$	$\begin{bmatrix} 0 & 1 & 0 & 1 & 0 & 0 \end{bmatrix}$
	$\begin{bmatrix} -3 & 1.7 & 4.1 & 2.4 & 0.6 & 0.9 & -3.7 & 3.9 \\ 4 & -0.4 & -4 & 2.3 & -3.2 & -2.1 & -2.9 & -4.3 \end{bmatrix}$	$\begin{bmatrix} 1 & 1 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 1 & 1 & 0 & 1 \end{bmatrix}$

№	P	T
28.	$\begin{bmatrix} -4.8 & -3.2 & -1.3 & 2.3 & -3.3 & -2.4 \\ -0.8 & 2.2 & 3.4 & 0.7 & 4.5 & 4.2 \end{bmatrix}$	$\begin{bmatrix} 1 & 1 & 0 & 0 & 0 & 0 \end{bmatrix}$
	$\begin{bmatrix} 1.9 & -1.1 & 2.8 & 1 & -4 & 0.4 & 3.9 & 2.3 \\ 0.5 & -4.4 & -1.7 & 2.4 & -3.8 & -0.2 & 2.9 & -4.5 \end{bmatrix}$	$\begin{bmatrix} 0 & 1 & 0 & 1 & 1 & 1 & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 & 1 & 1 & 0 \end{bmatrix}$
29.	$\begin{bmatrix} 0.3 & 3 & -4.4 & -4.9 & 2.8 & 3.8 \\ -4.2 & 4.8 & 4.3 & 1.8 & 0.3 & 3.9 \end{bmatrix}$	$\begin{bmatrix} 0 & 1 & 1 & 0 & 1 & 1 \end{bmatrix}$
	$\begin{bmatrix} -2.1 & 0.3 & -1 & -3.9 & -2.1 & 4.6 & 1.9 & -0.7 \\ -2.7 & -4.1 & -4 & 2.8 & 1 & -0.7 & 2.5 & 1.5 \end{bmatrix}$	$\begin{bmatrix} 0 & 1 & 1 & 0 & 0 & 1 & 1 & 0 \\ 1 & 1 & 1 & 0 & 0 & 1 & 0 & 0 \end{bmatrix}$
30.	$\begin{bmatrix} 0.1 & 2.1 & -1.5 & -1.6 & -0.5 & -2.9 \\ -4.2 & 4.9 & 4.7 & 3.8 & -0.9 & -3.8 \end{bmatrix}$	$\begin{bmatrix} 0 & 0 & 1 & 1 & 0 & 0 \end{bmatrix}$
	$\begin{bmatrix} -3.4 & 3.7 & 1.8 & 0.3 & 0.9 & -2.1 & -0.8 & -0.5 \\ 2.5 & -1.5 & -2.1 & 3.3 & -1.7 & -0.5 & -1.5 & 2.4 \end{bmatrix}$	$\begin{bmatrix} 1 & 1 & 0 & 1 & 0 & 0 & 0 & 1 \\ 0 & 1 & 1 & 0 & 1 & 0 & 0 & 0 \end{bmatrix}$

Литература

1. *Beale M., Hagan M., Demuth H.* Neural Network Toolbox User's guide R2011b. The MathWorks, 2011. –pp. 9-3–9-17.

2. *Медведев В. С., Потемкин В. Г.* Нейронные сети. MATLAB 6/Под общ. ред. к. т. н. В. Г. Потемкина – М.: ДИАЛОГ-МИФИ, 2006. – с. 102–114.

3. *Hagan M., Demuth H.* Neural Network Design. 1996. – Chapter 4. – 35 pp.

Лабораторная работа № 2.
Линейная нейронная сеть. Правило обучения Уидроу-Хоффа

Целью работы является исследование свойств линейной нейронной сети и алгоритмов ее обучения, применение сети в задачах аппроксимации и фильтрации.

Основные этапы работы:

1. Использовать линейную нейронную сеть с задержками для аппроксимации функции. В качестве метода обучения использовать адаптацию.

2. Использовать линейную нейронную сеть с задержками для аппроксимации функции и выполнения многошагового прогноза.

3. Использовать линейную нейронную сеть в качестве адаптивного фильтра для подавления помех. Для настройки весовых коэффициентов использовать метод наименьших квадратов.

Сценарий работы:

1. Задана временная последовательность $x(n)$. Построить и обучить линейную сеть с задержками, которая будет выполнять одношаговый прогноз для первой функции из варианта задания:

$$\hat{x}(n+1) = \sum_{i=1}^{D} w_i x(n-i+1) + b$$

где D задает глубину погружения временного ряда (*delays*), $\{w_i, b\}$ — весовые коэффициенты.

1.1 Построить обучающее множество: в качестве входного множества использовать значения первого входного сигнала на заданном интервале; преобразовать входное множество к последовательности входных образцов с помощью функции *con2seq*; эталонные выходы сети формируются из входной последовательности, чтобы сеть выполняла одношаговый прогноз.

1.2 Создать сеть с помощью функции *newlin*. Задать задержки от 1 до $D = 5$. Задать скорость обучения равной 0.01.

1.3 Инициализировать сеть случайными значениями.

1.4 Выполнить адаптацию с числом циклов равным 50. Занести в отчет величину ошибки обучения с помощью $sqrt(mse)$. Поскольку сеть имеет задержки, то в функцию адаптации необходимо отдельно передать первые 5 элементов входной последовательности для инициализации задержек (входной параметр *Pi*). В противном случае задержки будут инициализированы нулями, что приведет к увеличению ошибки обучения при выполнении адаптации. В дальнейшем использовать входную и выходную последовательности, начиная с 6 элемента.

1.5 Отобразить на графике эталонные значения и предсказанные сетью. График занести в отчет.

2. Для временной последовательности из задания 1 обучить линейную сеть с задержками (линейный адаптивный фильтр) и выполнить многошаговый прогноз.

2.1 Построить обучающее множество: в качестве входного множества использовать значения первого входного сигнала на заданном интервале; преобразовать входное множество к последовательности входных образцов с помощью функции *con2seq*; эталонные выходы сети формируются из входной последовательности, чтобы сеть выполняла одношаговый прогноз.

2.2 Создать сеть с помощью функции *newlin*. Задать задержки от 1 до $D = 3$. Задать скорость обучения с помощью функции *maxlinlr(cell2mat(P),'bias')*.

2.3 Инициализировать сеть случайными значениями.

2.4 Задать параметры обучения: число эпох обучения (*net.trainParam.epochs*) равным равными 600, предельное значение критерия обучения (*net.trainParam.goal*) равным 10^{-6}. Также

необходимо проинициализировать задержки *Pi*. Выполнить обучение сети с помощью функции *train*.

2.5 Занести в отчет весовые коэффициенты и смещение. Занести в отчет окно Performance и Neural Network Training. Отразить структуру сети и проведенное обучение в отчете, заполнив таблицу 1.

2.6 Рассчитать выход сети (*sim*) для обучающего множества. Сравнить выход сети с соответствующим эталонным множеством: рассчитать показатели качества обучения и заполнить таблицу 2. Отобразить на графике эталонные значения и предсказанные сетью, а также ошибку обучения. Графики занести в отчет.

2.7 Сформировать набор данных для выполнения прогноза: продлить временную последовательность с заданным шагом на 10 отсчетов. Использовать полученный набор данных для выполнения прогноза: рассчитать выход сети (*sim*) для полученного набора. Сравнить выход сети с соответствующим куском исходной временной последовательности: рассчитать показатели качества обучения и заполнить таблицу 2. Отобразить на графике эталонные значения и предсказанные сетью, а также ошибку обучения. Графики занести в отчет.

3. Построить и обучить линейную сеть, которая является адаптивным линейным фильтром. Задачей фильтра является моделирование источника шума, чтобы в последующем удалить помехи из полезного сигнала. Фильтр должен аппроксимировать отображение:

$$\hat{y}(n+1) = \sum_{i=1}^{D} w_i x(n-i+1) + b$$

Вместо задержек использовать погружение временного ряда.

3.1 Построить обучающее множество: в качестве входного множества использовать значения второго входного сигнала на заданном интервале; эталонными выходами сети являются значения второй эталонной функции на заданном интервале. Эталонный выходной сигнал соответствует входному сигналу, измененному по амплитуде и смещенному по фазе, поэтому диапазон значений и шаг для сигналов совпадают.

3.2 Вместо задержек необходимо расширить входное множество по формуле

$$P = zeros(D, Q)$$
$$P(i, i:Q) = x(1:Q-i+1),\ i = 1, \ldots, D$$

где Q — количество элементов. Задать глубину погружения временного ряда D равной 4.

3.3 Создать сеть с помощью функции *newlind*. Занести в отчет весовые коэффициенты и смещение.

3.4 Рассчитать выход сети (*sim*) для обучающего множества. Сравнить выход сети с эталонным множеством: рассчитать показатели качества обучения и заполнить таблицу 2. Отобразить на графике эталонные значения и предсказанные сетью, а также ошибку обучения. Графики занести в отчет.

Варианты заданий:

Номер варианта соответствует номеру студента в списке группы.

№	Входной сигнал	Выходной сигнал
1.	$x = \sin(t^2), \quad t \in [0, 5],\ h = 0.025$ $x = \sin(\frac{2\pi t}{3}), \quad t \in [0, 5],\ h = 0.025$	$y = 0.2 \sin(\frac{2\pi t}{3} + \frac{\pi}{2})$

№	P	T
2.	$x = \sin\left(\frac{1}{2}t^2 - 5t\right), \quad t \in [0, 2.2],\ h = 0.01$ $x = \sin\left(-3t^2 + 5t + 10\right), \quad t \in [0, 2.5],\ h = 0.01$	$y = \frac{1}{3}\sin\left(-3t^2 + 5t - 3\right)$
3.	$x = \sin\left(-2t^2 + 7t\right), \quad t \in [0, 5],\ h = 0.025$ $x = \sin\left(2.5t^2 - 5t\right), \quad t \in [0, 2.2],\ h = 0.01$	$y = \frac{1}{3}\sin\left(2.5t^2 - 5t + 4\pi\right)$
4.	$x = \sin\left(-3t^2 + 10t - 5\right), \quad t \in [0.5, 4],\ h = 0.01$ $x = \sin\left(2t^2 - 6t + 3\right), \quad t \in [0, 5],\ h = 0.02$	$y = \frac{1}{2}\sin\left(2t^2 - 6t - \pi\right)$
5.	$x = \sin\left(t^2 - 7t\right), \quad t \in [0, 5],\ h = 0.025$ $x = \sin\left(t^2 - 6t + 3\right), \quad t \in [0, 5],\ h = 0.025$	$y = \frac{1}{4}\sin\left(t^2 - 6t - 2\pi\right)$
6.	$x = \sin\left(t^2 - 2t + 5\right), \quad t \in [0, 5],\ h = 0.025$ $x = \sin\left(t^2 - 5t + 6\right), \quad t \in [0, 6],\ h = 0.02$	$y = \frac{1}{5}\sin\left(t^2 - 5t + 3\right)$
7.	$x = \sin\left(t^2 - 6t + 3\right), \quad t \in [0, 6],\ h = 0.025$ $x = \sin\left(\sin(t)t^2\right), \quad t \in [0, 3.5],\ h = 0.01$	$y = \frac{1}{4}\sin\left(\sin(t)t^2 - \pi\right)$
8.	$x = \sin\left(t^2 - 10t + 3\right), \quad t \in [1, 6],\ h = 0.025$ $x = \sin(-2t^2 + 7t), \quad t \in [0, 3.5],\ h = 0.01$	$y = \frac{1}{8}\sin(-2t^2 + 7t - \pi)$
9.	$x = \sin\left(t^2 - 2t + 3\right), \quad t \in [0, 6],\ h = 0.025$ $x = \sin\left(t^2 - 2t + 3\right), \quad t \in [0, 6],\ h = 0.025$	$y = \frac{1}{4}\sin\left(t^2 - 2t\right)$
10.	$x = \sin\left(-2t^2 + 7t\right) - \frac{1}{2}\sin(t), \quad t \in [0, 4.5],\ h = 0.025$ $x = \sin\left(t^2 - 6t + 3\right), \quad t \in [0, 6],\ h = 0.025$	$y = \frac{1}{3}\sin\left(t^2 - 6t - \frac{\pi}{6}\right)$
11.	$x = \sin\left(t^2 - 15t + 3\right) - \sin\left(t\right), \quad t \in [0, 3.5],\ h = 0.01$ $x = \cos\left(2.5t^2 - 5t\right), \quad t \in [0, 2.2],\ h = 0.01$	$y = \frac{1}{4}\cos\left(2.5t^2 - 5t + \pi\right)$
12.	$x = \sin\left(t^2 - 15t + 3\right) - \sin^2\left(t\right), \quad t \in [0.5, 3],\ h = 0.01$ $x = \cos\left(t^2\right), \quad t \in [0, 4],\ h = 0.02$	$y = \frac{1}{2}\cos\left(t^2 + 2\pi\right)$
13.	$x = \sin\left(\sin(t)t^2 - t\right), \quad t \in [1, 4.5],\ h = 0.01$ $x = \sin\left(-5t^2 + 10t - 5\right), \quad t \in [0, 2.5],\ h = 0.01$	$y = \frac{1}{7}\sin\left(-5t^2 + 10t - \pi\right)$
14.	$x = \sin\left(\sin(t)t^2\right), \quad t \in [0, 3.5],\ h = 0.01$ $x = \cos\left(-2t^2 + 7t\right), \quad t \in [0, 3.5],\ h = 0.01$	$y = \frac{1}{8}\cos\left(-2t^2 + 7t + 2\pi\right)$

№	P	T
15.	$x = \sin\big(\sin(t)t^3 - 10\big), \quad t \in [1,3],\ h = 0.01$ $x = \cos\big(t^2 - 10t + 3\big), \quad t \in [1,6],\ h = 0.025$	$y = \frac{1}{5}\cos\big(t^2 - 10t + 6\big)$
16.	$x = \sin\big(-\sin(t)t^2 + t\big), \quad t \in [0.5,4],\ h = 0.01$ $x = \cos\big(-5t^2 + 10t - 5\big), \quad t \in [0,2.5],\ h = 0.01$	$y = \frac{1}{8}\cos\big(-5t^2 + 10t\big)$
17.	$x = \sin\big(\sin(t)t^2 + 3t - 10\big), \quad t \in [2.5,5],\ h = 0.01$ $x = \cos\big(\cos(t)t^2 + 5t\big), \quad t \in [0,3.5],\ h = 0.01$	$y = \frac{1}{5}\cos\big(\cos(t)t^2 + 5t + 4\big)$
18.	$x = \sin\big(\sin(t)t^2 - 2t + 7\big), \quad t \in [0,4],\ h = 0.02$ $x = \cos\big(\cos(t)t^2 - t\big), \quad t \in [1,4.5],\ h = 0.01$	$y = \frac{1}{5}\cos\big(\cos(t)t^2 - t + \pi\big)$
19.	$x = \sin\big(-2\sin(t)t^2 + 7\big), \quad t \in [0,3.5],\ h = 0.01$ $x = \cos\big(t^2 - 2t + 3\big), \quad t \in [0,6],\ h = 0.025$	$y = \frac{1}{3}\cos\big(t^2 - 2t - \pi\big)$
20.	$x = \sin\big(-2\sin(t)t^2 + 7t\big), \quad t \in [0.5,3.2],\ h = 0.01$ $x = \cos\big(-\cos(t)t^2 + t\big), \quad t \in [0.5,4],\ h = 0.01$	$y = \frac{1}{4}\cos\big(-\cos(t)t^2 + t + 2\pi\big)$
21.	$x = \cos\big(-3t^2 + 10t - 5\big) - \cos(t), \quad t \in [0.5,4],\ h = 0.01$ $x = \cos\big(-3t^2 + 5t + 10\big), \quad t \in [0,2.5],\ h = 0.01$	$y = \frac{1}{6}\cos\big(-3t^2 + 5t + \frac{3\pi}{2}\big)$
22.	$x = \cos\big(\cos(t)t^2 - 2t + 7\big), \quad t \in [0,4],\ h = 0.02$ $x = \sin\big(t^2 - 10t + 3\big), \quad t \in [0,6],\ h = 0.025$	$y = \frac{1}{6}\sin\big(t^2 - 10t + \frac{\pi}{4}\big)$
23.	$x = \cos\big(t^2 - 2t + 3\big), \quad t \in [0,6],\ h = 0.025$ $x = \sin\big(t^2 - 7t\big), \quad t \in [0,5],\ h = 0.025$	$y = \frac{1}{2}\sin\big(t^2 - 7t + \frac{\pi}{4}\big)$
24.	$x = \cos\big(\frac{1}{2}t^2 - 5t\big), \quad t \in [0,2],\ h = 0.01$ $x = \sin\big(t^2 - 2t + 5\big), \quad t \in [0,5],\ h = 0.025$	$y = \frac{1}{7}\sin\big(t^2 - 2t + \pi\big)$
25.	$x = \cos\big(-\cos(t)t^2 + t\big), \quad t \in [0.5,4],\ h = 0.01$ $x = \sin\big(\frac{1}{2}t^2 - 5t\big), \quad t \in [0,2],\ h = 0.01$	$y = \frac{1}{10}\sin\big(\frac{1}{2}t^2 - 5t - \frac{3\pi}{2}\big)$
26.	$x = \cos\big(-2t^2 + 7t\big) - \frac{1}{2}\cos(t), \quad t \in [0,4.5],\ h = 0.025$ $x = \sin\big(t^2\big), \quad t \in [0,4],\ h = 0.02$	$y = \frac{1}{3}\sin\big(t^2 + \frac{\pi}{2}\big)$
27.	$x = \cos\big(\cos(t)t^2 + 3t - 10\big), \quad t \in [2.5,5],\ h = 0.01$ $x = \sin\big(-\sin(t)t^2 + t\big), \quad t \in [0.5,4],\ h = 0.01$	$y = \frac{1}{2}\sin\big(-\sin(t)t^2 + t - 2\pi\big)$

№	P	T
28.	$x = \cos\bigl(-2t^2 + 7t\bigr), \quad t \in [0, 5],\ h = 0.025$ $x = \sin\bigl(\sin(t)t^2 + 3t - 10\bigr), \quad t \in [2.5, 5],\ h = 0.01$	$y = \frac{1}{4}\sin\bigl(\sin(t)t^2 + 3t - 3\bigr)$
29.	$x = \cos\bigl(\cos(t)t^2\bigr), \quad t \in [0, 3.5],\ h = 0.01$ $x = \sin\bigl(\sin(t)t^2 - t\bigr), \quad t \in [1, 4.5],\ h = 0.01$	$y = \frac{1}{7}\sin\bigl(\sin(t)t^2 - t + \frac{3\pi}{2}\bigr)$
30.	$x = \cos\bigl(t^2 - 15t + 3\bigr) - \cos(t), \quad t \in [0.5, 3],\ h = 0.01$ $x = \sin\bigl(\sin(t)t^2 + 5t\bigr), \quad t \in [0, 3.5],\ h = 0.01$	$y = \frac{1}{4}\sin\bigl(\sin(t)t^2 + 5t - 2\pi\bigr)$

Литература

1. *Beale M., Hagan M., Demuth H.* Neural Network Toolbox User's guide R2011b. The MathWorks, 2011. –pp. 7-2–7-18, 9-18–9-33.

