

Введение

В ряде методов, созданных для реализации искусственного интеллекта, используются явные представления знаний и тщательно спроектированные алгоритмы перебора. Отличный от этого подход состоит в построении интеллектуальных программ с использованием моделей, имитирующих нейронные структуры в человеческом мозге или эволюцию разных альтернативных конфигураций, как это делается в генетических алгоритмах и искусственной жизни. Для того чтобы решать сложные и плохо формализуемые задачи и возникло направление, которое называется искусственные нейронные сети.

Искусственные нейронные сети состоят из нейроноподобных элементов, соединенных между собой в сеть. Существуют статические и динамические нейронные сети. В статических нейронных сетях изменение параметров системы происходит по некоторому алгоритму в процессе обучения. После обучения параметры сети не меняются. В динамических нейронных сетях отображение внешней информации, и ее обработка осуществляется в виде некоторого динамического процесса, то есть процесса, зависящего от времени.

Нейронные сети нашли применение практически во всех областях науки и техники. С использованием нейронных сетей успешно решаются многие проблемы бизнеса и финансов. Задачи управления, классификации, распознавания образов, прогнозирования, присущие практически всем прикладным областям, таким как медицина, военное дело, авиация и космос, строительство, все чаще решаются с применением нейросетевых технологий.

В связи с этим необходимо дать студентам, обучающимся современным информационным технологиям, фундаментальное представление об основных понятиях и моделях нейронных сетей, а также научить применять эти знания на практике.

Данные лабораторные работы направлены на выработку практических навыков построения искусственных нейронных сетей в среде MatLab. Особенностью выполнения работ является то, что студент приучается к самостоятельному принятию решений, развиваются его исследовательские способности, что особенно важно в динамично развивающемся мире. В настоящих методических указаниях рассматриваются нейронные сети прямой передачи сигнала, радиальные базисные сети, сеть Кохонена и сеть Хопфилда, являющиеся типовыми технологиями искусственных нейронных сетей.

Лабораторная работа № 1

Аппроксимация функции одной переменной

1.1. Цель работы

Научиться работать с сетью прямой передачи сигнала, функция `newff` [1]. Разобраться с алгоритмом обратного распространения ошибки.

1.2. Краткие теоретические сведения

В лабораторной работе рассматривается нейронная сеть с прямой передачей сигнала (с прямой связью) [2], то есть сеть, в которой сигналы передаются только в направлении от входного слоя к выходному, и элементы одного слоя связаны со всеми элементами следующего слоя. Важнейшим для реализации нейронных сетей является определение алгоритма обучения сети.

В настоящее время одним из самых эффективных и обоснованных методов обучения нейронных сетей является *алгоритм обратного распространения ошибки*, который применим к *однонаправленным многослойным сетям*. В многослойных нейронных сетях имеется множество скрытых нейронов, входы и выходы которых не являются входами и выходами нейронной сети, а соединяют нейроны внутри сети, то есть *скрытые нейроны*. Занумеруем выходы нейронной сети индексом $j = 1, 2, \dots, n$, а обучающие примеры индексом $M = 1, 2, \dots, M_0$. Тогда в качестве целевой функции можно выбрать функцию ошибки как сумму квадратов расстояний между реальными выходными состояниями y_{jM} нейронной сети, выдаваемых сетью на входных данных примеров, и правильными значениями функции d_{jM} , соответствующими этим примерам. Пусть $\mathbf{x} = \{x_i\}$ – столбец входных значений, где $i = 1, 2, \dots, n$. Тогда $\mathbf{y} = \{y_j\}$ – выходные значения, где $j = 1, 2, \dots, m$. В общем случае $n \neq m$. Рассмотрим разность $y_{jM} - d_{jM}$, где d_{ji} – точное (правильное) значение из примера. Эта разность должна быть минимальна. Введем расстояния согласно евклидовой метрике, определив норму

$$\|\mathbf{y} - \mathbf{d}\| = \sqrt{(\mathbf{y} - \mathbf{d}, \mathbf{y} - \mathbf{d})}^2. \quad (1)$$

Пусть целевая функция имеет вид

$$E = \frac{1}{2} \sum_{j,M} (y_{j,M} - d_{j,M})^2. \quad (2)$$

Коэффициент $\frac{1}{2}$ выбран из соображений более короткой записи последующих формул. Задача обучения нейронной сети состоит в том, чтобы найти такие коэффициенты $w_{\beta k}$, при которых достигается минимум $E(\mathbf{w})$ ($E \geq 0$).

