

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

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

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

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, 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.

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 Отобразить 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

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

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

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

Для отображения результата распознавания использовать вызов следующих функций:

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

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]

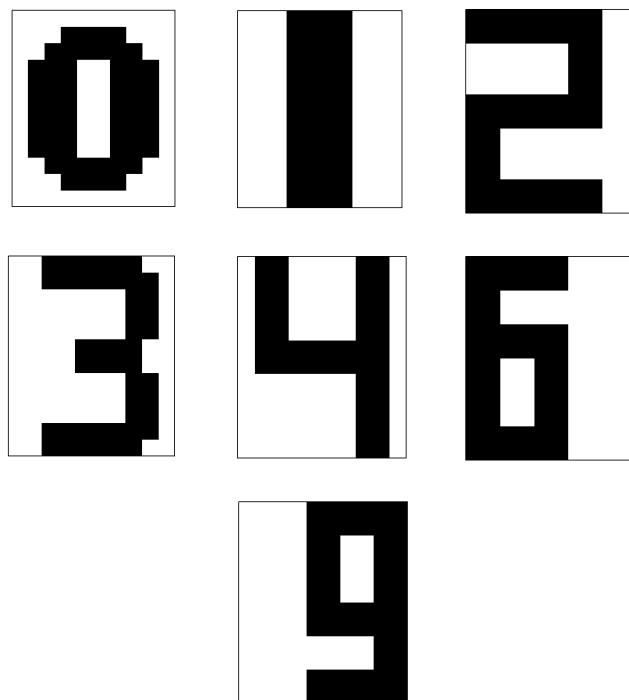


Рис. 2. Эталонные образы

№	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 Хэмминга, 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 сетью прямого распространения. Функционирование сети производится в соответствии с правилами:

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

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

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.