2. *Медведев В. С., Потемкин В. Г.* Нейронные сети. MATLAB 6/Под общ. ред. к. т. н. В. Г. Потемкина – М.: ДИАЛОГ-МИФИ, 2006. – с. 115–130, 188–198.

3. *Hagan M., Demuth H.* Neural Network Design. 1996. – Chapter 10. – 44 pp.

Лабораторная работа № 3.
Многослойные сети. Алгоритм обратного распространения ошибки

Целью работы является исследование свойств многослойной нейронной сети прямого распространения и алгоритмов ее обучения, применение сети в задачах классификации и аппроксимации функции.

Основные этапы работы:

1. Использовать многослойную нейронную сеть для классификации точек в случае, когда классы не являются линейно разделимыми.

2. Использовать многослойную нейронную сеть для аппроксимации функции. Произвести обучение с помощью одного из методов первого порядка.

3. Использовать многослойную нейронную сеть для аппроксимации функции. Произвести обучение с помощью одного из методов второго порядка.

Для обучения многослойных нейронных сетей прямого распространения используются методы поиска экстремума функций многих переменных.

Методы первого порядка:

- Метод градиентного спуска (*traingd*). Для работы алгоритма необходимо задать скорость обучения: *net.trainParam.lr = 0.05*.
- Метод градиентного спуска с моментом (*traingdm*). Для работы алгоритма необходимо задать скорость обучения и величину момента: *net.trainParam.lr = 0.05*, *net.trainParam.mc = 0.9*.
- Метод градиентного спуска с адаптивным шагом (*traingda*). Для работы алгоритма необходимо задать скорость обучения и коэффициент увеличения скорости настройки: *net.trainParam.lr = 0.05*, *net.trainParam.lr_inc = 1.05*.
- Метод градиентного спуска с адаптивным шагом и моментом (*traingdx*). Для работы алгоритма необходимо задать скорость обучения, коэффициент увеличения скорости настройки и величину момента: *net.trainParam.lr = 0.05*, *net.trainParam.lr_inc = 1.05*, *net.trainParam.mc = 0.9*.
- Метод гибкого распространения (*trainrp*). Другое название RProp (Resilient Backpropagation).
- Методы сопряженных градиентов: метод Флетчера-Ривса (*traincgf*), метод Полака-Рибейры (*traincgp*), метод Пауэлла-Биеле (*traincgb*), метод Моллера (*trainscg*). Для группы методов сопряженных градиентов рекомендуется задать число нейронов в скрытом слое равным 15.

Методы второго порядка:

- Квазиньютоновский метод, предложенный Бройденом, Флетчером, Гольдфарбом и Шанно (*trainbfg*).
- Метод Левенберга-Марквардта (*trainlm*).
- Одношаговый метод секущих (*trainoss*).

Сценарий работы:

Этап 1

1. Заданы 3 линейно неразделимых класса. Точки, принадлежащие одному классу, лежат на алгебраической линии. Построить и обучить многослойную сеть прямого распространения, которая будет классифицировать точки заданной области.

Обучающий набор $\{x_i, y_i\},\ i = 1, .., N$, число классов $K = 3$. Сеть реализует отображение вида:

$$f(x_i, y_i) = \{(z_k)_{k=1}^{K} = (0, .., 1, ..0) |\ z_{k=K^*} = 1 \text{ при } (x_i, y_i) \in K^*\}$$

1.1 В соответствии с вариантом задания для каждой линии сгенерировать множество точек. Далее для первого класса выбрать из исходного множества случайным образом 60 точек. Для второго и третьего классов 100 и 120 точек соответственно. Для выбора точек рекомендуется использовать функцию $randperm$, с помощью которой получить псевдослучайную последовательность индексов вектора.

1.2 Множество точек, принадлежащее каждому классу, разделить на обучающее, контрольное, и тестовое подмножества с помощью функции $dividerand$ в отношении 70%-20%-10%.

1.3 Отобразить с помощью функции $plot$ исходные множества точек для каждого из классов. Задать параметр $LineWidth$ равным 2, подписать линии, задать сетку. С помощью $axis$ задать границы для входного множества. Параметры отображения для классов:

Класс 1 Исходное множество: $-r$. Обучающее подмножество: or, $MarkerEdgeColor = k$, $MarkerFaceColor = r$, $MarkerSize = 7$. Контрольное подмножество: rV, $MarkerEdgeColor = k$, $MarkerFaceColor = c$, $MarkerSize = 7$.
Тестовое подмножество: rs, $MarkerEdgeColor = k$, $MarkerFaceColor = c$, $MarkerSize = 7$.

Класс 2 Исходное множество: $-g$. Обучающее подмножество: og, $MarkerEdgeColor = k$, $MarkerFaceColor = g$, $MarkerSize = 7$. Контрольное подмножество: gV, $MarkerEdgeColor = k$, $MarkerFaceColor = c$, $MarkerSize = 7$.
Тестовое подмножество: gs, $MarkerEdgeColor = k$, $MarkerFaceColor = c$, $MarkerSize = 7$.

Класс 3 Исходное множество: $-b$. Обучающее подмножество: ob, $MarkerEdgeColor = k$, $MarkerFaceColor = b$, $MarkerSize = 7$. Контрольное подмножество: bV, $MarkerEdgeColor = k$, $MarkerFaceColor = c$, $MarkerSize = 7$.
Тестовое подмножество: bs, $MarkerEdgeColor = k$, $MarkerFaceColor = c$, $MarkerSize = 7$.

1.4 Соответствующие подмножества точек каждого класса объединить в обучающее, контрольное, и тестовое подмножества обучающей выборки. Обучающая выборка состоит из последовательного объединения полученных обучающего, контрольного, и тестового подмножеств.

1.5 Создать сеть с помощью функции $feedforwardnet$. Сконфигурировать сеть ($configure$), указав диапазоны изменения для входного множества и эталонных выходов сети. Точки входного и выходного множеств лежат на отрезках $[-1.2, 1.2]$ и $[0, 1]$ по каждой из координат соответственно.

Число нейронов скрытого слоя задать равным 20. Использовать активационные функцию $tansig$ для скрытого и выходного слоев. Задать RProp в качестве алгоритма обучения.

1.6 Для разделения обучающего множества на подмножества использовать $net.divideFcn =' divideind'$. Также задать параметры:

$$net.divideParam.trainInd = 1 : trnInd;$$
$$net.divideParam.valInd = trnInd + 1 : tstInd;$$
$$net.divideParam.testInd = tstInd + 1 : proInd;$$

где $trnInd$, $tstInd$, $proInd$ задают количество примеров в обучающем, контрольном, и тестовом подмножествах.

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

1.8 Задать параметры обучения: число эпох обучения ($net.trainParam.epochs$) и число эпох, в течение которых может расти ошибка на контрольном подмножестве ($net.trainParam.max\_fail$), равными 1500, предельное значение критерия обучения ($net.trainParam.goal$) равным 10^{-5}.

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

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

1.11 Рассчитать выход сети (sim) для обучающего подмножества. Преобразовать значения по правилу

$$o_{ij} = \begin{cases} 1, & a_{ij} \geqslant 0.5; \\ 0, & a_{ij} < 0.5; \end{cases}$$

Занести в отчет количество правильно классифицированных точек.

1.12 Провести аналогичные расчеты для контрольного и тестового подмножеств.

1.13 Произвести классификацию точек области $[-1.2, 1.2] \times [-1.2, 1.2]$. Для этого задать сетку для указанной области с шагом $h = 0.025$. Рассчитать выход сети для всех узлов сетки.

1.14 Выход сети для каждой точки задает ее принадлежность к трем классам. Закодировать принадлежности к классам различными цветами и занести полученное изображение в отчет.

Для этого использовать функции $image$ и $colormap$. $image$ отображает матрицу, каждый элемент которой содержит ссылку на таблицу цветов, которая задается с помощью $colormap$. Элементы в матрице цветов должны находиться в диапазоне $[0, 1]$. Каждая компонента выходного вектора задает интенсивность одного из цветов в модели RGB. Например, $(1, 0, 0)$ — красный цвет, $(0, 1, 0)$ — зеленый, $(0, 0, 1)$ — голубой, $(1, 1, 0)$ — желтый цвет.

Сначала нужно сформировать таблицу цветов: округлить компоненты выходных векторов до десятых ($floor$ или $round$) и удалить повторяющиеся вектора с помощью функции $unique('rows')$. Затем каждый из выходных векторов заменить на номер строки из таблицы цветов. Для перехода от пакетного (batch) представления к матрице ссылок использовать функцию $reshape$ и операцию транспонирования.

Варианты заданий:

Номер варианта соответствует номеру в списке группы. Для генерации точек использовать параметрическое уравнение линии в канонической системе координат.

$$t = 0 : 0.025 : 2\pi$$
$$x = f(t)$$
$$y = g(t)$$

Константы a и b задают большую и малую полуоси эллипса, p — параметр параболы. Параметры преобразования прямоугольной системы координат на плоскости: угол поворота (α) и координаты параллельного переноса (x_0, y_0).

№	**Алгебраические линии**
1.	Эллипс: $a = 0.3$, $b = 0.3$, $\alpha = 0$, $x_0 = 0$, $y_0 = 0$ Эллипс: $a = 0.7$, $b = 0.7$, $\alpha = 0$, $x_0 = 0$, $y_0 = 0$ Эллипс: $a = 1$, $b = 1$, $\alpha = 0$, $x_0 = 0$, $y_0 = 0$

№	Алгебраические линии
2.	Эллипс: $a = 0.4$, $b = 0.15$, $\alpha = \pi/6$, $x_0 = 0$, $y_0 = 0$ Эллипс: $a = 0.7$, $b = 0.5$, $\alpha = 0$, $x_0 = 0$, $y_0 = 0$ Эллипс: $a = 1$, $b = 1$, $\alpha = 0$, $x_0 = 0$, $y_0 = 0$
3.	Эллипс: $a = 0.4$, $b = 0.15$, $\alpha = \pi/6$, $x_0 = 0$, $y_0 = 0$ Эллипс: $a = 0.7$, $b = 0.5$, $\alpha = \pi/3$, $x_0 = 0$, $y_0 = 0$ Эллипс: $a = 1$, $b = 1$, $\alpha = 0$, $x_0 = 0$, $y_0 = 0$
4.	Эллипс: $a = 0.4$, $b = 0.15$, $\alpha = \pi/6$, $x_0 = 0$, $y_0 = 0$ Эллипс: $a = 0.7$, $b = 0.5$, $\alpha = -\pi/3$, $x_0 = 0$, $y_0 = 0$ Эллипс: $a = 1$, $b = 1$, $\alpha = 0$, $x_0 = 0$, $y_0 = 0$
5.	Эллипс: $a = 0.4$, $b = 0.15$, $\alpha = \pi/6$, $x_0 = -0.1$, $y_0 = 0.15$ Эллипс: $a = 0.7$, $b = 0.5$, $\alpha = -\pi/3$, $x_0 = 0$, $y_0 = 0$ Эллипс: $a = 1$, $b = 1$, $\alpha = 0$, $x_0 = 0$, $y_0 = 0$
6.	Эллипс: $a = 0.4$, $b = 0.15$, $\alpha = \pi/6$, $x_0 = 0.1$, $y_0 = -0.15$ Эллипс: $a = 0.7$, $b = 0.5$, $\alpha = -\pi/3$, $x_0 = 0$, $y_0 = 0$ Эллипс: $a = 1$, $b = 1$, $\alpha = 0$, $x_0 = 0$, $y_0 = 0$
7.	Эллипс: $a = 0.2$, $b = 0.2$, $\alpha = 0$, $x_0 = 0.25$, $y_0 = -0.25$ Эллипс: $a = 0.7$, $b = 0.5$, $\alpha = -\pi/3$, $x_0 = 0$, $y_0 = 0$ Эллипс: $a = 1$, $b = 1$, $\alpha = 0$, $x_0 = 0$, $y_0 = 0$
8.	Эллипс: $a = 0.2$, $b = 0.2$, $\alpha = 0$, $x_0 = -0.25$, $y_0 = 0.25$ Эллипс: $a = 0.7$, $b = 0.5$, $\alpha = -\pi/3$, $x_0 = 0$, $y_0 = 0$ Эллипс: $a = 1$, $b = 1$, $\alpha = 0$, $x_0 = 0$, $y_0 = 0$
9.	Эллипс: $a = 0.2$, $b = 0.2$, $\alpha = 0$, $x_0 = -0.2$, $y_0 = 0$ Эллипс: $a = 0.7$, $b = 0.5$, $\alpha = -\pi/3$, $x_0 = 0$, $y_0 = 0$ Эллипс: $a = 1$, $b = 1$, $\alpha = 0$, $x_0 = 0$, $y_0 = 0$
10.	Эллипс: $a = 0.2$, $b = 0.2$, $\alpha = 0$, $x_0 = 0.2$, $y_0 = 0$ Эллипс: $a = 0.7$, $b = 0.5$, $\alpha = -\pi/3$, $x_0 = 0$, $y_0 = 0$ Эллипс: $a = 1$, $b = 1$, $\alpha = 0$, $x_0 = 0$, $y_0 = 0$
11.	Эллипс: $a = 0.4$, $b = 0.15$, $\alpha = \pi/3$, $x_0 = -0.2$, $y_0 = -0.18$ Эллипс: $a = 0.7$, $b = 0.5$, $\alpha = -\pi/3$, $x_0 = -0.2$, $y_0 = -0.18$ Эллипс: $a = 1$, $b = 1$, $\alpha = 0$, $x_0 = 0$, $y_0 = 0$
12.	Эллипс: $a = 0.2$, $b = 0.2$, $\alpha = \pi/3$, $x_0 = 0$, $y_0 = 0.4$ Эллипс: $a = 0.7$, $b = 0.5$, $\alpha = -\pi/3$, $x_0 = 0.2$, $y_0 = 0.18$ Эллипс: $a = 1$, $b = 1$, $\alpha = 0$, $x_0 = 0$, $y_0 = 0$

№	Алгебраические линии
13.	Эллипс: $a = 0.3$, $b = 0.3$, $\alpha = 0$, $x_0 = 0$, $y_0 = 0$ Эллипс: $a = 0.7$, $b = 0.7$, $\alpha = 0$, $x_0 = 0$, $y_0 = 0$ Парабола: $p = 1$, $\alpha = 0$, $x_0 = -0.8$, $y_0 = 0$
14.	Эллипс: $a = 0.4$, $b = 0.15$, $\alpha = \pi/6$, $x_0 = 0$, $y_0 = 0$ Эллипс: $a = 0.7$, $b = 0.7$, $\alpha = 0$, $x_0 = 0$, $y_0 = 0$ Парабола: $p = 1$, $\alpha = \pi/2$, $x_0 = 0$, $y_0 = -0.8$
15.	Эллипс: $a = 0.4$, $b = 0.15$, $\alpha = \pi/6$, $x_0 = 0$, $y_0 = 0$ Эллипс: $a = 0.7$, $b = 0.5$, $\alpha = \pi/3$, $x_0 = 0$, $y_0 = 0$ Парабола: $p = 1$, $\alpha = \pi/2$, $x_0 = 0$, $y_0 = -0.8$
16.	Эллипс: $a = 0.4$, $b = 0.15$, $\alpha = \pi/6$, $x_0 = 0.1$, $y_0 = -0.15$ Эллипс: $a = 0.7$, $b = 0.5$, $\alpha = \pi/3$, $x_0 = 0$, $y_0 = 0$ Парабола: $p = 1$, $\alpha = \pi/2$, $x_0 = 0$, $y_0 = -0.8$
17.	Эллипс: $a = 0.4$, $b = 0.15$, $\alpha = \pi/6$, $x_0 = 0.1$, $y_0 = -0.15$ Эллипс: $a = 0.7$, $b = 0.5$, $\alpha = \pi/3$, $x_0 = 0$, $y_0 = 0$ Парабола: $p = 1$, $\alpha = \pi/2$, $x_0 = -0.8$, $y_0 = 0$
18.	Эллипс: $a = 0.4$, $b = 0.4$, $\alpha = 0$, $x_0 = 0.1$, $y_0 = -0.15$ Эллипс: $a = 0.7$, $b = 0.7$, $\alpha = 0$, $x_0 = 0$, $y_0 = 0$ Парабола: $p = 1$, $\alpha = 0$, $x_0 = -0.8$, $y_0 = 0$
19.	Эллипс: $a = 0.4$, $b = 0.4$, $\alpha = 0$, $x_0 = 0.1$, $y_0 = -0.15$ Эллипс: $a = 0.7$, $b = 0.7$, $\alpha = 0$, $x_0 = 0$, $y_0 = 0$ Парабола: $p = -1$, $\alpha = 0$, $x_0 = 0.8$, $y_0 = 0$
20.	Эллипс: $a = 0.4$, $b = 0.4$, $\alpha = 0$, $x_0 = -0.1$, $y_0 = 0.15$ Эллипс: $a = 0.7$, $b = 0.7$, $\alpha = 0$, $x_0 = 0$, $y_0 = 0$ Парабола: $p = -1$, $\alpha = 0$, $x_0 = 0.8$, $y_0 = 0$
21.	Эллипс: $a = 0.5$, $b = 0.2$, $\alpha = \pi/3$, $x_0 = 0$, $y_0 = 0$ Эллипс: $a = 0.7$, $b = 0.7$, $\alpha = 0$, $x_0 = 0.08$, $y_0 = 0.05$ Парабола: $p = -1$, $\alpha = -\pi/2$, $x_0 = 0$, $y_0 = -0.8$
22.	Эллипс: $a = 0.5$, $b = 0.5$, $\alpha = 0$, $x_0 = 0$, $y_0 = 0$ Эллипс: $a = 0.8$, $b = 0.8$, $\alpha = 0$, $x_0 = 0$, $y_0 = 0$ Эллипс: $a = 1$, $b = 1$, $\alpha = 0$, $x_0 = 0$, $y_0 = 0$
23.	Эллипс: $a = 0.4$, $b = 0.5$, $\alpha = 0$, $x_0 = 0.05$, $y_0 = 0$ Эллипс: $a = 0.6$, $b = 0.6$, $\alpha = 0$, $x_0 = 0$, $y_0 = 0$ Эллипс: $a = 0.8$, $b = 1$, $\alpha = 0$, $x_0 = 0$, $y_0 = 0$