На рис. 1 показана архитектура нейронной сети с прямой передачей сигнала.

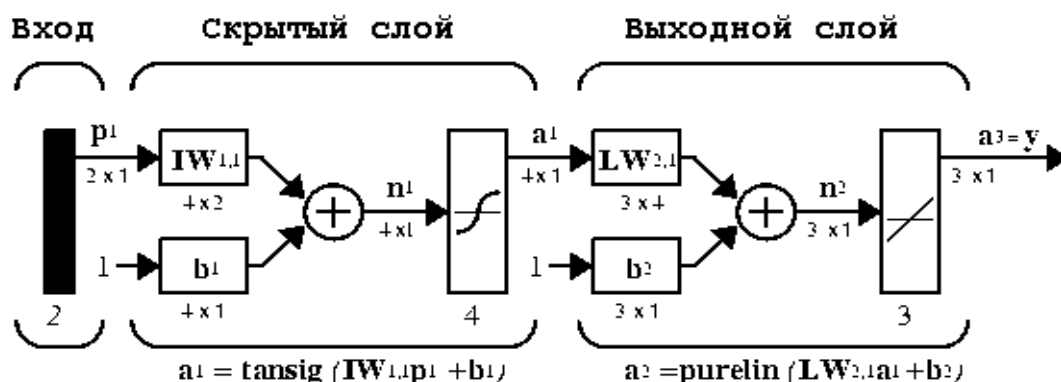


Рис. 1. Схема архитектуры нейронной сети с прямой передачей сигнала

Здесь приняты обозначения, используемые в [1], а именно, p^1 - вектор входа, $IW^{i,j}$, $LW^{i,j}$ - матрицы весов входа и выхода, b^i - смещение, a^i - выход слоя, y - выход сети, *tansig* (гиперболическая тангенциальная), *purelin*(линейная)- соответствующие функции активации.

Веса и смещения определяются с помощью алгоритма обратного распространения ошибок[3].

Обучение сети обратного распространения требует выполнения следующих операций:

1. Выбрать очередную обучающую пару из обучающего множества; подать входной вектор на вход сети.
2. Вычислить выход сети.
3. Вычислить разность между выходом сети и требуемым выходом (целевым вектором обучающей пары).
4. Скорректировать веса сети так, чтобы минимизировать ошибку.
5. Повторять шаги с 1 по 4 для каждого вектора обучающего множества до тех пор, пока ошибка на всем множестве не достигнет приемлемого уровня.

1.3. Пример решения типовой задачи

Выполнение лабораторной работы состоит из следующих этапов: прежде всего, необходимо оцифровать график функции $y=f(x)$, то есть получить ряд соответствующих значений по горизонтальной и вертикальной осям.

