

Лабораторная работа №3

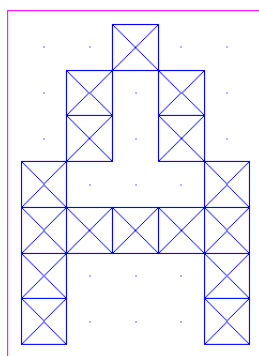
Цель: Изучение пакета прикладных программ MatLab Neural Network ToolBox на примере моделирования искусственной нейронной сети для распознавания символов латинского алфавита. А также, обзор основных возможностей GUI-интерфейса для NNT.

Вводная часть

Часто полезно иметь устройство, которое выполняет распознавание образов. В частности, очень эффективны и выгодны машины, которые могут читать символы. Машина, которая читает банковские чеки, может выполнять за то же самое время намного больше проверок, чем человек. Этот вид приложений сохраняет время и деньги, а также устраняет условия, при которых человек выполняет монотонную, периодически повторяющуюся работу. В данной лабораторной рассматривается задача распознавания символов, которая может быть выполнена при помощи искусственной нейронной сети с обратным распространением.

Постановка задачи

Требуется создать нейронную сеть для распознавания 26 символов латинского алфавита. В качестве датчика предполагается использовать систему распознавания, которая выполняет оцифровку каждого символа, находящегося в поле зрения. В результате каждый символ будет представлен шаблоном размера 7x5. Например, символ А может быть представлен, как это показано на рис. 1.



0	0	1	0	0
0	1	0	1	0
0	1	0	1	0
1	0	0	0	1
1	1	1	1	1
1	0	0	0	1
1	1	0	0	1

Рисунок 1. Представления символа в виде шаблона 7x5

Однако система считывания символов обычно работает неидеально, и отдельные элементы символов могут оказаться искаженными (рис. 2).

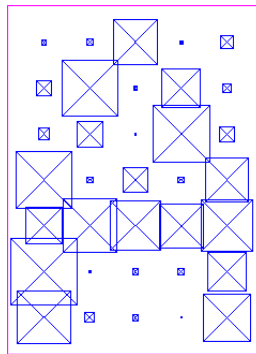


Рисунок 2. Зашумленный шаблон символа

Проектируемая нейронная сеть должна точно распознавать идеальные векторы входа и с максимальной точностью воспроизводить зашумленные векторы. М-функция `prprob` определяет 26 векторов входа, каждый из которых содержит 35 элементов, этот массив называется *алфавитом*. М-функция формирует выходные переменные `alphabet` и `targets`, которые определяют массивы алфавита и целевых векторов. Массив `targets` определяется как `eye(26)`. Для того чтобы восстановить шаблон для *i*-й буквы алфавита, надо выполнить следующие операторы:

```
[alphabet, targets] = prprob
i = 2
ti = alphabet(:, i)
letter{i} = reshape(ti, 5, 7)'
letter{i}
```

Для визуализации 35-размерного вектора, представляющего символ, можно использовать функцию `plotchar(ti)`.

Для генерации зашумленного образа символа “J” можно применять следующую последовательность команд:

```
noisyJ = alphabet(:, 10) + randn(35, 1)*0.2
```

Нейронная сеть

На вход сети поступает вектор входа с 35 элементами; вектор выхода содержит 26 элементов, только один из которых равен 1, а остальные - 0. Правильно функционирующая сеть должна ответить вектором со значением 1 для элемента, соответствующего номеру символа в алфавите. Кроме того, сеть должна быть способной распознавать символы в условиях действия шума. Предполагается, что шум - это случайная величина со средним значением 0 и стандартным отклонением, меньшим или равным 0.2.

Архитектура сети

Для работы нейронной сети требуется 35 входов и 26 нейронов в выходном слое. Для решения задачи выберем двухслойную нейронную сеть с логарифмическими

сигмоидальными функциями активации в каждом слое. Такая функция активации выбрана потому, что диапазон выходных сигналов для этой функции определен от 0 до 1, и этого достаточно, чтобы сформировать значения выходного вектора. Структурная схема такой нейронной сети показана на рис. 3.