№	Алгебраические линии
24.	Эллипс: $a = 0.3$, $b = 0.3$, $\alpha = 0$, $x_0 = 0$, $y_0 = 0.35$ Эллипс: $a = 0.5$, $b = 0.5$, $\alpha = 0$, $x_0 = 0$, $y_0 = 0.35$ Эллипс: $a = 0.8$, $b = 1$, $\alpha = 0$, $x_0 = 0$, $y_0 = 0$
25.	Эллипс: $a = 0.3$, $b = 0.3$, $\alpha = 0$, $x_0 = 0$, $y_0 = 0.2$ Эллипс: $a = 0.5$, $b = 0.5$, $\alpha = 0$, $x_0 = 0$, $y_0 = 0.2$ Эллипс: $a = 0.8$, $b = 1$, $\alpha = 0$, $x_0 = -0.1$, $y_0 = -0.1$
26.	Эллипс: $a = 0.3$, $b = 0.3$, $\alpha = 0$, $x_0 = 0$, $y_0 = 0$ Парабола: $p = 0.5$, $\alpha = 0$, $x_0 = -0.5$, $y_0 = 0$ Парабола: $p = 1$, $\alpha = 0$, $x_0 = -0.8$, $y_0 = 0$
27.	Парабола: $p = 0.3$, $\alpha = 0$, $x_0 = 0.$, $y_0 = 0$ Парабола: $p = 0.5$, $\alpha = 0$, $x_0 = -0.5$, $y_0 = 0$ Парабола: $p = 1$, $\alpha = 0$, $x_0 = -0.8$, $y_0 = 0$
28.	Парабола: $p = -0.3$, $\alpha = 0$, $x_0 = 0.$, $y_0 = 0$ Парабола: $p = -0.4$, $\alpha = 0$, $x_0 = 0.4$, $y_0 = 0$ Парабола: $p = -0.5$, $\alpha = 0$, $x_0 = 0.65$, $y_0 = 0$
29.	Эллипс: $a = 0.3$, $b = 0.15$, $\alpha = -\pi/6$, $x_0 = -0.05$, $y_0 = -0.05$ Эллипс: $a = 0.7$, $b = 0.5$, $\alpha = \pi/3$, $x_0 = 0$, $y_0 = 0$ Парабола: $p = 1$, $\alpha = \pi/2$, $x_0 = 0$, $y_0 = -0.8$
30.	Эллипс: $a = 0.4$, $b = 0.15$, $\alpha = \pi/6$, $x_0 = 0$, $y_0 = 0$ Парабола: $p = 0.7$, $\alpha = -\pi/3$, $x_0 = -0.2$, $y_0 = 0.4$ Парабола: $p = 1$, $\alpha = -\pi/3$, $x_0 = -0.4$, $y_0 = 0.6$

Этапы 2 и 3

2. Задан обучающий набор $\{x(i), y(i)\}$. Построить и обучить двухслойную нейронную сеть прямого распространения, которая будет выполнять аппроксимацию функции вида

$$\hat{y}(i) = f[x(i)]$$

Для обучения использовать алгоритм, реализующий метод поиска экстремума функции многих переменных первого порядка. Функция и метод обучения определяются вариантом задания.

2.1 Создать сеть с помощью функции *feedforwardnet*. Сконфигурировать сеть под обучающее множество с помощью функции *configure*. Число нейронов скрытого слоя задать равным 10. Использовать активационные функции, заданные по умолчанию (*tansig, purelin*). Алгоритм обучения определяется вариантом задания.

2.2 Для разделения обучающией выборки на обучающее, контрольное, и тестовое подмножества использовать функцию *divideind*. Выделить с конца временной последовательности 10% отсчетов на контрольное подмножество. Тестовое подмножество оставить пустым.

$$net.divideParam.testInd = []$$

2.3 Инициализировать сеть (*init*) с помощью функции, заданной по умолчанию.

2.4 Задать параметры обучения: значения параметров для некоторых методов обучения описаны выше, число эпох обучения (*net.trainParam.epochs*) и число эпох, в течение которых может расти ошибка на контрольном подмножестве (*net.trainParam.max_fail*), равными 600, предельное значение критерия обучения (*net.trainParam.goal*) равным 10^{-8},

2.5 Выполнить обучение сети с помощью функции *train*. Если необходимо, то произвести обучение несколько раз. Если результаты неудовлетворительные или наблюдается переобучение, то изменить число нейронов в функции *feedforwardnet*, увеличить число эпох обучения или уменьшить предельное значение критерия обучения. Занести в отчет весовые коэффициенты и смещения для двух слоев. Занести в отчет окна Performance и Neural Network Training, если это возможно для данного метода обучения.

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

2.7 Рассчитать выход сети (*sim*) для обучающего подмножества. Сравнить выход сети с соответствующим эталонным подмножеством: рассчитать показатели качества обучения и заполнить таблицу 2. Отобразить на графике эталонные значения и предсказанные сетью, а также ошибку обучения. Графики занести в отчет.

2.8 Проделать тоже самое для контрольного подмножества.

3. Построить и обучить двухслойную нейронную сеть прямого распространения, которая будет выполнять аппроксимацию функции. Для обучения использовать алгоритм, реализующий метод оптимизации функций многих переменных второго порядка. Функция и метод обучения определяются вариантом задания.

Последовательности шагов для выполнения 2 и 3 этапов работы совпадают.

Варианты заданий:

Номер варианта соответствует номеру студента в списке группы.

№	Функция	Методы обучения
1.	$x = \sin\left(t^2\right), \quad t \in [0, 4],\ h = 0.02$	*trainrp, trainoss*
2.	$x = \sin\left(t^2 - 2t + 3\right), \quad t \in [0, 6],\ h = 0.025$	*traincgb, trainlm*
3.	$x = \cos\left(2.5t^2 - 5t\right), \quad t \in [0, 2.2],\ h = 0.01$	*trainscg, trainbfg*
4.	$x = \sin\left(\sin(t)t^2 + 5t\right), \quad t \in [0, 3.5],\ h = 0.01$	*traingd, trainbfg*
5.	$x = \cos\left(-3t^2 + 5t + 10\right), \quad t \in [0, 2.5],\ h = 0.01$	*traingdm, trainoss*
6.	$x = \sin\left(0.5t^2 - 5t\right), \quad t \in [0, 2],\ h = 0.01$	*traincgp, trainlm*
7.	$x = \sin\left(\sin(t)t^2 - t\right), \quad t \in [1, 4.5],\ h = 0.01$	*traincgp, trainbfg*
8.	$x = \sin\left(-2t^2 + 7t\right), \quad t \in [0, 3.5],\ h = 0.01$	*traingdx, trainoss*
9.	$x = \sin\left(t^2 - 2t + 5\right), \quad t \in [0, 5],\ h = 0.025$	*traingd, trainbfg*

№	T	trainFcn
10.	$x = \sin\left(t^2 - 7t\right), \quad t \in [0, 5],\ h = 0.025$	*trainscg, trainoss*
11.	$x = \cos\left(t^2\right), \quad t \in [0, 4],\ h = 0.02$	*traingda, trainbfg*
12.	$x = \cos\left(-\cos(t)t^2 + t\right), \quad t \in [0.5, 4],\ h = 0.01$	*traincgf, trainlm*
13.	$x = \cos\left(t^2 - 2t + 3\right), \quad t \in [0, 5],\ h = 0.02$	*traingdx, trainbfg*
14.	$x = \cos\left(t^2 - 10t + 3\right), \quad t \in [1, 6],\ h = 0.025$	*trainrp, trainlm*
15.	$x = \cos\left(-2t^2 + 7t\right), \quad t \in [0, 3.5],\ h = 0.01$	*traingda, trainoss*
16.	$x = \sin\left(\sin(t)t^2 + 3t - 10\right), \quad t \in [2.5, 5],\ h = 0.01$	*traincgb, trainbfg*
17.	$x = \cos\left(-5t^2 + 10t - 5\right), \quad t \in [0, 2.5],\ h = 0.01$	*trainscg, trainoss*
18.	$x = \cos\left(\cos(t)t^2 - t\right), \quad t \in [1, 4.5],\ h = 0.01$	*traincgf, trainlm*
19.	$x = \sin\left(-5t^2 + 10t - 5\right), \quad t \in [0, 2.5],\ h = 0.01$	*traingda, trainoss*
20.	$x = \cos\left(\cos(t)t^2 + 5t\right), \quad t \in [0, 3.5],\ h = 0.01$	*traingdx, trainlm*
21.	$x = \sin\left(t^2 - 10t + 3\right), \quad t \in [1, 6],\ h = 0.025$	*traingdx, trainoss*
22.	$x = \sin\left(-\sin(t)t^2 + t\right), \quad t \in [0.5, 4],\ h = 0.01$	*traincgb, trainlm*
23.	$x = \sin\left(t^2 - 6t + 3\right), \quad t \in [0, 5],\ h = 0.025$	*traingd, trainbfg*
24.	$x = \sin\left(0.66\pi t\right), \quad t \in [0, 5],\ h = 0.025$	*traingdx, trainoss*
25.	$x = \sin\left(t^2 - 6t + 3\right), \quad t \in [0, 6],\ h = 0.025$	*traincgp, trainlm*
26.	$x = \sin\left(\sin(t)t^2\right), \quad t \in [0, 3.5],\ h = 0.01$	*trainrp, trainbfg*
27.	$x = \sin\left(t^2 - 5t + 6\right), \quad t \in [0, 6],\ h = 0.02$	*traingda, trainoss*
28.	$x = \sin\left(2.5t^2 - 5t\right), \quad t \in [0, 2.2],\ h = 0.01$	*traincgf, trainbfg*

№	T	trainFcn
29.	$x = \sin\bigl(2t^2 - 6t + 3\bigr), \quad t \in [0, 5],\ h = 0.02$	*traincgp, trainlm*
30.	$x = \sin\bigl(-3t^2 + 5t + 10\bigr), \quad t \in [0, 2.5],\ h = 0.01$	*trainscg, trainlm*

Литература

1. *Beale M., Hagan M., Demuth H.* Neural Network Toolbox User's guide R2011b. The MathWorks, 2011. – –pp. 2-2–2-32.

2. *Медведев В. С., Потемкин В. Г.* Нейронные сети. MATLAB 6/Под общ. ред. к. т. н. В. Г. Потемкина – М.: ДИАЛОГ-МИФИ, 2006. – с. 47–101.

3. *Hagan M., Demuth H.* Neural Network Design. 1996. – Chapter 11 and 12. –pp. 11-1–12-52.

4. *Бортаковский А. С., Пантелеев А. В.* Аналитическая геометрия в примерах и задачах: Учеб. пособие. – М.: Высш. шк., 2005. – с. 135–137, 268-289.

Лабораторная работа № 4.
Сети с радиальными базисными элементами

Целью работы является исследование свойств некоторых видов сетей с радиальными базисными элементами, алгоритмов обучения, а также применение сетей в задачах классификации и аппроксимации функции.

Основные этапы работы:

1. Использовать вероятностную нейронную сеть для классификации точек в случае, когда классы не являются линейно разделимыми.

2. Использовать сеть с радиальными базисными элементами (RBF) для классификации точек в случае, когда классы не являются линейно разделимыми.

3. Использовать обобщенно-регрессионную нейронную сеть для аппроксимации функции. Проверить работу сети с рыхлыми данными.

Сценарий работы:

1. Для трех линейно неразделимых классов из лабораторной работы № 3 решить задачу классификации. Точки, принадлежащие одному классу, лежат на алгебраической линии. Построить вероятностную сеть, которая будет классифицировать точки заданной области.

Обучающий набор $\{x_i, y_i\},\ i = 1, .., N$, число классов $K = 3$. Сеть реализует отображение вида:

$$f(x_i, y_i) = \{(z_k)_{k=1}^{K} = (0, .., 1, ..0)|\ z_{k=K^*} = 1 \text{ при } (x_i, y_i) \in K^*\}$$

1.1 В соответствии с вариантом задания для каждой линии сгенерировать множество точек. Далее для первого класса выбрать из исходного множества случайным образом 60 точек. Для второго и третьего классов 100 и 120 точек соответственно.

1.2 Множество точек, принадлежащее каждому классу, разделить на обучающее и тестовое подмножества с помощью функции dividerand в отношении 80%-20%.

1.3 Способом, описанным в Л.р. № 3, отобразить множества точек для каждого класса, а также соответствующие обучающие и тестовые подмножества.

1.4 Соответствующие подмножества точек объединить в обучающее и тестовое подмножества обучающей выборки.

1.5 Эталонное распределение точек обучающей выборки по классам преобразовать к индексам ($ind2vec$).

1.6 Константу $SPREAD$ задать равной 0.3. Создать сеть с помощью функции $newpnn$. Подать в сеть обучающее подмножество обучающей выборки.

1.7 Отразить структуру сети, заполнив таблицу 1.

1.8 Проверить качество обучения: рассчитать выход сети для обучающего подмножества обучающей выборки. Преобразовать выходные значения с помощью функции ($vec2ind$). Занести в отчет количество правильно классифицированных точек.

1.9 Провести аналогичные расчеты для тестового подмножеств.

1.10 Произвести классификацию точек области $[-1.2, 1.2] \times [-1.2, 1.2]$. Закодировать принадлежности к классам различными цветами и занести полученное изображение в отчет. Для этого использовать методику, описанную в лабораторной работе № 3.

1.11 Константу $SPREAD$ задать равной 0.1. Создать сеть с помощью функции $newpnn$.

1.12 Произвести классификацию точек области $[-1.2, 1.2] \times [-1.2, 1.2]$. Закодировать принадлежности к классам различными цветами и занести полученное изображение в отчет. Использовать методику, описанную в лабораторной работе № 3.

2. Для трех линейно неразделимых классов из лабораторной работы № 3 решить задачу классификации. Точки, принадлежащие одному классу, лежат на алгебраической линии. Построить сеть с радиальными базисными элементами, которая будет классифицировать точки заданной области.

2.1 В соответствии с вариантом задания для каждой линии сгенерировать множество точек. Далее для первого класса выбрать из исходного множества случайным образом 60 точек. Для второго и третьего классов 100 и 120 точек соответственно.

2.2 Множество точек, принадлежащее каждому классу, разделить на обучающее и тестовое подмножества с помощью функции dividerand в отношении 80%-20%.

2.3 Способом, описанным в Л.р. № 3, отобразить множества точек для каждого класса, а также соответствующие обучающие и тестовые подмножества.

2.4 Соответствующие подмножества точек объединить в обучающее и тестовое подмножества обучающей выборки.

2.5 Создать сеть с помощью $newrb$, задав следующие параметры: предельное значение критерия обучения ($goal$) — 10^{-5}, $SPREAD$ — 0.3, размер обучающей выборки — число элементов в обучающем подмножестве. В сеть подается обучающее подмножество обучающей выборки.

2.6 Занести в отчет окно Training with newrb. Отразить структуру сети, заполнив таблицу 1. Указать число радиальных базисных нейронов.

2.7 Проверить качество обучения: рассчитать выход сети для обучающего подмножества обучающей выборки. Занести в отчет количество правильно классифицированных точек.

2.8 Провести аналогичные расчеты для тестового подмножеств.

2.9 Произвести классификацию точек области $[-1.2, 1.2] \times [-1.2, 1.2]$. Закодировать принадлежности к классам различными цветами и занести полученное изображение в отчет. Для этого использовать методику, описанную в лабораторной работе № 3.

2.10 Константу $SPREAD$ задать равной 0.1. Создать сеть с помощью функции $newrb$.

2.11 Произвести классификацию точек области $[-1.2, 1.2] \times [-1.2, 1.2]$. Закодировать принадлежности к классам различными цветами и занести полученное изображение в отчет. Использовать методику, описанную в лабораторной работе № 3.

3. Задан обучающий набор $\{x(i), y(i)\}$. Построить обобщенно-регрессионную нейронную сеть, которая будет выполнять аппроксимацию функции

$$\hat{y}(i) = f[x(i)]$$

Функцию, соответствующего варианта, взять из лабораторной работы № 3.

3.1 Создать сеть с помощью функции *newgrnn(P1,T1,SPREAD)*. Константу *SPREAD* задать равной h, где h — величина шага для заданной функции.

3.2 Произвести разделение обучающей выборки на обучающее и тестовое подмножества. Индексы обучающего подмножества использовать для создания сети.

$$P1 = P(trainInd);$$
$$T1 = T(trainInd);$$

Выделить с конца временной последовательности 10% отсчетов на тестовое подмножество.

3.3 Если результаты неудовлетворительные, то изменить значение *SPREAD* и создать новую сеть.

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

3.5 Рассчитать выход сети (*sim*) для обучающего подмножества. Сравнить выход сети с соответствующим эталонным подмножеством: рассчитать показатели качества обучения и заполнить таблицу 2. Отобразить на графике эталонные значения и предсказанные сетью. Отобразить на отдельном графике ошибку обучения. Графики занести в отчет.

3.6 Получить апостериорную оценку качества работы сети: проделать аналогичные действия для тестового подмножества.

3.7 Сформировать обучающее множество с рыхлыми данными. Для этого произвести разделение обучающей выборки на обучающее и тестовое подмножества. с помощью функции (*dividerand*) в соотношении 80% и 20%.

3.8 Рассчитать выход сети (*sim*) для обучающего подмножества. Сравнить выход сети с соответствующим эталонным подмножеством: рассчитать показатели качества обучения и заполнить таблицу 2. Отобразить на графике эталонные значения и предсказанные сетью, а также ошибку обучения. Графики занести в отчет.

