

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 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:

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

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

1.1 Число Вольфа — один из характерных показателей солнечной активности. Для заданного момента времени задает количество пятен на Солнце. Для аппроксимации использовать средне-месячные 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 (P_i). 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 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 Отобразить 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

№	MM/TTTT
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:

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

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

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

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 Отобразить 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

$$P = [\text{repmat}(p_1, 1, r_1), p_2, \text{repmat}(p_1, 1, r_2), p_2, \text{repmat}(p_1, 1, r_3), p_2]$$

$$T = [\text{repmat}(t_1, 1, r_1), t_2, \text{repmat}(t_1, 1, r_2), t_2, \text{repmat}(t_1, 1, r_3), t_2]$$

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 текущих (*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, P_i, A_i 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 Отобразить 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

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

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.

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

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

$$\begin{aligned} P &= [\text{repmat}(p_1, 1, r_1), p_2, \text{repmat}(p_1, 1, r_2), p_2, \text{repmat}(p_1, 1, r_3), p_2] \\ T &= [\text{repmat}(t_1, 1, r_1), t_2, \text{repmat}(t_1, 1, r_2), t_2, \text{repmat}(t_1, 1, r_3), t_2] \end{aligned}$$

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, P_i, A_i 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 Отобразить 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

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

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

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

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]

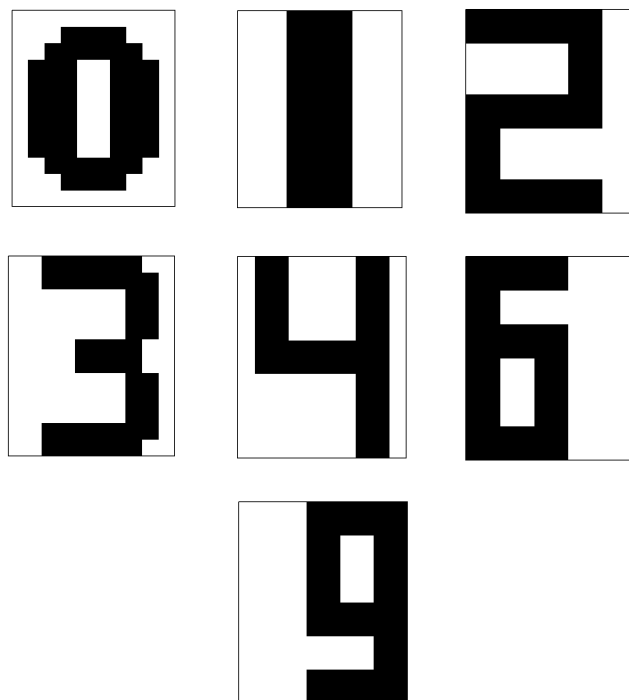


Рис. 3. Эталонные образы

№	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	<i>Ex: newp</i>
Входной слой	<i>размерность входного вектора</i>
Скрытый слой	<i>число нейронов</i>
Выходной слой	<i>размерность выходного вектора</i>
Активационные функции	<i>указать для каждого слоя</i>
Динамика	<i>обратные связи, величины TDL</i>
Функция разделения обучающего множества	<i>Ex: divideind</i>
Число примеров в подмножествах	<i>Ex: 170-20-10</i>
Метод обучения	<i>Ex: trainlm</i>
Параметры обучения	<i>Ex: lr = 0.01</i>
Метод инициализации сети	<i>Ex: rands</i>
Критерий окончания обучения	<i>Ex: epochs = 1000, goal = 10^{-8}</i>
Причина окончания обучения	<i>Ex: достигнуто предельное значение критерия обучения</i>
Число эпох обучения	<i>Ex: 723</i>

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:

$$R^2 = 1 - \frac{SSE}{SS_{yy}},$$

$$SSE = \sum (y - \hat{y})^2,$$

$$SS_{yy} = \sum (y - \bar{y})^2.$$

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__