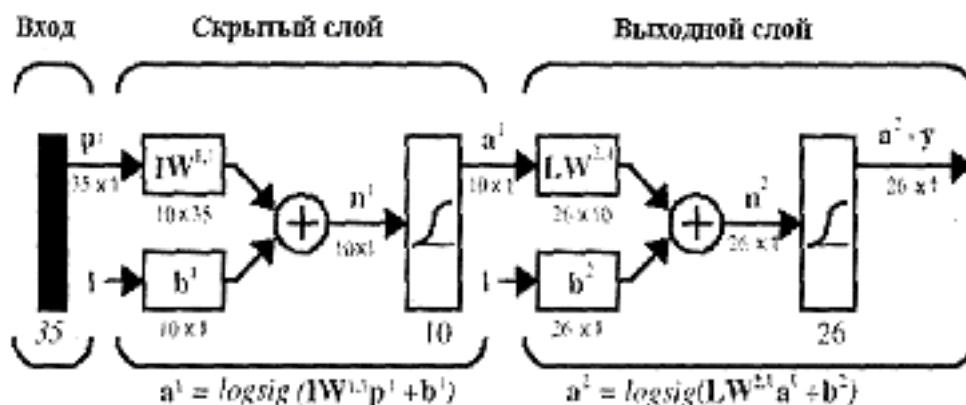


Рисунок 3. Структура проектируемой нейронной сети

Скрытый слой имеет 10 нейронов. Такое число нейронов выбрано на основе опыта и разумных предположений. Если при обучении сети возникнут затруднения, то можно увеличить количество нейронов этого уровня. Сеть обучается так, чтобы сформировать единицу в единственном элементе вектора выхода, позиция которого соответствует номеру символа, и заполнить остальную часть вектора нулями. Однако наличие шумов может приводить к тому, что сеть не будет формировать вектора выхода, состоящего точно из единиц и нулей. Поэтому по завершении этапа обучения выходной сигнал обрабатывается М-функцией `comet`, которая присваивает значение 1 единственному элементу вектора выхода, а всем остальным - значение 0.

Обучение

Чтобы создать нейронную сеть, которая может обрабатывать зашумленные векторы входа, следует выполнить обучение сети как на идеальных, так и на зашумленных векторах. Сначала сеть обучается на идеальных векторах, пока не будет обеспечена минимальная сумма квадратов погрешностей. Затем сеть обучается на 10 наборах идеальных и зашумленных векторов. Две копии свободного от шума алфавита используются для того, чтобы сохранить способность сети классифицировать идеальные векторы входа. К сожалению, после того, как описанная выше сеть обучается классифицировать сильно зашумленные векторы, она теряет способность правильно классифицировать некоторые векторы, свободные от шума. Следовательно, сеть снова надо обучить на идеальных векторах. Это гарантирует, что сеть будет работать правильно, когда на ее вход будет передан идеальный символ. Обучение выполняется с помощью функции `trainbpx`, которая реализует метод обратного распространения ошибки с возмущением и адаптацией параметра скорости настройки.

Если необходима более высокая точность распознавания, сеть может быть обучена либо в течение более длительного времени, либо с использованием большего количества нейронов в скрытом слое. Можно также увеличить размер векторов, чтобы пользоваться шаблоном с более мелкой сеткой, например 10x14 точек вместо 5x7.

Заключение

Эта задача демонстрирует, как может быть разработана простая система распознавания изображений при помощи пакета MatLab Neural Network ToolBox. Обучение сети на различных наборах зашумленных векторов позволяет обучить сеть работать с изображениями, искаженными шумами, что характерно для реальной практики.

Варианты заданий

Вариант №1	Вариант №2
<p>1. Выполнить построение нейронной сети <u>в режиме командной строки MatLab</u>, используя предопределенные функции пакета Neural Network ToolBox.</p> <p>2. Обучить нейронную сеть.</p> <p>3. Протестировать полученную нейронную сеть, используя описанную выше методику.</p>	<p>1. Выполнить построение нейронной сети <u>в графическом интерфейсе пользователя NNTool среды MatLab</u>, используя возможности “Импорта” и “Экспорта”.</p> <p>2. Обучить нейронную сеть.</p> <p>3. Протестировать полученную нейронную сеть, используя описанную выше методику.</p>

Приложения

- файл *Neural Network Toolbox_functions.pdf* – приведена информация по основным функциям для работы в ППП NNT;
- файл *Neural Network Toolbox_forms.pdf* – методический материал по GUI-интерфейсу для NNT.