Литература

1. *Beale M., Hagan M., Demuth H.* Neural Network Toolbox User's guide R2011b. The MathWorks, 2011. –pp. 5-2–5-16.

2. *Медведев В. С., Потемкин В. Г.* Нейронные сети. MATLAB 6/Под общ. ред. к. т. н. В. Г. Потемкина – М.: ДИАЛОГ-МИФИ, 2006. – с. 131–146.

3. *Круглов В. В., Дли М. И., Голунов Р. Ю.* Нечеткая логика и искусственные нейронные сети. – М.: Физматлит, 2001. – с. 94–100.

Лабораторная работа № 5.
Сети с обратными связями

Целью работы является исследование свойств сетей Хопфилда, Хэмминга и Элмана, алгоритмов обучения, а также применение сетей в задачах распознавания статических и динамических образов.

Основные этапы работы:

1. Использовать сеть Элмана для распознавания динамических образов. Проверить качество распознавания.

2. Использовать сеть Хопфилда для распознавания статических образов. Проверить качество распознавания.

3. Использовать сеть Хэмминга для распознавания статических образов. Проверить качество распознавания.

Сценарий работы:

Этап 1

1. Построить и обучить сеть Элмана, которая будет выполнять распознавание динамического образа. Проверить качество распознавания.

1.1 Входная последовательность обучающего множества состоит из комбинации основного сигнала (p_1) и сигнала, подлежащего распознаванию (p_2). Каждому значению основного сигнала соответствует -1 целевого выхода, каждому значению сигнала p_2 соответствует 1 целевого выхода.

$$p_1(k) = \sin(4\pi k), \quad t_1(k) = -1, \quad k \in [0, 1] \text{ с шагом } h = 0.025$$
$$p_2(k) = g(k), \quad t_2(k) = 1, \quad k \in [a_2, b_2] \text{ с шагом } h = 0.025$$

Функция $g(k)$ определяется вариантом задания. Длительность основного сигнала задается набором чисел $R = \{r_1, r_2, r_3\}$. Значения R также определяются вариантом задания. Входное множество формируется по формуле

$$P = [repmat(p_1, 1, r_1), p_2, repmat(p_1, 1, r_2), p_2, repmat(p_1, 1, r_3), p_2]$$
$$T = [repmat(t_1, 1, r_1), t_2, repmat(t_1, 1, r_2), t_2, repmat(t_1, 1, r_3), t_2]$$

Преобразовать обучающее множество с помощью функции $con2seq$. Не выделять из обучающего множества контрольное и тестовое подмножества.

1.2 Создать сеть с помощью функции $layrecnet$. Задать задержки $1 : 2$. Число нейронов скрытого слоя задать равным 8. Для обучения сети использовать одношаговый метод секущих ($trainoss$). Для скрытого и выходного слоев использовать $tansig$ в качестве активационной функции ($net.layers\{i\}.transferFcn$). Сконфигурировать сеть ($configure$) под обучающее множество.

1.3 С помощью функции $preparets$ сформировать массивы ячеек для функции обучения, содержащие обучающее множество и значения для инициализации задержек обратной связи (P, T, Pi, Ai соответственно). Если при выполнении заданий используется версия MATLAB, которая не поддерживает эту функцию, то обучать и выполнять расчет выходов сети без инициализации задержек.

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 Произвести обучение сети. Если необходимо, то произвести обучение несколько раз. Если результаты неудовлетворительные, то увеличить число нейронов сети. Занести в отчет содержимое Performance и Neural Network Training.

1.6 Отразить структуру сети и проведенное обучение, заполнив таблицу 1.

1.7 Рассчитать выход сети (sim) для обучающего подмножества. Отобразить на графике эталонные значения и предсказанные сетью. С помощью функции *legend* подписать кривые.

1.8 Преобразовать значения по правилу

$$o_{ij} = \begin{cases} 1, & a_{ij} \geqslant 0; \\ -1, & a_{ij} < 0; \end{cases}$$

Сравнить выход сети с эталонными значениями. Занести в отчет количество правильно классифицированных точек.

1.9 Для проверки качества распознавания сформировать новое обучающее множество, изменив одно из значений $R = \{r_1, r_2, r_3\}$. Рассчитать выходы сети для измененной входной последовательности.

1.10 Рассчитать выход сети (sim) для обучающего подмножества. Отобразить на графике эталонные значения и предсказанные сетью. С помощью функции *legend* подписать кривые.

1.11 Преобразовать значения по правилу. Сравнить выход сети с эталонными значениями. Занести в отчет количество правильно классифицированных точек.

Варианты заданий:

Номер варианта соответствует номеру студента в списке группы.

№	Динамический образ	Длительность $p_1(k)$
1.	$g(k) = \sin(-3k^2 + 10k - 5), \quad k \in [0.62, 3.14]$	$[0, 8, 6]$
2.	$g(k) = \cos(-2k^2 + 7k), \quad k \in [0.92, 4.07]$	$[2, 4, 7]$
3.	$g(k) = \sin(k^2 - 15k + 3) - \sin^2(k) + 0.5, \quad k \in [0.9, 3.1]$	$[3, 5, 2]$
4.	$g(k) = \sin(\sin(k)k^3 - 10), \quad k \in [1.56, 3.12]$	$[0, 1, 5]$
5.	$g(k) = 1.5\sin(\sin(k)k^2) - 0.5, \quad k \in [0.74, 3.14]$	$[3, 3, 4]$
6.	$g(k) = \sin(k^2 - 5k + 6), \quad k \in [0.67, 4.98]$	$[2, 6, 5]$
7.	$g(k) = \cos(\cos(k)k^2 - k), \quad k \in [2.16, 4.04]$	$[1, 4, 7]$
8.	$g(k) = \cos(\cos(k)k^2 + 5k), \quad k \in [2.38, 4.1]$	$[1, 3, 5]$
9.	$g(k) = \sin(\sin(k)k^2 - k), \quad k \in [1.13, 3.6]$	$[7, 0, 7]$

№	G	R
10.	$g(k) = \sin(-3k^2 + 5k + 10) + 0.8, \quad k \in [0.46, 3.01]$	$[0, 2, 2]$
11.	$g(k) = \cos(-\cos(k)k^2 + k), \quad k \in [2.9, 4.55]$	$[6, 7, 1]$
12.	$g(k) = \sin(\sin(k)k^2 + 5k), \quad k \in [1.86, 3.86]$	$[4, 3, 0]$
13.	$g(k) = \sin(2k^2 - 6k + 3), \quad k \in [-0.02, 2.36]$	$[2, 5, 6]$
14.	$g(k) = \sin(\sin(k)k^2 + 3k - 10), \quad k \in [4.45, 5.86]$	$[6, 5, 7]$
15.	$g(k) = \cos(-2k^2 + 7k), \quad k \in [0.92, 3.25]$	$[0, 4, 2]$
16.	$g(k) = \sin(\sin(k)k^2) - 0.1, \quad k \in [0.48, 2.71]$	$[7, 0, 3]$
17.	$g(k) = \sin(2.5k^2 - 5k), \quad k \in [-1.14, 1.16]$	$[5, 5, 4]$
18.	$g(k) = \cos(-5k^2 + 10k - 5), \quad k \in [0.45, 2.48]$	$[2, 1, 4]$
19.	$g(k) = \sin(-\sin(k)k^2 + k), \quad k \in [0.01, 2.77]$	$[3, 1, 3]$
20.	$g(k) = 1.5\sin(-5k^2 + 10k - 5) + 0.4, \quad k \in [0.78, 2.35]$	$[2, 2, 5]$
21.	$g(k) = \sin(\sin(k)k^2 - k), \quad k \in [1.12, 3.6]$	$[3, 0, 5]$
22.	$g(k) = \cos(-3k^2 + 5k + 10), \quad k \in [0.24, 2.7]$	$[2, 4, 4]$
23.	$g(k) = \sin(-2k^2 + 7k), \quad k \in [0.01, 2.96]$	$[3, 4, 6]$
24.	$g(k) = \cos(k^2 - 10k + 3), \quad k \in [2.84, 6.25]$	$[3, 4, 6]$
25.	$g(k) = 1.5\sin(k^2 - 6k + 3) - 0.8, \quad k \in [1.49, 3.52]$	$[5, 3, 3]$
26.	$g(k) = \sin(k^2 - 10k + 3), \quad k \in [2.5, 4.84]$	$[1, 2, 3]$
27.	$g(k) = \sin(k^2 - 2k + 3), \quad k \in [-0.05, 4.25]$	$[0, 1, 6]$
28.	$g(k) = \cos(\cos(k)k^2), \quad k \in [2.47, 4.26]$	$[7, 1, 3]$

№	G	R
29.	$g(k) = \sin(-2\sin(k)k^2 + 7), \quad k \in [1.41, 3.1]$	$[2, 3, 8]$
30.	$g(k) = \sin(-2k^2 + 7k) - 0.5\sin(k), \quad k \in [0.01, 2.98]$	$[4, 5, 2]$

Этап 2

2. Построить сеть Хопфилда, которая будет хранить образы из заданного набора. Эталонными образами являются двоичные изображения цифр $0, 1, 2, 3, 4, 6, 9$ (рис. 3) размером 12x10. Проверить работу сети с зашумленными образами.

2.1 Создать сеть с помощью функции $newhop$. Аттракторами построенной сети должны быть 3 образа, которые определяются вариантом задания. Каждый эталонный образ задается матрицей. Цветам точек соответствуют -1 и 1. Для синтеза сети необходимо объединить эталонные образы по формуле $T = [p1(:), p2(:), p3(:)]$.

2.2 Подать в сеть первый образ, рассчитать выход сети. Число итераций задать равным 600. Результат распознавания занести в отчет. Для этого с помощью функции $reshape(p1, 12, 10)$ преобразовать выход сети и заменить в полученной матрице значения по правилу

$$x_{ij} = \begin{cases} 2, & a_{ij} \geqslant 0; \\ 1, & a_{ij} < 0; \end{cases}$$

Для отображения результата распознавания использовать вызов следующих функций:

```
map = [1, 1, 1; 0, 0, 0];
image(X); colormap(map)
axis off
axis image
```

2.3 Произвести зашумление второго образа на 20%, полученный образ занести в отчет. Рассчитать выход сети. Результат распознавания занести в отчет.

Зашумление произвести следующим образом: для каждой точки изображения изменить цвет по правилу

if $r_{ij} < M$ *then инвертировать цвет точки*

где M — степень зашумления, r —реализация случайной величины, распределенной по равномерному закону (функция *rand*).

2.4 Произвести зашумление третьего образа на 30%, полученный образ занести в отчет. Рассчитать выход сети. Число итераций задать равным 600. Если необходимо, то произвести обучение несколько раз. Если результаты распознавания неудовлетворительные, то увеличить число итераций. Результат распознавания занести в отчет.

Варианты заданий:

Номер варианта соответствует номеру в списке группы.

№	Цифры
1.	$[1, 0, 6]$
2.	$[1, 6, 4]$

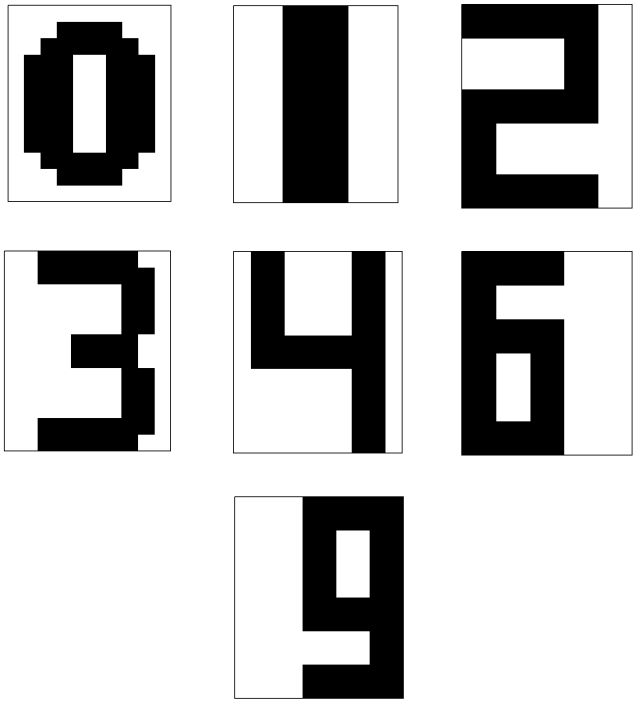

Рис. 2. Эталонные образы

№	Цифры
3.	$[4, 3, 2]$
4.	$[6, 3, 9]$
5.	$[9, 0, 4]$
6.	$[9, 2, 3]$
7.	$[3, 1, 0]$
8.	$[2, 4, 1]$
9.	$[0, 9, 2]$
10.	$[4, 3, 0]$
11.	$[9, 6, 1]$
12.	$[2, 1, 6]$

№	Цифры
13.	$[0, 1, 4]$
14.	$[6, 2, 3]$
15.	$[4, 2, 9]$
16.	$[3, 6, 0]$
17.	$[0, 1, 3]$
18.	$[1, 4, 2]$
19.	$[9, 3, 0]$
20.	$[6, 2, 9]$
21.	$[3, 0, 4]$
22.	$[9, 1, 3]$
23.	$[6, 9, 2]$
24.	$[6, 1, 0]$
25.	$[0, 2, 3]$
26.	$[4, 0, 6]$
27.	$[3, 4, 2]$
28.	$[2, 1, 6]$
29.	$[9, 3, 2]$
30.	$[1, 4, 0]$

Этап 3

3. Построить сеть Хэммиинга, которая будет хранить образы из заданного набора. Эталонными образами являются двоичные изображения цифр $0, 1, 2, 3, 4, 6, 9$ (рис. 3) размером 12x10. Проверить работу сети с зашумленными образами.

3.1 Реализовать сеть Хэмминга. Сеть Хэмминга является двухслойной сеть прямого распространения. Функционирование сети производится в соответствии с правилами:

$$IW = \begin{pmatrix} p_1^T \\ \vdots \\ p_Q^T \end{pmatrix} \qquad b^1 = \begin{pmatrix} R \\ \vdots \\ R \end{pmatrix} \qquad a^1 = IW * p + b^1$$

$$LW = \begin{pmatrix} 1 & -\varepsilon & \dots & -\varepsilon \\ -\varepsilon & 1 & \dots & -\varepsilon \\ \dots & & & \\ -\varepsilon & -\varepsilon & \dots & 1 \end{pmatrix} \qquad a^2(k) = poslin\left(LW * a^2(k-1)\right)$$

где Q — число эталонных образов, $\varepsilon = 1/(Q-1)$, R — размерность входного вектора.

3.2 Первый слой вычисляет расстояние Хэмминга между входным и эталонными векторами. Вычисления, проводимые в первом слое, реализовать по приведенному правилу.

3.3 Для реализации работы второго слоя использовать сеть Хопфилда. Создать сеть с помощью функции $newhop(a^1)$. Использовать $poslin$ в качестве активационной функции ($net.layers\{1\}.transferFcn$). Весовые коэффициенты и смещения (LW^{11}, b^1) задать по приведенным правилам.

3.4 Подать в сеть первый образ. Число итераций задать равным 600 и рассчитать выход сети. В результате работы сети в выходном векторе должна быть одна ненулевая компонента. Если ненулевых компонент несколько, то выбрать наибольшую компоненту. Индекс этой компоненты соответствует строке матрицы IW, содержащей эталонный образ. Занести выход сети и номер образа в отчет.

3.5 Рассчитать выход сети для зашумленного на 20% образа из Этапа 2. Занести выход сети и номер образа в отчет.

3.6 Рассчитать выход сети для зашумленного на 30% образа из Этапа 2. Занести выход сети и номер образа в отчет.

Литература

1. *Beale M., Hagan M., Demuth H.* Neural Network Toolbox User's guide R2011b. The MathWorks, 2011. –pp. 3-29–3-31, 9-34–9-41.

2. *Медведев В. С., Потемкин В. Г.* Нейронные сети. MATLAB 6/Под общ. ред. к. т. н. В. Г. Потемкина – М.: ДИАЛОГ-МИФИ, 2006. – с. 175–188.

3. *Осовский С.* Нейронные сети для обработки информации. – М.: Финансы и статистика, 2002. – с. 210–219.

4. *Круглов В. В., Дли М. И., Голунов Р. Ю.* Нечеткая логика и искусственные нейронные сети. – М.: Физматлит, 2001. – с. 90–94.

Лабораторная работа № 6.
Сети Кохонена

Целью работы является исследование свойств слоя Кохонена, карты Кохонена, а также сетей векторного квантования, обучаемых с учителем, алгоритмов обучения, а также применение сетей в задачах кластеризации и классификации.

Основные этапы работы:

1. Использовать слой Кохонена для выполнения кластеризации множества точек. Проверить качество разбиения.

2. Использовать карту Кохонена для выполнения кластеризации множества точек.

3. Использовать карту Кохонена для нахождения одного из решений задачи коммивояжера.

4. Использовать сеть векторного квантования, обучаемую с учителем, (LVQ-сеть) для классификации точек в случае, когда классы не являются линейно разделимыми.

Сценарий работы:

1. Построить и обучить слой Кохонена, который будет содержать координаты центров 8 сформированных кластеров.

1.1 Сформировать множество случайных точек, которые изначально сгруппированы в 8 кластеров. Использовать функцию $nngenc(X, clusters, points, deviation)$, где $X = [0, 1.5; 0, 1.5]$ задает область, в которой находятся точки множества; $clusters = 8$ задает число кластеров; $points = 10$ задает число точек в каждом из кластеров; $deviation = 0.1$ задает среднее квадратическое отклонение от центра кластера.

1.2 Не разбивать обучающее множество на подмножества.

1.3 Создать сеть с помощью функции $competlayer$. Число кластеров задать равным 8. Сконфигурировать сеть под обучающее множество.

1.4 Задать число эпох обучения $net.trainParam.epochs$ равным 50. Произвести обучение сети с помощью метода, заданного по умолчанию. Занести в отчет весовые коэффициенты первого слоя, которые являются центрами кластеров. Занести в отчет содержимое окна Neural Network Training.

