

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 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:

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

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

- 1.1 Число Вольфа — один из 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
7.	04/1761

№	MM/TTTT
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

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

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

Преобразовать обучающее множество с помощью функции *con2seq*.

2.2 Создать сеть с помощью функции *distdelaynet*. Задать задержки $[0 : 4]$ для входного и скрытого слоев. Число нейронов скрытого слоя задано 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, 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.

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 Отобразить 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

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

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:

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

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 $u(n)$.

3.1 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 $y(k)$.

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 Отобразить 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.