



ДОНСКОЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ
УПРАВЛЕНИЕ ДИСТАНЦИОННОГО ОБУЧЕНИЯ И ПОВЫШЕНИЯ
КВАЛИФИКАЦИИ

Кафедра «Робототехника и Мехатроника»

Учебно-методическое пособие по дисциплине

«Искусственный интеллект в мехатронике и робототехнике»

на тему:

Нахождение экстремума функций с использованием рекуррентной нейронной сети

Авторы
Тугенгольд А.К.,
Юсупов А.Р.,
Волошин Р.Н.

Ростов-на-Дону, 2015



Аннотация

Методические указания предназначены для студентов специальности 15.03.01 очной формы обучения.

Авторы

д.т.н. профессор Тугенгольд А.К.

ст. преподаватель Юсупов А.Р.

магистрант группы УММР11 Волошин Р.Н.





Содержание

Цели и задачи	5
Общие сведения.....	5
Процедура обучения	7
Порядок выполнения работы в системе MATLAB.	7
Последовательность выполнения работы.....	11
Задания.....	12
Вопросы для самоконтроля.	12
Литература.....	13



Введение

Определение нейронных сетей возникло из исследований в области искусственного интеллекта, в ходе которых ученые наблюдали способность биологических нервных систем обучаться и исправлять ошибки, моделировали низкоуровневую структуру мозга. Основным направлением исследований в области искусственного интеллекта в 60-е - 80-е годы были экспертные системы. Такие системы основывались на высокоуровневом моделировании процесса мышления. Однако, повторить структуру головного мозга не удалось и по сей день, поэтому разработки проводят на основе построения нейронных сетей, выполняющих часть функций головного мозга.

Нейроны - это нервные клетки, способные распространять электрохимические сигналы. Нейрон имеет разветвленную структуру ввода информации (дендриты), ядро и разветвляющийся выход (аксон). Аксоны клетки [1] соединяются с дендритами других клеток с помощью синапсов. При активации нейрон посылает электрохимический сигнал по своему аксону. Через синапсы этот сигнал достигает других нейронов, которые могут в свою очередь активироваться. Нейрон активируется тогда, когда суммарный уровень сигналов, пришедших в его ядро из дендритов, превысит определенный уровень (порог активации). Таким образом, будучи построен из очень большого числа совсем простых элементов, мозг способен решать чрезвычайно сложные задачи. При этом каждый из нейронов осуществляет суммирование входных сигналов и в случае, если сумма значений на входе превышает определенный уровень, представляет на выходе двоичный сигнал. Ис-



Искусственный интеллект в мехатронике и робототехнике

Искусственные нейронные сети имеют ту же структуру организации и поведения, что и сети в головном мозге. На основе нейронных сетей строятся системы, способные выполнять некоторые функции головного мозга. В частности, можно обучить нейронную сеть таким образом, чтобы она определяла экстремумы заданной функции.

Цели и задачи

Цель работы: изучение алгоритма настройки параметров нейронных сетей с помощью процедуры обучения **train** в системе MATLAB.

Освоение практического применения ИНС для решения задач нахождения экстремальных значений функций заданных таблично с помощью нейронной сети.

Задачи, решаемые в данной работе:

- Получение навыков обучения нейронной сети,
- Получение навыков программирования в пакете MATLAB,
- создание и обучение нейронной сети определения экстремума
- Экспериментальное исследование работы ИНС в соответствии с вариантом задания.

Общие сведения

Для настройки параметров нейронной сети используется рекуррентное обучающее правило наименьших квадратов, которое позволяет получить лучший результат, по сравнению с обучающим правилом персептрона [2,3].

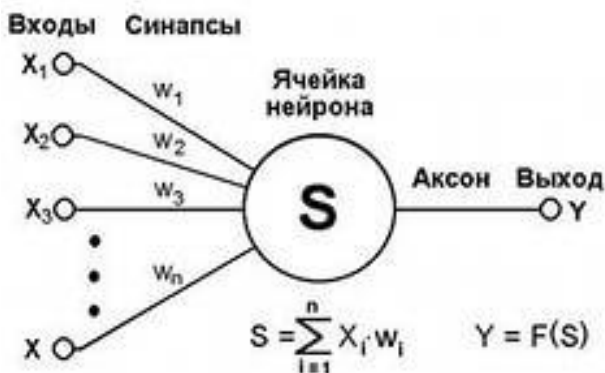


Рисунок 1. Структура искусственного нейрона.