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

1.6 Проверить качество разбиения: случайным образом задать 5 точек и подать их в сеть. Преобразовать выход сети с помощью функции $vec2ind$ и занести в отчет номера кластеров. Занести в отчет график, на котором отобразить множество точек, полученные центры кластеров ($net.IW_{1,1}$) и 5 дополнительных точек. На графике исходное множество, центры кластеров и дополнительные точки должны быть изображены разным цветом, также должна быть отображена сетка.

2. Построить и обучить карту Кохонена размера 2x4 с гексагональной сеткой, которая будет содержать координаты центров кластеров.

2.1 Сформировать множество случайных точек, которые изначально сгруппированы в 8 кластеров. Использовать функцию $nngenc(X, clusters, points, deviation)$, где $X = [0, 1.5; 0, 1.5]$ задает область, в которой находятся точки множества; $clusters = 8$ задает число кластеров; $points = 10$ задает число точек в каждом из кластеров; $deviation = 0.1$ задает среднее квадратическое отклонение от центра кластера. Не разбивать обучающее множество на подмножества.

2.2 Создать сеть с помощью функции $newsom(X, [2, 4])$.

2.3 Задать число эпох обучения $net.trainParam.epochs$ равным 150. Произвести обучение сети с помощью метода, заданного по умолчанию. Занести в отчет весовые коэффициенты первого слоя после обучения. Занести в отчет графики SOM Sample Hits, SOM Weight Positions, а также содержимое окна Neural Network Training.

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

2.5 Проверить качество разбиения: случайным образом задать 5 точек и подать их в сеть. Преобразовать выход сети с помощью функции $vec2ind$ и занести в отчет номера кластеров. Занести в отчет график, на котором отобразить множество точек, полученные центры кластеров, связи между ними ($plotsom(net.IW\{1,1\}, net.layers\{1\}.distances)$), и 5 дополнительных точек. На графике исходное множество, центры кластеров и дополнительные точки должны быть изображены разным цветом, также должна быть отображена сетка.

3. Построить и обучить карту Кохонена, которая будет находить одно из решений задачи коммивояжера.

3.1 С помощью функции $rand$ сгенерировать набор из $N = 20$ случайных точек $T = \{x_i, y_i\}$ в диапазоне $[-1.5, 1.5]$. Точки задают координаты каждого города, их последовательность — маршрут, который должен пройти через все города. В данной задаче замыкать маршрут не требуется. Отобразить маршрут, полученный при создании набора точек:

```
hold on
plot(T(1,:),T(2,:), '-V','MarkerEdgeColor','k','MarkerFaceColor','g','MarkerSize',7), grid
hold off
```

Занести график о отчет.

3.2 Не разбивать обучающее множество на подмножества. Создать сеть Кохонена, нейроны которой образуют одномерную цепочку, с помощью функции $newsom(T, N)$.

3.3 Задать число эпох обучения $net.trainParam.epochs$ равным 600. Произвести обучение сети с помощью метода, заданного по умолчанию. При обучении сети города, расположенные по соседству друг с другом, будут отображаться на нейроны, расположенные по соседству в сети Кохонена. Провести обучение несколько раз, добиться, чтобы маршрут, генерируемый сетью Кохонена, проходил через наибольшее число городов.

3.4 Отобразить координаты городов и центры кластеров, сгенерированные сетью.

```
hold on
plotsom(net.IW1,1,net.layers1.distances),
plot(T(1,:),T(2,:), 'V','MarkerEdgeColor','k', 'MarkerFaceColor','g', 'MarkerSize',7), grid
hold off
```

График занести в отчет.

4. Для обучающей выборки построить LVQ-сеть, которая будет правильно относить точки к двум классам. Классы не являются линейно разделимыми. Отобразить границы классов, которые построила сеть.

4.1 Отобразить входное множество и эталонное распределение по классам с помощью функции $plotpv$, график занести в отчет. Для отображения необходимо индекс класса -1 заменить на 0.

4.2 Не разбивать обучающее множество на подмножества. Построить вектор индексов классов с помощью функции $ind2vec$. Перед этим в векторе, содержащим распределение по классам заменить -1 на 1, а 1 на 2.

4.3 Создать сеть с помощью функции $newlvq$. Число нейронов конкурентного слоя задать равным 12. Задать скорость обучения равной 0.1. При создании сети задать процентную долю принадлежности входных векторов к классу с индексом 1 и классу с индексом 2.

4.4 Задать число эпох обучения $net.trainParam.epochs$ равным 300. Произвести обучение сети с помощью метода, заданного по умолчанию. Занести в отчет содержимое окна Performance и Neural Network Training.

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

4.6 Проверить качество обучения: произвести классификацию точек области $[-1.5, 1.5] \times [-1.5, 1.5]$. Для этого задать сетку для указанной области с шагом $h = 0.1$. Рассчитать выход сети

(sim) для всех узлов сетки. Преобразовать выходные значения с помощью функции $vec2ind$. Преобразовать индексы классов: $1 \Rightarrow 0$, $2 \Rightarrow 1$. Отобразить границы классов с помощью:

```
plotpv(xyGrid,output);
point = findobj(gca,'type','line');
set(point,'Color','g');
hold on;
plotpv(P,T)
hold off;
```

где P, T — обучающее множество, $xyGrid$ — набор узлов сетки, $output$ — выход LVQ-сети. Полученный график занести в отчет.

Варианты заданий:

Номер варианта соответствует номеру в списке группы.

№	Обучающее множество
1.	$\begin{bmatrix} 0.5 & 0.7 & 0.4 & 0.6 & -0.7 & -1.3 & 0.5 & 1.3 & -0.2 & 0.7 & -1 & -0.2 \\ 0.7 & -0.4 & -1 & -1.5 & -1.4 & 0.9 & -0.6 & -1.4 & -0.4 & 0.8 & -0.1 & 0.4 \end{bmatrix}$ $\begin{bmatrix} 1 & 1 & -1 & -1 & 1 & 1 & -1 & 1 & -1 & 1 & 1 & 1 \end{bmatrix}$
2.	$\begin{bmatrix} 0.1 & 0 & -0.9 & -1.2 & -0.3 & -0.5 & 1.2 & -1 & 0.5 & 0 & 1 \\ 0.6 & 1.4 & -1.2 & -1.4 & -0.2 & 0.7 & 0.8 & 1.4 & -1.1 & -1.3 & 0-0.1 \end{bmatrix}$ $\begin{bmatrix} -1 & -1 & 1 & 1 & 1 & 1 & -1 & 1 & 1 & 1 & -1 & 1 \end{bmatrix}$
3.	$\begin{bmatrix} 0 & 0.3 & -1.3 & 1.2 & -1.2 & -0.5 & 0.7 & -1.4 & 0.3 & 0.6 & 0.8 & 0.5 \\ 0.7 & -1.3 & 0.8 & 0.1 & 0.9 & -0.7 & -1.5 & 0.5 & 0 & 0.6 & -0.7 & 0.1 \end{bmatrix}$ $\begin{bmatrix} -1 & -1 & -1 & -1 & -1 & -1 & -1 & -1 & 1 & 1 & 1 & 1 \end{bmatrix}$
4.	$\begin{bmatrix} -0.3 & -1.2 & -1 & 0.8 & 0.8 & -0.7 & 0.5 & -0.2 & -1.4 & 0.8 & -1.2 & -1.3 \\ 0.4 & 1.3 & -0.8 & -0.1 & -0.4 & -1.4 & -0.3 & 0.3 & -0.6 & 0.5 & -1.2 & -1.5 \end{bmatrix}$ $\begin{bmatrix} 1 & 1 & -1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & -1 & -1 \end{bmatrix}$
5.	$\begin{bmatrix} -1.2 & -0.6 & -0.4 & -1 & -1.5 & 0.4 & -1.1 & 0.8 & -0.2 & 0.7 & -1.4 & 0.5 \\ 0 & 0.1 & -0.3 & -0.8 & 1.2 & 1.2 & 1.2 & 0.2 & -0.8 & -0.9 & 0.8 & 0.6 \end{bmatrix}$ $\begin{bmatrix} -1 & 1 & 1 & -1 & -1 & -1 & -1 & -1 & -1 & -1 & -1 & -1 \end{bmatrix}$

№	Обучающее множество
6.	$\begin{bmatrix} 0.5 & 0.7 & 0.4 & 0.6 & -0.7 & -1.3 & 0.5 & 1.3 & -0.2 & 0.7 & -1 & -0.2 \\ 0.7 & -0.4 & -1 & -1.5 & -1.4 & 0.9 & -0.6 & -1.4 & -0.4 & 0.8 & -0.1 & 0.4 \end{bmatrix}$ $\begin{bmatrix} -1 & -1 & 1 & 1 & -1 & -1 & 1 - 1 & -1 & -1 & -1 & -1 \end{bmatrix}$
7.	$\begin{bmatrix} 1.2 & -1.3 & -0.5 & -1.1 & -1.2 & -0.1 & 0.6 & 1.1 & 0.5 & -1.5 & 0 & 1.2 \\ 0.8 & -0.8 & 0.5 & 0.6 & 0.4 & 0.8 & 1.2 & -0.5 & -1 & 0.7 & -0.1 & 0.3 \end{bmatrix}$ $\begin{bmatrix} -1 & -1 & 1 & -1 & -1 & 1 & -1 & -1 & -1 & -1 & 1 & -1 \end{bmatrix}$
8.	$\begin{bmatrix} 0.6 & 0.2 & 1.2 & 0.9 & 0.2 & -0.3 & -1.1 & -0.3 & 0.2 & 0.5 & 0.4 & -1.3 \\ -0.5 & -1.2 & 1.1 & -0.8 & -1.5 & -0.6 & -1 & -1.3 & -0.1 & 0.5 & -1.4 & -0.6 \end{bmatrix}$ $\begin{bmatrix} -1 & 1 & -1 & -1 & 1 & -1 & -1 & -1 & 1 & -1 & 1 & -1 \end{bmatrix}$
9.	$\begin{bmatrix} 1 & -0.4 & 0.7 & -0.5 & 0.2 & -1.4 & -0.8 & 0.5 & 0.7 & 0.5 & -0.2 & -0.6 \\ -0.1 & 0.5 & 0 & -1.1 & -0.8 & 0.7 & -0.2 & -0.5 & -0.4 & 0.6 & -1.5 & -0.3 \end{bmatrix}$ $\begin{bmatrix} 1 & 1 & 1 & 1 & -1 & 1 & 1 & 1 & 1 & 1 & -1 & 1 \end{bmatrix}$
10.	$\begin{bmatrix} 0 & 0.3 & -1.3 & 1.2 & -1.2 & -0.5 & 0.7 & -1.4 & 0.3 & 0.6 & 0.8 & 0.5 \\ 0.7 & -1.3 & 0.8 & 0.1 & 0.9 & -0.7 & -1.5 & 0.5 & 0 & 0.6 & -0.7 & 0.1 \end{bmatrix}$ $\begin{bmatrix} -1 & -1 & -1 & -1 & -1 & -1 & -1 & -1 & 1 & 1 & -1 & 1 \end{bmatrix}$
11.	$\begin{bmatrix} -0.3 & -0.3 & -1.1 & -1.1 & 1.1 & 0.5 & 0 & 0.2 & -0.7 & -0.3 & 0.1 & -0.3 \\ 1.4 & 0.3 & -0.4 & 0.7 & -0.5 & -0.7 & 0.9 & -0.5 & -0.2 & -0.5 & 0.7 & -0.3 \end{bmatrix}$ $\begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & -1 & -1 & 1 & -1 \end{bmatrix}$
12.	$\begin{bmatrix} -0.4 & -1.5 & -1 & -0.4 & 0.7 & -1 & -0.8 & -0.9 & -1.3 & -1 & 0.6 & 0.4 \\ -1.1 & -0.3 & 0.6 & 1 & 0.2 & 1.3 & 1.2 & -0.4 & 0.4 & -1.4 & -0.5 & -0.4 \end{bmatrix}$ $\begin{bmatrix} 1 & 1 & -1 & 1 & 1 & -1 & -1 & -1 & 1 & 1 & 1 & 1 \end{bmatrix}$
13.	$\begin{bmatrix} -0.6 & -0.5 & 0 & 0.3 & -0.4 & 0 & 1.4 & -0.1 & -1.1 & 1.2 & 0.6 & -0.5 \\ -1 & -0.9 & 1.2 & -1.2 & -1.4 & -0.3 & 0.9 & 1.1 & -0.4 & 1.2 & 0.3 & 1.3 \end{bmatrix}$ $\begin{bmatrix} 1 & 1 & 1 & 1 & 1 & -1 & 1 & 1 & -1 & 1 & -1 & 1 \end{bmatrix}$
14.	$\begin{bmatrix} 0.2 & 1.3 & 0.6 & -1.5 & 0.4 & -0.4 & 0.9 & -0.4 & 1.1 & 0.5 & 0.4 & -0.3 \\ 1.1 & 0.1 & 0.2 & -0.2 & 0 & 1.3 & 1 & 0.2 & 1.3 & -0.9 & -1.3 & 0.5 \end{bmatrix}$ $\begin{bmatrix} 1 & 1 & -1 & 1 & -1 & 1 & -1 & 1 & -1 & 1 & 1 & 1 \end{bmatrix}$

№	Обучающее множество
15.	$\begin{bmatrix} 1.4 & 1.4 & 0 & -0.3 & -1.3 & -1.4 & 0.9 & 0.3 & 1.1 & -1 & 1.1 & 0 \\ 0.2 & 0.1 & -0.6 & -0.1 & 1.1 & -0.2 & -0.4 & 0.9 & 1.2 & -0.8 & 0.2 & 0.3 \end{bmatrix}$ $\begin{bmatrix} 1 & 1 & 1 & -1 & 1 & 1 & 1 & -1 & 1 & 1 & 1 & -1 \end{bmatrix}$
16.	$\begin{bmatrix} -1.3 & -0.8 & -0.3 & -0.1 & -0.8 & -1.3 & -1.5 & -1.5 & -1.1 & -1 & 0.2 & 1.3 \\ -0.3 & -0.7 & -1.2 & 0.6 & 0.8 & -0.4 & -0.9 & -1 & -0.7 & -1.1 & 1.2 & -0.9 \end{bmatrix}$ $\begin{bmatrix} -1 & 1 & -1 & -1 & -1 & -1 & -1 & -1 & 1 & 1 & -1 & -1 \end{bmatrix}$
17.	$\begin{bmatrix} -1.3 & 0 & 0.6 & 0 & 1 & -0.6 & 0.7 & -1.2 & 0 & 0.6 & -0.7 & 1.2 \\ -0.6 & -1.4 & 0.1 & 0.9 & 0.8 & -0.2 & -1.2 & -0.7 & 1.4 & -0.6 & 1 & 0.4 \end{bmatrix}$ $\begin{bmatrix} -1 & 1 & 1 & 1 & 1 & -1 & 1 & -1 & 1 & 1 & 1 & 1 \end{bmatrix}$
18.	$\begin{bmatrix} 0.6 & 0.7 & 1.3 & 1.3 & -1.4 & 0.2 & 1.2 & -0.3 & 0.6 & 0.9 & -0.1 & -0.4 \\ 0.1 & 1.4 & 0.1 & -1.2 & -0.6 & 0 & 0.1 & 0.1 & -1.5 & -1.1 & -0.8 & 0.4 \end{bmatrix}$ $\begin{bmatrix} 1 & -1 & -1 & 1 & -1 & -1 & -1 & -1 & 1 & 1 & -1 & -1 \end{bmatrix}$
19.	$\begin{bmatrix} 0.3 & -1 & -0.5 & -1.1 & 0.3 & 0.9 & 1.3 & 0.7 & 0.9 & -0.2 & 1.2 & 1.1 \\ 1.1 & 0.7 & -0.3 & 0.9 & 0.7 & -1.3 & -0.1 & 0.7 & -1.1 & 0.3 & 1 & 0.2 \end{bmatrix}$ $\begin{bmatrix} -1 & 1 & 1 & 1 & -1 & 1 & 1 & 1 & 1 & -1 & 1 & 1 \end{bmatrix}$
20.	$\begin{bmatrix} 0.8 & 0.7 & -0.8 & -0.9 & -0.7 & -1.4 & -1.1 & -1 & 1.4 & -1.4 & -0.4 & -0.9 \\ -0.4 & 1.1 & -1.2 & -0.5 & 1.2 & 0.2 & 1 & 0 & -0.5 & -0.9 & -0.5 & 1.3 \end{bmatrix}$ $\begin{bmatrix} 1 & 1 & 1 & -1 & 1 & -1 & 1 & -1 & 1 & -1 & 1 & 1 \end{bmatrix}$
21.	$\begin{bmatrix} -1.2 & 0.5 & 1.4 & 0.7 & -0.3 & -0.1 & 0.7 & -1.3 & -0.4 & -1 & 1.2 & 1.4 \\ -0.8 & 1 & -0.9 & 0.2 & 0 & 0.4 & -0.6 & 0.9 & 0.8 & -1.2 & 1.1 & 1 \end{bmatrix}$ $\begin{bmatrix} -1 & 1 & -1 & -1 & 1 & 1 & -1 & -1 & 1 & -1 & -1 & -1 \end{bmatrix}$
22.	$\begin{bmatrix} 1.2 & 0.8 & -1 & 0.3 & 0.4 & 0.7 & 0.3 & 1.2 & 0 & -0.4 & -0.8 & 0.6 \\ 1.1 & 1.2 & 0 & 1.2 & -0.4 & 0.9 & -1.3 & -1.4 & -1.2 & 0.9 & 1.1 & -0.4 \end{bmatrix}$ $\begin{bmatrix} 1 & 1 & -1 & 1 & -1 & 1 & 1 & 1 & 1 & 1 & 1 & -1 \end{bmatrix}$