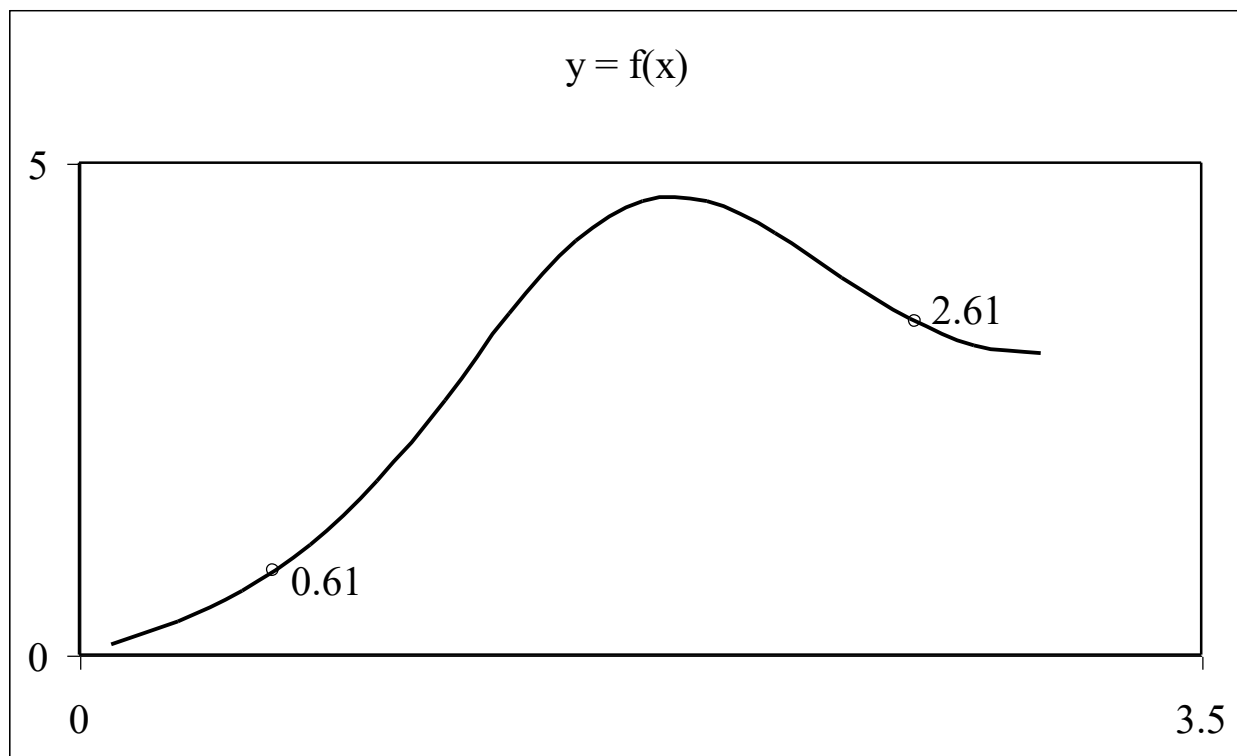


Рис. 2. Пример зависимости для функции одной переменной

В примере, показанном на рис. 2 были получены два массива, каждый из которых состоит из 15 значений. По горизонтальной оси— [0.10 0.31 0.51 0.72 0.93 1.14 1.34 1.55 1.76 1.96 2.17 2.38 2.59 2.79 3.00]. По вертикальной оси— [0.1010 0.3365 0.6551 1.1159 1.7632 2.5847 3.4686 4.2115 4.6152 4.6095 4.2887 3.8349 3.4160 3.1388 3.0603].

Ниже приводится программа создания, обучения нейронной сети и вывода результатов.

```
x=[0.10 0.31 0.51 0.72 0.93 1.14 ...
1.34 1.55 1.76 1.96 2.17 2.38 ...
2.59 2.79 3.00];
y=[0.1010 0.3365 0.6551 1.1159 1.7632 2.5847 ...
3.4686 4.2115 4.6152 4.6095 4.2887 3.8349 ...
3.4160 3.1388 3.0603];
net=newff([0 3],[5,1],{'tansig','purelin'},'trainbfg');
net.trainParam.epochs=300;
net.trainParam.show=50;
net.trainParam.goal=1.37e-2;
[net,tr]=train(net,x,y);
an=sim(net,x);
plot(x,y,'+r',x,an,'-g'); hold on;
xx=[0.61 2.61];
v=sim(net,xx)
plot(xx,v,'ob','MarkerSize',5,'LineWidth',2)
```

В результате выполнения программы получаются следующие результаты, отражённые на рис. 3 и 4:

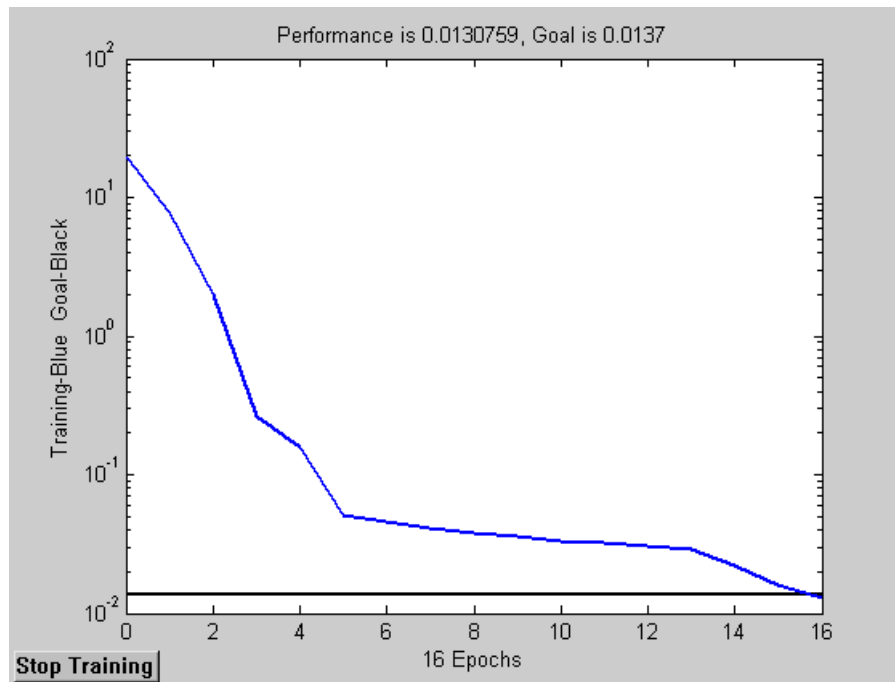


Рис. 3. Характеристика точности обучения в зависимости от числа эпох обучения

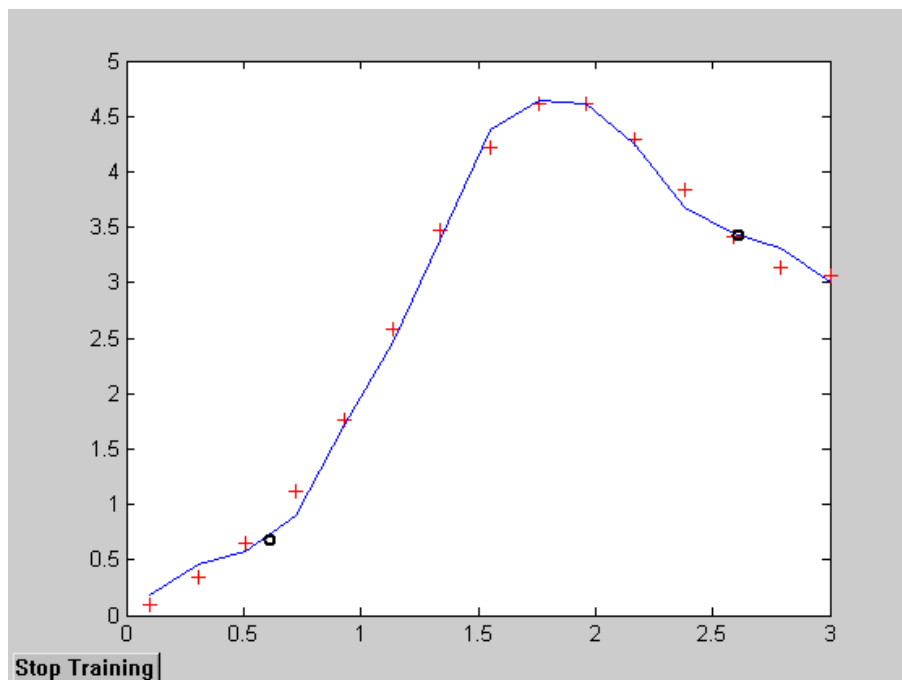


Рис. 4. Результаты моделирования сети: + - исходные данные; сплошная линия и символ «о» – результаты моделирования всей зависимости и в контрольных точках

В массиве \mathbf{v} содержатся приближённые значения для двух контрольных точек, указанных на графике (рис. 2) $\mathbf{xx} = [0.61 \ 2.61]$. При данных параметрах сети получены значения: $\mathbf{v} = [1.05 \ 3.35]$. Сравнив эти

приближённые значения с точными значениями [0.85 3.37], можно сделать вывод о корректности построения нейронной сети.

1.4. Отчёт о выполнении работы

Отчёт о выполнении лабораторной работы №1 должен быть выполнен на листах формата А4 и содержать следующие результаты:

1. Исходные данные (рис. 2);
2. Текст программы с подробными комментариями;
3. Характеристику точности обучения (рис. 3);
4. Результаты моделирования (рис. 4);
5. Сопоставление результатов в контрольных точках;
6. Краткие выводы о результатах работы.