Правило наименьших квадратов, или правило обучения ВН (Уидроу–Хоффа), минимизирует среднее значение суммы квадратов ошибок обучения. Рекуррентные нейронные сети — вид нейронных сетей, в которых имеется обратная связь. При этом под обратной связью подразумевается связь от логически более удалённого элемента к менее удалённому. Наличие обратных связей позволяет запоминать и воспроизводить целые последовательности реакций на один стимул. С точки зрения программирования в таких сетях появляется аналог циклического выполнения.

Отклик таких сетей является динамическим, т. е. после приложения нового входа вычисляется выход и, передаваясь по сети обратной связи, модифицирует вход. Затем выход повторно вычисляется, и процесс повторяется снова и снова. Для устойчивой сети последовательные итерации приводят к все меньшим изменениям выхода, пока в итоге выход не становится постоянным. Для многих сетей процесс никогда не заканчивается, такие сети называют неустойчивыми.



Процедура обучения

Для обучения нейронной сети может быть применена типовая процедура обучения с помощью функции **train** [3]. Эта функция для каждого вектора входа выполняет настройку весов и смещений, используя функцию **learnp**. В результате сеть будет настраиваться по сумме всех коррекций.

Каждый пересчет весов и смещений для набора входных векторов называется эпохой. Это и отличает процедуру обучения **learnp** от процедуры адаптации **adapt**, когда настройка параметров реализуется при представлении каждого отдельного вектора входа.

Затем процедура **train** моделирует настроенную сеть для имеющегося набора векторов, сравнивает результаты с набором целевых векторов и вычисляет среднеквадратичную ошибку. Как только значение ошибки становится меньше заданного или истерпано предельное число эпох, обучение прекращается.

Порядок выполнения работы в системе MATLAB.

1. В программной среде MATLAB создаем m-файл, называем его при помощи латинских букв, сохраняем на устройстве.

2. Входные данные:

Вектор входов: **P= [1 -1.2]**. Вектор целей: **T= [0.5, 1]**.

Вес: **w= -0,2279**. Смещение: **b= 0,7249**.

3. Открываем созданный m-файл, в появившемся окне начинаем ввод программы:



Искусственный интеллект в мехатронике и робототехнике

3.1 Задаем вектор входов **P**, и вектор целей **T**.

clear, P = [1 -1.2]; T = [0.5, 1];

3.2 Задаем максимальное значение параметра обучения, по достижении которого система будет останавливать обучение нейронной сети, где **maxlr** – параметр скорости настройки.

maxlr = 0.40*maxlinlr(P,'bias');

3.3 Создание нейронной сети:

net = newlin([-2,2],1,[0],maxlr);

где [-2,2] – диапазон значений входа, массив размера Px2 минимальных и максимальных значений для P векторов входа; 1 – число нейронов в слое; [0] – описание линий задержек на входе слоя.

3.4 Расчет функции критерия качества

w_range=-1:0.2:1; b_range=-1:0.2:1;

ES = errsurf(P,T, w_range, b_range, 'purelin');

3.5 Построение поверхности функции критерия качества

surfc (w_range, b_range, ES);

На рисунке 2.а, изображена поверхность функции ошибки в пространстве параметров сети. В процессе обучения нейронной сети траектория обучения (показанная на рисунке 2.б) будет перемещаться из начальной точки в точку минимума критерия качества.

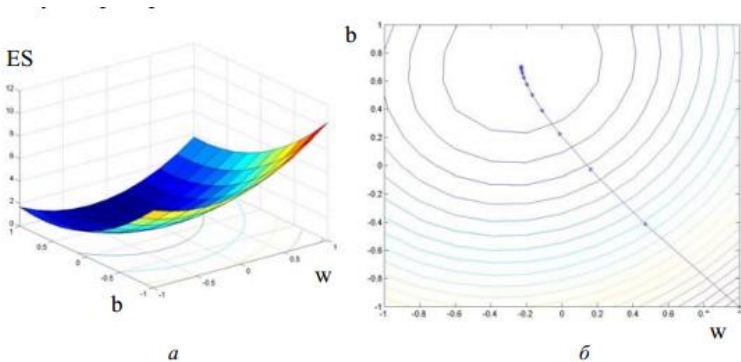


Рисунок 2. Функция ошибки и траектория обучения

Выполним расчет и построим траекторию обучения нейронной сети для заданных начальных значений веса и смещения:

4. Построение алгоритма расчета траектории обучения

```
x = zeros(1,50); y = zeros(1,50);
net.IW{1}=1; net.b{1}= -1;
x(1) = net.IW{1}; y(1) = net.b{1};
net.trainParam.goal = 0.001;
net.trainParam.epochs = 1;
```

5. Построение цикла вычисления весов и смещения для одной эпохи:

```
for i = 2:50,
  [net, tr] = train(net,P,T);
  x(i) = net.IW{1};
  y(i) = net.b{1};
end
```



Искусственный интеллект в мехатронике и робототехнике

6. Результат, представленный в рабочей области программы:

TRAINB, Epoch 0/1, MSE 5.245/0.001.

TRAINB, Epoch 1/1, MSE 2.049/0.001.

TRAINB, Maximum epoch reached.

TRAINB, Epoch 0/1, MSE 2.049/0.001.

.....

TRAINB, Epoch 0/1, MSE 0.000483544/0.001.

TRAINB, Performancegoalmet.

7. Построение линий уровня и траектории обучения, аналогично представленных на рисунке 2.а и 2.б:

**clf, contour(w_range, b_range, ES, 20), hold on
plot(x, y, '-*'), grid on, hold off,**

Для проверки полученных данных проведем обучение нейронной сети без построения траектории процесса обучения. Для этого обратимся к функции **train** только один раз.

8. Проверка:

**net.IW{1}=1; net.b{1}= -1;
net.trainParam.epochs = 50;
net.trainParam.goal = 0.001;
[net, tr] = train(net,P,T);
net.IW, net.b**

9. Результат проверки:

TRAINB, Epoch 0/50, MSE 5.245/0.001.

TRAINB, Epoch 11/50, MSE 0.000483544/0.001.



TRAINB, Performance goal met.

ans = [-0.2289]

ans = [0.7052]

Последовательность выполнения работы

1. Для заданного преподавателем варианта задания (таблица 1) создать нейронную сеть с помощью функции **newlin** и осуществить ее настройку (обучение) при помощи функции **train**.

2. Построить график функции ошибки и график траектории процесса обучения в системе MATLAB (аналогично рис. 1).

3. Провести проверку полученных результатов при помощи однократного использования функции **train**.

3. Выводы по проделанной работе.

4. Составить отчет, который должен содержать:

- цель лабораторной работы;
- структурную схему нейрона;
- последовательность обучения нейронной сети, текст программы, графики функции ошибки и траектории процесса обучения;

– выводы.



Задания

Таблица 1. Варианты заданий

Номер варианта	Количество входов - 1; количество нейронов – 1.		
	Диапазон значений входов	Значение входа нейрона	Целевой выход
1	-3..+3	{-2 0}	{1 1}
2	-4..+4	{-2 1}	{1 -2}
3	-2..+2	{-1 0}	{-1 0}
4	-4..+4	{0 2}	{0 -2}
5	-3..+3	{-1 1}	{-2 0}
6	-2..+2	{1 0}	{1 -1}
7	-3..+3	{1 -1}	{0 1}
8	-4..+4	{-2 2}	{2 -2}
9	-3..+3	{2 1}	{2 1}
10	-2..+2	{-1 1}	{-1 2}

Вопросы для самоконтроля

1. Что такое нейронная сеть? Что такое рекуррентная нейронная сеть?
2. Для чего применяются нейронные сети?
3. Какие правила нахождения экстремумов функции вам известны?
4. Описать последовательность обучения нейронной сети в пакете MATLAB.
5. Что входит в состав нейрона?



Литература

1. Тугенгольд А.К., Рубанчик В.В. Искусственный интеллект в машиностроительных технологических системах: учеб. пособие, Ростов-на-Дону: издательский центр ДГТУ/ 1996, 140 с.
2. Рекуррентная нейронная сеть.
https://ru.wikipedia.org/wiki/Рекуррентная_нейронная_сеть. Дата обращения 16.04.2014
3. Заде Л. Понятие лингвистической переменной и ее применение к принятию приближенных решений. М.: Мир, 1976. 165 с.
4. Леоненков А.В. Нечеткое моделирование в среде MATLAB и fuzzyTECH. – СПб: БХВ-Петербург, 2005. –221с.