№	Обучающее множество
23.	$\begin{bmatrix} -1 & 0.6 & -1.2 & 0.2 & 0 & 0.5 & 1.1 & -0.5 & 0.1 & -0.6 & 0.8 & -0.5 \\ -1.5 & 1.1 & -1.4 & 0.3 & -1.5 & 1.3 & -1.2 & -0.8 & 0.3 & 0.8 & 0.8 & -1.4 \end{bmatrix}$ $\begin{bmatrix} 1 & 1 & 1 & -1 & -1 & 1 & 1 & 1 & -1 & 1 & 1 & 1 \end{bmatrix}$
24.	$\begin{bmatrix} 0 & -1.2 & -0.5 & 0.2 & -1 & 1.2 & -0.7 & 1.1 & -0.7 & 0.2 & -1.5 & 0 \\ 0.8 & 0.3 & -0.5 & 1 & 0.5 & -1 & -0.1 & 0 & 0.1 & -0.3 & 0.6 & -0.4 \end{bmatrix}$ $\begin{bmatrix} 1 & 1 & -1 & 1 & 1 & 1 & 1 & 1 & 1 & -1 & 1 & -1 \end{bmatrix}$
25.	$\begin{bmatrix} -1.3 & -0.8 & -0.7 & 1.4 & 0.6 & 0.5 & -0.4 & -0.5 & -0.4 & 0 & -0.9 & 0.2 \\ 0.6 & -0.4 & 0.9 & 0.4 & 1.2 & 0.2 & 0.4 & -0.3 & -0.1 & 1.2 & -0.5 & -0.1 \end{bmatrix}$ $\begin{bmatrix} 1 & -1 & 1 & 1 & 1 & 1 & -1 & -1 & -1 & 1 & -1 & 1 \end{bmatrix}$
26.	$\begin{bmatrix} 0.9 & 0.6 & 0.6 & 0 & 0.8 & -0.8 & 0.3 & -1.2 & -0.7 & -0.3 & 1.3 & -1.4 \\ 0.8 & -0.1 & 1.3 & -0.6 & -0.8 & -0.2 & 0.3 & -1.2 & -0.7 & 0.9 & -0.4 & 0.2 \end{bmatrix}$ $\begin{bmatrix} -1 & 1 & -1 & 1 & 1 & -1 & -1 & -1 & -1 & -1 & -1 & -1 \end{bmatrix}$
27.	$\begin{bmatrix} -0.7 & -1.5 & -0.3 & 1.3 & -0.7 & 0.2 & -1.3 & -0.7 & 0.6 & -0.5 & 1.3 & 0.8 \\ 0 & 1.1 & -0.6 & -0.2 & -1.3 & -1.1 & 0.2 & -0.5 & 1 & 1.3 & -0.9 & 0.3 \end{bmatrix}$ $\begin{bmatrix} -1 & 1 & -1 & 1 & -1 & -1 & 1 & -1 & 1 & 1 & 1 & 1 \end{bmatrix}$
28.	$\begin{bmatrix} -0.8 & -1 & 0.1 & -0.4 & 1 & -0.3 & -0.9 & 0.5 & -0.5 & -1 & -0.1 & 0.8 \\ 1.1 & -1.2 & -0.6 & 0.8 & 0.5 & 0.4 & 0.3 & 0.3 & -0.5 & 0.8 & -0.5 & -0.8 \end{bmatrix}$ $\begin{bmatrix} -1 & -1 & 1 & -1 & -1 & -1 & -1 & -1 & 1 & -1 & 1 & -1 \end{bmatrix}$
29.	$\begin{bmatrix} 0 & 1.4 & -1.2 & 0.2 & -0.3 & -1.3 & 0.4 & 0.6 & -1.3 & -0.8 & 0.6 & 0.1 \\ 1.2 & 0.2 & 1.2 & 1.1 & 0.3 & 1.2 & -1.2 & -0.4 & -0.9 & -0.9 & 0.7 & 0.1 \end{bmatrix}$ $\begin{bmatrix} -1 & 1 & 1 & -1 & 1 & 1 & 1 & 1 & 1 & 1 & -1 & 1 \end{bmatrix}$
30.	$\begin{bmatrix} 1.2 & -0.5 & -0.1 & 0 & 0.6 & 0.4 & -0.9 & -1.1 & -0.9 & -1.5 & -1.5 & -0.7 \\ -1.3 & 0.2 & 0.9 & -0.9 & -1.1 & 0.4 & -1 & -1.1 & 1.3 & 1.3 & 1.4 & 0 \end{bmatrix}$ $\begin{bmatrix} 1 & -1 & -1 & 1 & 1 & -1 & 1 & 1 & 1 & 1 & 1 & -1 \end{bmatrix}$

№	Обучающее множество

Литература

1. *Beale M., Hagan M., Demuth H.* Neural Network Toolbox User's guide R2011b. The MathWorks, 2011. –pp. 6-3–6-48.

2. *Медведев В. С., Потемкин В. Г.* Нейронные сети. MATLAB 6/Под общ. ред. к. т. н. В. Г. Потемкина – М.: ДИАЛОГ-МИФИ, 2006. – с. 147–174.

3. *Hagan M., Demuth H.* Neural Network Design. 1996. – Chapter 13-14. – 76 pp.

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 $\{x_i, y_i\},\ i = 1, .., N$. 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 Отразить 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.

$$t = 0 : h : 2\pi$$
$$x = f(t)$$
$$y = g(t)$$

Константы a и b задают большую и малую полуоси эллипса. Параметры преобразования прямоугольной системы координат на плоскости: угол поворота (α) 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 $\{x_i, y_i\}$, $i = 1, .., N$. 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.	$r = \dfrac{2\sin\varphi}{\varphi}, \quad \varphi \in [0.01, \pi]$
11.	$r = 5, \quad \varphi \in [0.01, \dfrac{11\pi}{6}]$
12.	$r = \sqrt{\varphi}$
13.	$r = \dfrac{\pi}{\cos\dfrac{\varphi}{3}}, \quad \varphi \in [0.01, \pi]$
14.	$r = 2\varphi^2$
15.	$r = -3\varphi^2 + 1$
16.	$r = \left(\dfrac{1}{2}\right)^{\varphi}$
17.	$r = 2\cos\varphi, \quad \varphi \in [0.01, \dfrac{11\pi}{6}]$
18.	$r = \dfrac{2(\pi - 2\varphi)}{\pi\cos\varphi}, \quad \varphi \in [0.01, \pi]$
19.	$r = \operatorname{ctg}\varphi, \quad \varphi \in [0.01, \pi]$
20.	$r = \dfrac{8\sin\varphi}{\cos^2\varphi}, \quad \varphi \in [0.01, \dfrac{\pi}{4}]$

№	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 $\{x_i, y_i\},\ i = 1, .., N$. 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 Отразить 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. Харитонова)

2. *Хайкин С.* Нейронные стеи: 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.

Лабораторная работа № 8.
Динамические сети

Целью работы является исследование свойств некоторых динамических нейронных сетей, алгоритмов обучения, а также применение сетей в задачах аппроксимации функций и распознавания динамических образов.

Основные этапы работы:

1. Использовать сеть прямого распространения с запаздыванием для предсказания значений временного ряда и выполнения многошагового прогноза.

2. Использовать сеть сеть прямого распространения с распределенным запаздыванием для распознавания динамических образов.

3. Использовать нелинейную авторегрессионную сеть с внешними входами для аппроксимации траектории динамической системы и выполнения многошагового прогноза.

Сценарий работы:

Этап 1

1. Построить и обучить сеть прямого распространения с запаздыванием (Focused Time-Delay Neural Network, FTDNN), которая будет аппроксимировать последовательность чисел Вольфа, а также выполнить многошаговый прогноз. Сеть должна выполнять отображение вида:

$$\hat{y}(n+1) = F[y(n), \ldots, y(n-D)]$$

где D задает глубину погружения временного ряда ($delays$).

1.1 Число Вольфа — один характерных из показателей солнечной активности. Для заданного момента времени задает количество пятен на Солнце. Для аппроксимации использовать среднемесячные значения чисел Вольфа. Данные рекомендуется загрузить по адресу `ftp://ftp.ngdc.noaa.gov/STP/SOLAR_DATA/SUNSPOT_NUMBERS/` или `http://sidc.oma.be/sunspot-data/`.

1.2 Импортировать загруженные данные, выбрав пункт меню *File > Import Data*. Рекомендуется перед импортированием удалить незаполненные строки в начале и в конце файла.

1.3 В соответствии с вариантом выделить часть временной последовательности. Начало последовательности определяется вариантом задания. Выполнить сглаживание траектории с помощью усредняющего фильтра ($smooth$) с шириной окна равной 12. Преобразовать входную последовательность в матрицу-строку.

1.4 Глубина погружения временного ряда $D = 5$. Выделить часть временной последовательности для инициализации задержек (Pi). Сформировать обучающее, контрольное, тестовое подмножества: задать число временных отсчетов равным 500, 100, и 50 соответственно.

1.5 Выделенные подмножества объединить в обучающую выборку последовательно. При формировании эталонной выборки учесть, что сеть по значению на текущем шаге должна предсказывать значения на следующем, т.е. выполнять одношаговый прогноз.

1.6 Преобразовать обучающее множество с помощью функции $con2seq$.

1.7 Создать сеть с помощью функции $timedelaynet$. Число нейронов скрытого слоя задать равным 8. Задать задержки от 1 до $D = 5$. Для обучения сети использовать метод Левенберга-Марквардта ($trainlm$). Для скрытого и выходного слоев использовать активационные функции $tansig$ и $purelin$ соответственно.

1.8 При обучении сети использовать разделение обучающего множества на подмножества с помощью функции $divideind$. Индексы задать задать в соответствии с тем, что подмножества выделяются последовательно.

1.9 Сконфигурировать сеть ($configure$) под обучающее множество.

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

1.11 Задать параметры обучения: число эпох обучения ($net.trainParam.epochs$) и число эпох, в течение которых может расти ошибка на контрольном подмножестве ($net.trainParam.max\_fail$), равными 600, предельное значение критерия обучения ($net.trainParam.goal$) равным 10^{-5}.

1.12 Произвести обучение сети, инициализировав соответствующие линии задержек. Если необходимо, то произвести обучение несколько раз. Если результаты неудовлетворительные, то увеличить число нейронов сети. Занести в отчет содержимое окон Performance и Neural Network Training.

1.13 Отразить структуру сети и проведенное обучение, заполнив таблицу 1.

1.14 Рассчитать выход сети (sim) для обучающего подмножества, инициализировав соответствующие линии задержек (Pi). Сравнить выход сети с соответствующим эталонным подмножеством: рассчитать показатели качества обучения и заполнить таблицу 2. Отобразить на графике эталонные значения и предсказанные сетью, а также ошибку обучения (на отдельном графике). Графики занести в отчет.

1.15 Выполнить многошаговый прогноз: рассчитать выход сети (sim) для тестового подмножества. Сформировать отдельное подмножество для инициализации задержек, выделив последние D элементов контрольного подмножества. Сравнить выход сети с соответствующим эталонным подмножеством: рассчитать показатели качества обучения и заполнить таблицу 2. Отобразить на графике эталонные значения и предсказанные сетью, а также ошибку обучения (на отдельном графике). Графики занести в отчет.

Варианты заданий:

Номер варианта соответствует номеру студента в списке группы. Для каждого варианта задается месяц и год, которые задают начало среднемесячной последовательности чисел Вольфа.

№	Начало временной последовательности
1.	07/1777
2.	10/1860
3.	04/1800
4.	10/1848
5.	05/1873
6.	05/1811
7.	04/1761

№	ММ/ГГГГ
8.	04/1830
9.	03/1902
10.	12/1804
11.	05/1784
12.	11/1879
13.	04/1761
14.	07/1816
15.	10/1889
16.	10/1787
17.	11/1750
18.	08/1778
19.	10/1868
20.	03/1777
21.	05/1874
22.	02/1859
23.	03/1816
24.	07/1752
25.	03/1800
26.	05/1814

№	ММ/ГГГГ
27.	02/1847
28.	05/1913
29.	03/1899
30.	01/1850

Этап 2

2. Построить и обучить сеть прямого распространения с распределенным запаздыванием (Distributed Time-Delay Neural Network, TDNN), которая будет выполнять распознавание динамического образа. Проверить качество распознавания.

2.1 Обучающее множество взять из лабораторной работы №5. Входная последовательность обучающего множества состоит из комбинации основного сигнала (p_1) и сигнала, подлежащего распознаванию (p_2). Каждому значению основного сигнала соответствует -1 целевого выхода, каждому значению сигнала p_2 соответствует 1 целевого выхода.

$$p_1(k) = \sin(4\pi k), \quad t_1(k) = -1, \quad k \in [0, 1] \text{ с шагом } h = 0.025$$
$$p_2(k) = g(k), \quad t_2(k) = 1, \quad k \in [a_2, b_2] \text{ с шагом } h = 0.025$$

Функция $g(k)$ определяется вариантом задания. Длительность основного сигнала задается набором чисел $R = \{r_1, r_2, r_3\}$. Значения R также определяются вариантом задания. Входное множество формируется по формуле

$$P = [repmat(p_1, 1, r_1), p_2, repmat(p_1, 1, r_2), p_2, repmat(p_1, 1, r_3), p_2]$$
$$T = [repmat(t_1, 1, r_1), t_2, repmat(t_1, 1, r_2), t_2, repmat(t_1, 1, r_3), t_2]$$

Преобразовать обучающее множество с помощью функции $con2seq$.

2.2 Создать сеть с помощью функции $distdelaynet$. Задать задержки $[0 : 4]$ для входного и скрытого слоев. Число нейронов скрытого слоя задать равным 8. Для обучения сети использовать одношаговый метод секущих ($trainoss$). Для скрытого и выходного слоев использовать $tansig$ в качестве активационной функции ($net.layers\{i\}.transferFcn$). При обучении сети не использовать разделение обучающего множества на подмножества ($net.divideFcn = ''$).

2.3 Сконфигурировать сеть ($configure$) под обучающее множество.

2.4 С помощью функции $preparets$ сформировать массивы ячеек для функции обучения, содержащие обучающее множество и значения для инициализации задержек скрытого и выходного слоев (P, T, Pi, Ai соответственно). Если при выполнении заданий используется версия MATLAB, которая не поддерживает эту функцию, то обучать и выполнять расчет выходов сети без инициализации задержек.

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

2.6 Произвести обучение сети. Если необходимо, то произвести обучение несколько раз. Если результаты неудовлетворительные, то увеличить число нейронов сети. Занести в отчет содержимое окон Performance и Neural Network Training.

2.7 Отразить структуру сети и проведенное обучение, заполнив таблицу 1.

2.8 Рассчитать выход сети (sim) для обучающего множества, инициализировав соответствующие линии задержек. Отобразить на графике эталонные значения и предсказанные сетью. С помощью функции $legend$ подписать кривые.

2.9 Преобразовать значения по правилу

$$o_{ij} = \begin{cases} 1, & a_{ij} \geqslant 0; \\ -1, & a_{ij} < 0; \end{cases}$$

Сравнить выход сети с эталонными значениями. Занести в отчет процент правильно классифицированных точек.

2.10 Для проверки качества распознавания сформировать новое обучающее множество, изменив одно из значений $R = \{r_1, r_2, r_3\}$. Рассчитать выходы сети для измененной входной последовательности.

2.11 Рассчитать выход сети (sim) для обучающего множества, инициализировав соответствующие линии задержек. Отобразить на графике эталонные значения и предсказанные сетью. С помощью функции $legend$ подписать кривые.

2.12 Преобразовать значения по правилу. Сравнить выход сети с эталонными значениями. Занести в отчет процент правильно классифицированных точек.

Этап 3

3. Построить и обучить нелинейную авторегрессионную сеть с внешними входами (Non-linear AutoRegressive network with eXogeneous inputs, NARX), которая будет выполнять аппроксимацию траектории динамической системы, также выполнить многошаговый прогноз значений системы. Сеть должна выполнять отображение вида:

$$\hat{y}(n+1) = F[y(n), \ldots, y(n - D_y), u(n), \ldots, u(n - D_u)]$$

где $y(n)$ — значение выходного сигнала для текущего момента времени, $u(n)$ — значение входного управляющего сигнала для текущего момента времени, D_y, D_u — глубина погружения временного ряда ($delays$) для $y(n)$ и $u(n)$.

3.1 Построить обучающее множество. Динамическая система задается разностным уравнением вида

$$u(k) = f(k), \quad k \in [0, 10] \text{ с шагом } h = 0.01$$
$$y(0) = 0$$
$$y(k+1) = \frac{y(k)}{1 + y^2(k)} + u^3(k)$$

Входная последовательность формируется из входного управляющего сигнала $u(k)$ и выходного сигнала $y(k)$. Функция $f(k)$ определяется вариантом задания. Последовательность целевых выходов задает выходной сигнал $y(k)$.

3.2 Глубина погружения временного ряда $D = 3$. Выделить часть временной последовательности для инициализации задержек (Pi). Сформировать обучающее, контрольное, тестовое подмножества: задать число временных отсчетов равным 700, 200, и 97 соответственно.

3.3 Выделенные подмножества объединить в обучающую выборку последовательно. При формировании эталонной выборки учесть, что сеть по значению на текущем шаге должна предсказывать значения на следующем, т.е. выполнять одношаговый прогноз.

3.4 Преобразовать обучающее множество с помощью функции $con2seq$.

3.5 Создать NARX сеть с последовательно-параллельной архитектурой с помощью функции $narxnet$. Задать задержки $[1 : 3]$ для каждого из входов сети. Число нейронов скрытого слоя задать равным 10. Для обучения сети использовать метод Левенберга-Марквардта. Для скрытого и выходного слоев использовать активационные функции $tansig$ и $purelin$ соответственно.

3.6 При обучении сети использовать разделение обучающего множества на подмножества с помощью функции $divideind$. Индексы задать в соответствии с тем, что подмножества выделяются последовательно.

3.7 Сконфигурировать сеть ($configure$) под обучающее множество. При этом необходимо учитывать, что сеть имеет 2 входа.

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

3.9 Задать параметры обучения: число эпох обучения ($net.trainParam.epochs$) и число эпох, в течение которых может расти ошибка на контрольном подмножестве ($net.trainParam.max\_fail$), равными 600, предельное значение критерия обучения ($net.trainParam.goal$) равным 10^{-8}.

3.10 Произвести обучение сети. Если необходимо, то произвести обучение несколько раз. Занести в отчет содержимое окон Performance и Neural Network Training.

3.11 Отразить структуру сети и проведенное обучение, заполнив таблицу 1.

3.12 Рассчитать выход сети (sim) для обучающего подмножества, инициализировав соответствующие линии задержек (Pi). Сравнить выход сети с соответствующим эталонным подмножеством: рассчитать показатели качества обучения и заполнить таблицу 2. Отобразить на графике эталонные значения и предсказанные сетью, а также ошибку обучения (на отдельном графике). Графики занести в отчет.

3.13 Выполнить многошаговый прогноз: рассчитать выход сети (sim) для тестового подмножества. Сформировать отдельное подмножество для инициализации задержек, выделив последние D элементов контрольного подмножества. Сравнить выход сети с соответствующим эталонным подмножеством: рассчитать показатели качества обучения и заполнить таблицу 2. Отобразить на графике эталонные значения и предсказанные сетью, а также ошибку обучения (на отдельном графике). Графики занести в отчет.

Варианты заданий:

Номер варианта соответствует номеру студента в списке группы.

№	Управляющий сигнал
1.	$u(k) = \sin(k^2)$
2.	$u(k) = \sin(-2k^2 + 7k)$
3.	$u(k) = \sin(-3k^2 + 10k - 5)$
4.	$u(k) = \sin(k^2 - 7k)$
5.	$u(k) = \sin(k^2 - 2k + 5)$
6.	$u(k) = \sin(k^2 - 6k + 3)$
7.	$u(k) = \sin(k^2 - 10k + 3)$

№	Управляющий сигнал
8.	$u(k) = \sin(k^2 - 2k + 3)$
9.	$u(k) = \sin(-2k^2 + 7k) - 0.5\sin(k)$
10.	$u(k) = \sin(k^2 - 15k + 3) - \sin(k)$
11.	$u(k) = \cos(k^2)$
12.	$u(k) = \cos(k^2 - 15k + 3) - \cos(k)$
13.	$u(k) = \frac{1}{7}\sin(k^2 - 2k + \pi)$
14.	$u(k) = \frac{1}{2}\sin(k^2 - 7k + \frac{\pi}{4})$
15.	$u(k) = \frac{1}{4}\sin(k^2 - 6k - 2\pi)$
16.	$u(k) = \cos(k^2 - 15k + 3) - \cos(k)$
17.	$u(k) = \cos(-2k^2 + 7k)$
18.	$u(k) = \sin(2k^2 - 6k + 3)$
19.	$u(k) = \cos(k^2 - 10k + 3)$
20.	$u(k) = \cos(k^2 - 2k + 3)$
21.	$u(k) = \sin(2k^2 - 6k - \pi)$
22.	$u(k) = \sin(-k^2 + 2)$
23.	$u(k) = \sin(-k^2 + 8k) - \sin(2k)$
24.	$u(k) = \sin(k^2 + 3k) + \sin(k)$
25.	$u(k) = \sin(k^2) - 6\sin(k)$
26.	$u(k) = \cos(k^2) - \cos(k)$

№	Управляющий сигнал
27.	$u(k) = \cos(k^2) - \cos^2(k)$
28.	$u(k) = \cos(-k^2 - 8k) + \cos^2(k)$
29.	$u(k) = \sin(-k^2 + k) + \sin(2k)$
30.	$u(k) = \sin(k^2) + \sin^2(k)$

Литература

1. *Beale M., Hagan M., Demuth H.* Neural Network Toolbox User's guide R2011b. The MathWorks, 2011. –pp. 3-2–3-29.

2. *Медведев В. С., Потемкин В. Г.* Нейронные сети. MATLAB 6/Под общ. ред. к. т. н. В. Г. Потемкина – М.: ДИАЛОГ-МИФИ, 2006. – с. 258–260.

3. *Осовский С.* Нейронные сети для обработки информации. – М.: Финансы и статистика, 2002. – с. 200–210.

Курсовая работа.
Динамические сети

Целью работы является исследование свойств некоторых динамических нейронных сетей, алгоритмов обучения, а также применение сетей в задачах аппроксимации функций и распознавания динамических образов.

Основные этапы работы:

1. Использовать сеть прямого распространения с запаздыванием для предсказания значений временного ряда и выполнения многошагового прогноза.

2. Использовать нелинейную авторегрессионную сеть с внешними входами для аппроксимации траектории динамической системы и выполнения многошагового прогноза.

3. Использовать сеть Элмана для распознавания динамических образов.

4. Использовать сеть сеть прямого распространения с распределенным запаздыванием для распознавания динамических образов.

5. Использовать сеть Хопфилда для распознавания статических образов.

Сценарий работы:

Этап 1

1. Построить и обучить сеть прямого распространения с запаздыванием (Focused Time-Delay Neural Network, FTDNN), которая будет аппроксимировать последовательность чисел Вольфа, а также выполнить многошаговый прогноз. Сеть должна выполнять отображение вида:

$$\hat{y}(n+1) = F[y(n), \ldots, y(n-D)]$$

где D задает глубину погружения временного ряда ($delays$).

1.1 Число Вольфа — один характерных из показателей солнечной активности. Для заданного момента времени задает количество пятен на Солнце. Для аппроксимации использовать среднемесячные значения чисел Вольфа. Данные рекомендуется загрузить по адресу
`ftp://ftp.ngdc.noaa.gov/STP/SOLAR_DATA/SUNSPOT_NUMBERS/` или
`http://sidc.oma.be/sunspot-data/`.

1.2 Импортировать загруженные данные, выбрав пункт меню *File > Import Data*. Рекомендуется перед импортированием удалить незаполненные строки в начале и в конце файла.

1.3 В соответствии с вариантом выделить часть временной последовательности. Начало последовательности определяется вариантом задания. Выполнить сглаживание траектории с помощью усредняющего фильтра ($smooth$) с шириной окна равной 12. Преобразовать входную последовательность в матрицу-строку.

1.4 Глубина погружения временного ряда $D = 5$. Выделить часть временной последовательности для инициализации задержек (Pi). Сформировать обучающее, контрольное, тестовое подмножества: задать число временных отсчетов равным 500, 100, и 50 соответственно.

1.5 Выделенные подмножества объединить в обучающую выборку последовательно. При формировании эталонной выборки учесть, что сеть по значению на текущем шаге должна предсказывать значения на следующем, т.е. выполнять одношаговый прогноз.

1.6 Преобразовать обучающее множество с помощью функции $con2seq$.

1.7 Создать сеть с помощью функции $timedelaynet$. Число нейронов скрытого слоя задать равным 8. Задать задержки от 1 до $D = 5$. Для обучения сети использовать метод Левенберга-Марквардта ($trainlm$). Для скрытого и выходного слоев использовать активационные функции $tansig$ и $purelin$ соответственно.

1.8 При обучении сети использовать разделение обучающего множества на подмножества с помощью функции $divideind$. Индексы задать задать в соответствии с тем, что подмножества выделяются последовательно.

1.9 Сконфигурировать сеть ($configure$) под обучающее множество.

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

1.11 Задать параметры обучения: число эпох обучения ($net.trainParam.epochs$) и число эпох, в течение которых может расти ошибка на контрольном подмножестве ($net.trainParam.max\_fail$), равными 600, предельное значение критерия обучения ($net.trainParam.goal$) равным 10^{-5}.

1.12 Произвести обучение сети, инициализировав соответствующие линии задержек. Если необходимо, то произвести обучение несколько раз. Если результаты неудовлетворительные, то увеличить число нейронов сети. Занести в отчет содержимое окон Performance и Neural Network Training.

1.13 Отразить структуру сети и проведенное обучение, заполнив таблицу 1.

1.14 Рассчитать выход сети (sim) для обучающего подмножества, инициализировав соответствующие линии задержек (Pi). Сравнить выход сети с соответствующим эталонным подмножеством: рассчитать показатели качества обучения и заполнить таблицу 2. Отобразить на графике эталонные значения и предсказанные сетью, а также ошибку обучения (на отдельном графике). Графики занести в отчет.

1.15 Выполнить многошаговый прогноз: рассчитать выход сети (sim) для тестового подмножества. Сформировать отдельное подмножество для инициализации задержек, выделив последние D элементов контрольного подмножества. Сравнить выход сети с соответствующим эталонным подмножеством: рассчитать показатели качества обучения и заполнить таблицу 2. Отобразить на графике эталонные значения и предсказанные сетью, а также ошибку обучения (на отдельном графике). Графики занести в отчет.

Варианты заданий:

Номер варианта соответствует номеру студента в списке группы. Для каждого варианта задается месяц и год, которые задают начало среднемесячной последовательности чисел Вольфа.

№	Начало временной последовательности
1.	07/1777
2.	10/1860
3.	04/1800
4.	10/1848
5.	05/1873
6.	05/1811

№	ММ/ГГГГ
7.	04/1761
8.	04/1830
9.	03/1902
10.	12/1804
11.	05/1784
12.	11/1879
13.	04/1761
14.	07/1816
15.	10/1889
16.	10/1787
17.	11/1750
18.	08/1778
19.	10/1868
20.	03/1777
21.	05/1874
22.	02/1859
23.	03/1816
24.	07/1752
25.	03/1800

№	ММ/ГГГГ
26.	05/1814
27.	02/1847
28.	05/1913
29.	03/1899
30.	01/1850

Этап 2

2. Построить и обучить нелинейную авторегрессионную сеть с внешними входами (Non-linear AutoRegressive network with eXogeneous inputs, NARX), которая будет выполнять аппроксимацию траектории динамической системы, также выполнить многошаговый прогноз значений системы. Сеть должна выполнять отображение вида:

$$\hat{y}(n+1) = F[y(n), \ldots, y(n-D_y), u(n), \ldots, u(n-D_u)]$$

где $y(n)$ — значение выходного сигнала для текущего момента времени, $u(n)$ — значение входного управляющего сигнала для текущего момента времени, D_y, D_u — глубина погружения временного ряда ($delays$) для $y(n)$ и $u(n)$.

2.1 Построить обучающее множество. Динамическая система задается разностным уравнением вида

$$u(k) = f(k), \quad k \in [0, 10] \text{ с шагом } h = 0.01$$
$$y(0) = 0$$
$$y(k+1) = \frac{y(k)}{1 + y^2(k)} + u^3(k)$$

Входная последовательность формируется из входного управляющего сигнала $u(k)$ и выходного сигнала $y(k)$. Функция $f(k)$ определяется вариантом задания. Последовательность целевых выходов задает выходной сигнал $y(k)$.

2.2 Глубина погружения временного ряда $D = 3$. Выделить часть временной последовательности для инициализации задержек (Pi). Сформировать обучающее, контрольное, тестовое подмножества: задать число временных отсчетов равным 700, 200, и 97 соответственно.

2.3 Выделенные подмножества объединить в обучающую выборку последовательно. При формировании эталонной выборки учесть, что сеть по значению на текущем шаге должна предсказывать значения на следующем, т.е. выполнять одношаговый прогноз.

2.4 Преобразовать обучающее множество с помощью функции $con2seq$.

2.5 Создать NARX сеть с последовательно-параллельной архитектурой с помощью функции $narxnet$. Задать задержки $[1 : 3]$ для каждого из входов сети. Число нейронов скрытого слоя задать равным 10. Для обучения сети использовать метод Левенберга-Марквардта. Для скрытого и выходного слоев использовать активационные функции $tansig$ и $purelin$ соответственно.

2.6 При обучении сети использовать разделение обучающего множества на подмножества с помощью функции $divideind$. Индексы задать в соответствии с тем, что подмножества выделяются последовательно.

2.7 Сконфигурировать сеть ($configure$) под обучающее множество. При этом необходимо учитывать, что сеть имеет 2 входа.

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

2.9 Задать параметры обучения: число эпох обучения ($net.trainParam.epochs$) и число эпох, в течение которых может расти ошибка на контрольном подмножестве ($net.trainParam.max\_fail$), равными 600, предельное значение критерия обучения ($net.trainParam.goal$) равным 10^{-8}.

2.10 Произвести обучение сети. Если необходимо, то произвести обучение несколько раз. Занести в отчет содержимое окон Performance и Neural Network Training.

2.11 Отразить структуру сети и проведенное обучение, заполнив таблицу 1.

2.12 Рассчитать выход сети (sim) для обучающего подмножества, инициализировав соответствующие линии задержек (Pi). Сравнить выход сети с соответствующим эталонным подмножеством: рассчитать показатели качества обучения и заполнить таблицу 2. Отобразить на графике эталонные значения и предсказанные сетью, а также ошибку обучения (на отдельном графике). Графики занести в отчет.

2.13 Выполнить многошаговый прогноз: рассчитать выход сети (sim) для тестового подмножества. Сформировать отдельное подмножество для инициализации задержек, выделив последние D элементов контрольного подмножества. Сравнить выход сети с соответствующим эталонным подмножеством: рассчитать показатели качества обучения и заполнить таблицу 2. Отобразить на графике эталонные значения и предсказанные сетью, а также ошибку обучения (на отдельном графике). Графики занести в отчет.

Варианты заданий:

Номер варианта соответствует номеру студента в списке группы.

№	Управляющий сигнал
1.	$u(k) = \sin(k^2)$
2.	$u(k) = \sin(-2k^2 + 7k)$
3.	$u(k) = \sin(-3k^2 + 10k - 5)$
4.	$u(k) = \sin(k^2 - 7k)$
5.	$u(k) = \sin(k^2 - 2k + 5)$
6.	$u(k) = \sin(k^2 - 6k + 3)$
7.	$u(k) = \sin(k^2 - 10k + 3)$
8.	$u(k) = \sin(k^2 - 2k + 3)$

№	Управляющий сигнал
9.	$u(k) = \sin(-2k^2 + 7k) - 0.5\sin(k)$
10.	$u(k) = \sin(k^2 - 15k + 3) - \sin(k)$
11.	$u(k) = \cos(k^2)$
12.	$u(k) = \cos(k^2 - 15k + 3) - \cos(k)$
13.	$u(k) = \frac{1}{7}\sin(k^2 - 2k + \pi)$
14.	$u(k) = \frac{1}{2}\sin(k^2 - 7k + \frac{\pi}{4})$
15.	$u(k) = \frac{1}{4}\sin(k^2 - 6k - 2\pi)$
16.	$u(k) = \cos(k^2 - 15k + 3) - \cos(k)$
17.	$u(k) = \cos(-2k^2 + 7k)$
18.	$u(k) = \sin(2k^2 - 6k + 3)$
19.	$u(k) = \cos(k^2 - 10k + 3)$
20.	$u(k) = \cos(k^2 - 2k + 3)$
21.	$u(k) = \sin(2k^2 - 6k - \pi)$
22.	$u(k) = \sin(-k^2 + 2)$
23.	$u(k) = \sin(-k^2 + 8k) - \sin(2k)$
24.	$u(k) = \sin(k^2 + 3k) + \sin(k)$
25.	$u(k) = \sin(k^2) - 6\sin(k)$
26.	$u(k) = \cos(k^2) - \cos(k)$
27.	$u(k) = \cos(k^2) - \cos^2(k)$

№	Управляющий сигнал
28.	$u(k) = \cos(-k^2 - 8k) + \cos^2(k)$
29.	$u(k) = \sin(-k^2 + k) + \sin(2k)$
30.	$u(k) = \sin(k^2) + \sin^2(k)$

Этап 3

3. Построить и обучить сеть Элмана, которая будет выполнять распознавание динамического образа. Проверить качество распознавания.

3.1 Входная последовательность обучающего множества состоит из комбинации основного сигнала (p_1) и сигнала, подлежащего распознаванию (p_2). Каждому значению основного сигнала соответствует -1 целевого выхода, каждому значению сигнала p_2 соответствует 1 целевого выхода.

$$p_1(k) = \sin(4\pi k), \quad t_1(k) = -1, \quad k \in [0, 1] \text{ с шагом } h = 0.025$$
$$p_2(k) = g(k), \quad t_2(k) = 1, \quad k \in [a_2, b_2] \text{ с шагом } h = 0.025$$

Функция $g(k)$ определяется вариантом задания. Длительность основного сигнала задается набором чисел $R = \{r_1, r_2, r_3\}$. Значения R также определяются вариантом задания. Входное множество формируется по формуле

$$P = [repmat(p_1, 1, r_1), p_2, repmat(p_1, 1, r_2), p_2, repmat(p_1, 1, r_3), p_2]$$
$$T = [repmat(t_1, 1, r_1), t_2, repmat(t_1, 1, r_2), t_2, repmat(t_1, 1, r_3), t_2]$$

Преобразовать обучающее множество с помощью функции $con2seq$. Не выделять из обучающего множества контрольное и тестовое подмножества.

3.2 Создать сеть с помощью функции $layrecnet$. Задать задержки $1 : 2$. Число нейронов скрытого слоя задать равным 8. Для обучения сети использовать одношаговый метод секущих ($trainoss$). Для скрытого и выходного слоев использовать $tansig$ в качестве активационной функции ($net.layers\{i\}.transferFcn$). Сконфигурировать сеть ($configure$) под обучающее множество.

3.3 С помощью функции $preparets$ сформировать массивы ячеек для функции обучения, содержащие обучающее множество и значения для инициализации задержек обратной связи (P, T, Pi, Ai соответственно). Если при выполнении заданий используется версия MATLAB, которая не поддерживает эту функцию, то обучать и выполнять расчет выходов сети без инициализации задержек.

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

3.5 Произвести обучение сети. Если необходимо, то произвести обучение несколько раз. Если результаты неудовлетворительные, то увеличить число нейронов сети. Занести в отчет содержимое Performance и Neural Network Training.

3.6 Отразить структуру сети и проведенное обучение, заполнив таблицу 1.

3.7 Рассчитать выход сети (sim) для обучающего подмножества. Отобразить на графике эталонные значения и предсказанные сетью. С помощью функции *legend* подписать кривые.

3.8 Преобразовать значения по правилу

$$o_{ij} = \begin{cases} 1, & a_{ij} \geqslant 0; \\ -1, & a_{ij} < 0; \end{cases}$$

Сравнить выход сети с эталонными значениями. Занести в отчет количество правильно классифицированных точек.

3.9 Для проверки качества распознавания сформировать новое обучающее множество, изменив одно из значений $R = \{r_1, r_2, r_3\}$. Рассчитать выходы сети для измененной входной последовательности.

3.10 Рассчитать выход сети (sim) для обучающего подмножества. Отобразить на графике эталонные значения и предсказанные сетью. С помощью функции *legend* подписать кривые.

3.11 Преобразовать значения по правилу. Сравнить выход сети с эталонными значениями. Занести в отчет количество правильно классифицированных точек.

Варианты заданий:

Номер варианта соответствует номеру студента в списке группы.

№	Динамический образ	Длительность $p_1(k)$
1.	$g(k) = \sin(-3k^2 + 10k - 5), \quad k \in [0.62, 3.14]$	$[0, 8, 6]$
2.	$g(k) = \cos(-2k^2 + 7k), \quad k \in [0.92, 4.07]$	$[2, 4, 7]$
3.	$g(k) = \sin(k^2 - 15k + 3) - \sin^2(k) + 0.5, \quad k \in [0.9, 3.1]$	$[3, 5, 2]$
4.	$g(k) = \sin(\sin(k)k^3 - 10), \quad k \in [1.56, 3.12]$	$[0, 1, 5]$
5.	$g(k) = 1.5\sin(\sin(k)k^2) - 0.5, \quad k \in [0.74, 3.14]$	$[3, 3, 4]$
6.	$g(k) = \sin(k^2 - 5k + 6), \quad k \in [0.67, 4.98]$	$[2, 6, 5]$
7.	$g(k) = \cos(\cos(k)k^2 - k), \quad k \in [2.16, 4.04]$	$[1, 4, 7]$
8.	$g(k) = \cos(\cos(k)k^2 + 5k), \quad k \in [2.38, 4.1]$	$[1, 3, 5]$
9.	$g(k) = \sin(\sin(k)k^2 - k), \quad k \in [1.13, 3.6]$	$[7, 0, 7]$
10.	$g(k) = \sin(-3k^2 + 5k + 10) + 0.8, \quad k \in [0.46, 3.01]$	$[0, 2, 2]$
11.	$g(k) = \cos(-\cos(k)k^2 + k), \quad k \in [2.9, 4.55]$	$[6, 7, 1]$
12.	$g(k) = \sin(\sin(k)k^2 + 5k), \quad k \in [1.86, 3.86]$	$[4, 3, 0]$
13.	$g(k) = \sin(2k^2 - 6k + 3), \quad k \in [-0.02, 2.36]$	$[2, 5, 6]$

№	G	R
14.	$g(k) = \sin(\sin(k)k^2 + 3k - 10), \quad k \in [4.45, 5.86]$	$[6, 5, 7]$
15.	$g(k) = \cos(-2k^2 + 7k), \quad k \in [0.92, 3.25]$	$[0, 4, 2]$
16.	$g(k) = \sin(\sin(k)k^2) - 0.1, \quad k \in [0.48, 2.71]$	$[7, 0, 3]$
17.	$g(k) = \sin(2.5k^2 - 5k), \quad k \in [-1.14, 1.16]$	$[5, 5, 4]$
18.	$g(k) = \cos(-5k^2 + 10k - 5), \quad k \in [0.45, 2.48]$	$[2, 1, 4]$
19.	$g(k) = \sin(-\sin(k)k^2 + k), \quad k \in [0.01, 2.77]$	$[3, 1, 3]$
20.	$g(k) = 1.5\sin(-5k^2 + 10k - 5) + 0.4, \quad k \in [0.78, 2.35]$	$[2, 2, 5]$
21.	$g(k) = \sin(\sin(k)k^2 - k), \quad k \in [1.12, 3.6]$	$[3, 0, 5]$
22.	$g(k) = \cos(-3k^2 + 5k + 10), \quad k \in [0.24, 2.7]$	$[2, 4, 4]$
23.	$g(k) = \sin(-2k^2 + 7k), \quad k \in [0.01, 2.96]$	$[3, 4, 6]$
24.	$g(k) = \cos(k^2 - 10k + 3), \quad k \in [2.84, 6.25]$	$[3, 4, 6]$
25.	$g(k) = 1.5\sin(k^2 - 6k + 3) - 0.8, \quad k \in [1.49, 3.52]$	$[5, 3, 3]$
26.	$g(k) = \sin(k^2 - 10k + 3), \quad k \in [2.5, 4.84]$	$[1, 2, 3]$
27.	$g(k) = \sin(k^2 - 2k + 3), \quad k \in [-0.05, 4.25]$	$[0, 1, 6]$
28.	$g(k) = \cos(\cos(k)k^2), \quad k \in [2.47, 4.26]$	$[7, 1, 3]$
29.	$g(k) = \sin(-2\sin(k)k^2 + 7), \quad k \in [1.41, 3.1]$	$[2, 3, 8]$
30.	$g(k) = \sin(-2k^2 + 7k) - 0.5\sin(k), \quad k \in [0.01, 2.98]$	$[4, 5, 2]$

Этап 4

4. Построить и обучить сеть прямого распространения с распределенным запаздыванием (Distributed Time-Delay Neural Network, TDNN), которая будет выполнять распознавание динамического образа. Проверить качество распознавания.

4.1 Обучающее множество взять из Этапа работы №3. Входная последовательность обучающего множества состоит из комбинации основного сигнала (p_1) и сигнала, подлежащего распознаванию (p_2). Каждому значению основного сигнала соответствует -1 целевого выхода,

каждому значению сигнала p_2 соответствует 1 целевого выхода.

$$p_1(k) = \sin(4\pi k), \quad t_1(k) = -1, \quad k \in [0, 1] \text{ с шагом } h = 0.025$$
$$p_2(k) = g(k), \quad t_2(k) = 1, \quad k \in [a_2, b_2] \text{ с шагом } h = 0.025$$

Функция $g(k)$ определяется вариантом задания. Длительность основного сигнала задается набором чисел $R = \{r_1, r_2, r_3\}$. Значения R также определяются вариантом задания. Входное множество формируется по формуле

$$P = [repmat(p_1, 1, r_1), p_2, repmat(p_1, 1, r_2), p_2, repmat(p_1, 1, r_3), p_2]$$
$$T = [repmat(t_1, 1, r_1), t_2, repmat(t_1, 1, r_2), t_2, repmat(t_1, 1, r_3), t_2]$$

Преобразовать обучающее множество с помощью функции $con2seq$.

4.2 Создать сеть с помощью функции $distdelaynet$. Задать задержки $[0 : 4]$ для входного и скрытого слоев. Число нейронов скрытого слоя задать равным 8. Для обучения сети использовать одношаговый метод секущих ($trainoss$). Для скрытого и выходного слоев использовать $tansig$ в качестве активационной функции ($net.layers\{i\}.transferFcn$). При обучении сети не использовать разделение обучающего множества на подмножества ($net.divideFcn = ''$).

4.3 Сконфигурировать сеть ($configure$) под обучающее множество.

4.4 С помощью функции $preparets$ сформировать массивы ячеек для функции обучения, содержащие обучающее множество и значения для инициализации задержек скрытого и выходного слоев (P, T, Pi, Ai соответственно). Если при выполнении заданий используется версия MATLAB, которая не поддерживает эту функцию, то обучать и выполнять расчет выходов сети без инициализации задержек.

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

4.6 Произвести обучение сети. Если необходимо, то произвести обучение несколько раз. Если результаты неудовлетворительные, то увеличить число нейронов сети. Занести в отчет содержимое окон Performance и Neural Network Training.

4.7 Отразить структуру сети и проведенное обучение, заполнив таблицу 1.

4.8 Рассчитать выход сети (sim) для обучающего множества, инициализировав соответствующие линии задержек. Отобразить на графике эталонные значения и предсказанные сетью. С помощью функции $legend$ подписать кривые.

4.9 Преобразовать значения по правилу

$$o_{ij} = \begin{cases} 1, & a_{ij} \geqslant 0; \\ -1, & a_{ij} < 0; \end{cases}$$

Сравнить выход сети с эталонными значениями. Занести в отчет процент правильно классифицированных точек.

4.10 Для проверки качества распознавания сформировать новое обучающее множество, изменив одно из значений $R = \{r_1, r_2, r_3\}$. Рассчитать выходы сети для измененной входной последовательности.

4.11 Рассчитать выход сети (sim) для обучающего множества, инициализировав соответствующие линии задержек. Отобразить на графике эталонные значения и предсказанные сетью. С помощью функции $legend$ подписать кривые.

4.12 Преобразовать значения по правилу. Сравнить выход сети с эталонными значениями. Занести в отчет процент правильно классифицированных точек.

Этап 5

5. Построить сеть Хопфилда, которая будет хранить образы из заданного набора. Эталонными образами являются двоичные изображения цифр $0, 1, 2, 3, 4, 6, 9$ (рис. 3) размером 12x10. Проверить работу сети с зашумленными образами.

5.1 Создать сеть с помощью функции $newhop$. Аттракторами построенной сети должны быть 3 образа, которые определяются вариантом задания. Каждый эталонный образ задается матрицей. Цветам точек соответствуют -1 и 1. Для синтеза сети необходимо объединить эталонные образы по формуле $T = [p1(:), p2(:), p3(:)]$.

5.2 Подать в сеть первый образ, рассчитать выход сети. Число итераций задать равным 600. Результат распознавания занести в отчет. Для этого с помощью функции $reshape(p1, 12, 10)$ преобразовать выход сети и заменить в полученной матрице значения по правилу

$$x_{ij} = \begin{cases} 2, & a_{ij} \geqslant 0; \\ 1, & a_{ij} < 0; \end{cases}$$

Для отображения результата распознавания использовать вызов следующих функций:

```
map = [1, 1, 1; 0, 0, 0];
image(X); colormap(map)
axis off
axis image
```

5.3 Произвести зашумление второго образа на 20%, полученный образ занести в отчет. Рассчитать выход сети. Результат распознавания занести в отчет.

Зашумление произвести следующим образом: для каждой точки изображения изменить цвет по правилу

if $r_{ij} < M$ *then инвертировать цвет точки*

где M — степень зашумления, r —реализация случайной величины, распределенной по равномерному закону (функция *rand*).

5.4 Произвести зашумление третьего образа на 30%, полученный образ занести в отчет. Рассчитать выход сети. Число итераций задать равным 600. Если необходимо, то произвести обучение несколько раз. Если результаты распознавания неудовлетворительные, то увеличить число итераций. Результат распознавания занести в отчет.

Варианты заданий:

Номер варианта соответствует номеру в списке группы.

№	Цифры
1.	$[1, 0, 6]$
2.	$[1, 6, 4]$
3.	$[4, 3, 2]$
4.	$[6, 3, 9]$
5.	$[9, 0, 4]$

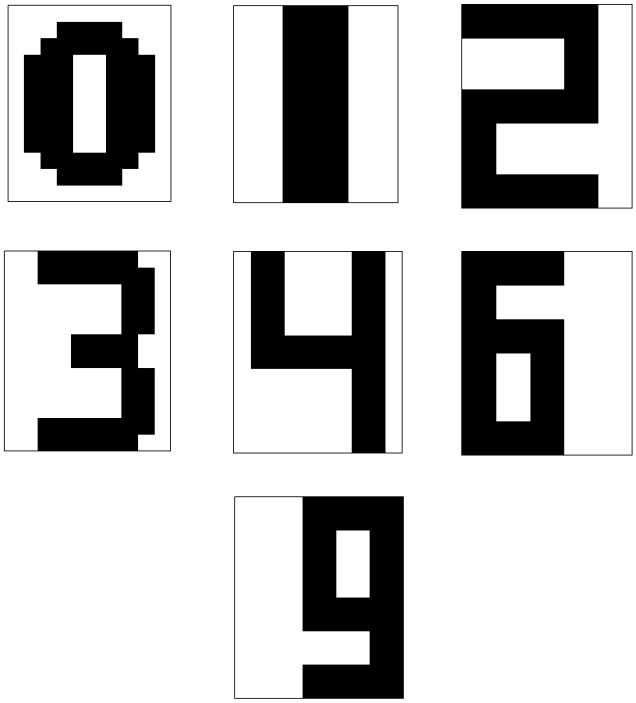

Рис. 3. Эталонные образы

№	Цифры
6.	$[9, 2, 3]$
7.	$[3, 1, 0]$
8.	$[2, 4, 1]$
9.	$[0, 9, 2]$
10.	$[4, 3, 0]$
11.	$[9, 6, 1]$
12.	$[2, 1, 6]$
13.	$[0, 1, 4]$
14.	$[6, 2, 3]$
15.	$[4, 2, 9]$

№	Цифры
16.	$[3, 6, 0]$
17.	$[0, 1, 3]$
18.	$[1, 4, 2]$
19.	$[9, 3, 0]$
20.	$[6, 2, 9]$
21.	$[3, 0, 4]$
22.	$[9, 1, 3]$
23.	$[6, 9, 2]$
24.	$[6, 1, 0]$
25.	$[0, 2, 3]$
26.	$[4, 0, 6]$
27.	$[3, 4, 2]$
28.	$[2, 1, 6]$
29.	$[9, 3, 2]$
30.	$[1, 4, 0]$

Литература

1. *Beale M., Hagan M., Demuth H.* Neural Network Toolbox User's guide R2011b. The MathWorks, 2011. –pp. 3-2–3-31, 9-34–9-41.

2. *Круглов В. В., Дли М. И., Голунов Р. Ю.* Нечеткая логика и искусственные нейронные сети. – М.: Физматлит, 2001. – с. 90–94.

3. *Медведев В. С., Потемкин В. Г.* Нейронные сети. MATLAB 6/Под общ. ред. к. т. н. В. Г. Потемкина – М.: ДИАЛОГ-МИФИ, 2006. – с. 175–188, 258–260.

4. *Осовский С.* Нейронные сети для обработки информации. – М.: Финансы и статистика, 2002. – с. 200–219.

Таблицы № 1 и № 2

Таблица 1. Информация о структуре сети и проведенном обучении

Функция создания сети	*Ex: newp*
Входной слой	*размерность входного вектора*
Скрытый слой	*число нейронов*
Выходной слой	*размерность выходного вектора*
Активационные функции	*указать для каждого слоя*
Динамика	*обратные связи, величины TDL*
Функция разделения обучающего множества	*Ex: divideind*
Число примеров в подмножествах	*Ex: 170-20-10*
Метод обучения	*Ex: trainlm*
Параметры обучения	*Ex: lr = 0.01*
Метод инициализации сети	*Ex: rands*
Критерий окончания обучения	*Ex: epochs = 1000, goal* = 10^{-8}
Причина окончания обучения	*Ex: достигнуто предельное значение критерия обучения*
Число эпох обучения	*Ex: 723*

Таблица 2. Информация о качестве работы сети на заданном наборе данных

R квадрат	
MSE	
RMSE	
Относительная СКО, %	
MAE	
min absolute error	
max absolute error	
MAPE, %	
Доля с ошибкой менее 5%, %	
Доля с ошибкой от 5% до 10%, %	
Доля с ошибкой от 10% до 20%, %	
Доля с ошибкой от 20% до 30%, %	
Доля с ошибкой более 30%, %	

Показатели качества обучения

Коэффициент множественной детерминации (R^2, R квадрат) — статистический индикатор, применяемый при анализе методом множественной регрессии. Позволяет оценить точность

модели по отношению к тривиальной модели, т. е. среднему значению выхода по всем примерам. При хорошем совпадении предсказаний будет стремиться к 1. При очень плохом коэффициент будет равен 0. Если предсказания нейросетевой модели хуже, чем предсказания тривиальной модели, то коэффициент будет меньше 0. R квадрат вычисляется по формуле:

$$R^2 = 1 - \frac{SSE}{SSyy},$$
$$SSE = \sum (y - \hat{y})^2,$$
$$SSyy = \sum (y - \bar{y})^2.$$

где y — истинное значение, $\hat{y}$ — предсказанное значение, $\bar{y} = E[y] = mean(y)$ — среднее значение для y. SSE — сумма квадратов ошибок.

Средний квадрат ошибки (mean squared error, MSE) вычисляется по формуле

$$E[(y - \hat{y})^2].$$

Средняя квадратичная ошибка (СКО, RMSE) вычисляется по формуле $\sqrt{MSE}$.

Относительная СКО к диапазону вычисляется по формуле

$$\frac{RMSE}{\max(y) - \min(y)} \cdot 100\%.$$

Средняя абсолютная ошибка (mean absolute error, MAE) вычисляется по формуле

$$E(|y - \hat{y}|).$$

Минимальная абсолютная ошибка (minimal absolute error) вычисляется по формуле

$$\min(|y - \hat{y}|).$$

Максимальная абсолютная ошибка вычисляется по формуле

$$\max(|y - \hat{y}|).$$

Средняя относительная ошибка (mean absolute percentage error, MAPE) вычисляется по формуле

$$E\left(\frac{|y - \hat{y}|}{y}\right) \cdot 100\%.$$

Для вычисления показателей долей (***доля с ошибкой менее/более N***) необходимо вычислить относительную ошибку в каждой точке, т.е. последовательность

$$\left\{\frac{|y - \hat{y}|}{y} \cdot 100\%\right\}.$$

Далее нужно пересчитать значения, лежащие в соответствующих диапазонах.

Рекомендации по оформлению лабораторных работ

Общие требования к оформлению

- Лабораторная работа распечатывается на бумаге формата А4 (210 мм × 297 мм).
- Поля: верхнее — 20 мм, нижнее — 20 мм, левое — 30 мм, правое — 15 мм.
- Шрифт — Times New Roman, междустрочный интервал — одинарный, основной размер шрифта – 14 пт., отступы первой строки абзаца — 1,25 см, выравнивание — по ширине.
- Номер страницы указывать снизу по центру страницы.
- Основные разделы выделять полужирным шрифтом.

Разделы отчета

Цель работы: см. задание

Основные этапы работы:

- этап № 1 см. задание
- этап № 2
- этап № 3

Оборудование:
Указать характеристики процессора и объем оперативной памяти.

Программное обеспечение:
Указать версию и разрядность MATLAB.

Сценарий выполнения работы:
В разделе содержатся числовые данные и графики. Размер каждого изображения не должен превышать одной трети листа. Изображения сопровождаются краткими подрисуночными подписями.

Шаблон для отображения графиков:

```
hold on
plot(t, Target, '-r', t, netOutput, '-b'),
xlabel('t'), ylabel(''),
title(''),
legend('Target','Net output'), grid
hold off
close;
```

Код программы:
Вставить код программы, выполняющей этапы задания лабораторной работы. Рекомендуется сделать размер шрифта равным 10 пт. и убрать пустые строки.

Выводы:
Краткие выводы о содержании лабораторной работы.

Оформления титульного листа отчета

МОСКОВСКИЙ АВИАЦИОННЫЙ ИНСТИТУТ
(НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ)

Кафедра вычислительной математики и программирования

Лабораторная работа № 1
по спецкурсу «Нейроинформатика»

Персептроны. Процедура обучения Розенблатта

Выполнил: Иванов И. И.
Группа: 08-40_, вариант 1
Преподаватели: Тюменцев Ю. В.
Козлов Д.С.

Москва, 20__