электронный журнал
МОЛОДЕЖНЫЙ НАУЧНО-ТЕХНИЧЕСКИЙ ВЕСТНИК
Издатель ФГБОУ ВПО "МГТУ им. Н.Э. Баумана". Эл No. ФС77-51038.

УДК 004.032.26

Обучение двухслойного персептрона в среде MATLAB

***Тимофеева М. С.**, студент*
Россия, 105005, г. Москва, МГТУ им. Н.Э. Баумана,
кафедра «Радиоэлектронные системы и устройства»

Научный руководитель: Ахияров В.В.,к.т.н, доцент
кафедра «Радиоэлектронные системы и устройства»,
Россия, 105505, г. Москва, МГТУ им. Н.Э.Баумана,
rl1@bmstu.ru

Под нейронной сетью обычно понимают специальную математическую модель и ее программную и (или) аппаратную реализацию. Существуют нейронные сети различной архитектуры. Наиболее распространенными являются сети на основе многослойных персептронов. В данной работе рассматривается двухслойная сеть прямого распространения или двухслойный персептрон, как наиболее часто использующаяся на практике для решения задач классификации и распознавания. Под персептроном понимается алгоритм классификации, принцип работы которого основан на модели нервной клетки – нейрона.

Алгоритм работает в двух режимах – режиме обучения и режиме тестирования (работы обученной модели). Пусть нам известны входные данные и выходной результат. Наша задача подобрать такие параметры сети, которые по заданным входам дают известный выход. Режим такого подбора и называется режимом обучения.

На рис. 1 представлено строение двухслойного персептрона, который будет использоваться в данной работе для решения задачи распознавания образов. Кружками обозначены нейроны. В первом слое три нейрона, во втором – два.

Персептрон может иметь различное количество входов, выходов (обычно определяется постановкой задачи), скрытых (внутренних) слоев, нейронов на слое. В процессе обучения в простейшем случае подбираются веса связей. На входном слое каждая входная величина расходится по связям (по каждой исходящей из вершины входного слоя прямой идет “сигнал”, равный входной величине), умножается на вес прямой и суммируется

на скрытом слое, а далее преобразуется передаточной функцией нейрона. На выходном слое обычно сигнал только суммируется.

Суть обучения заключается в решении задачи подбора параметров сети (весов связей) для правильного распознавания. Причем из всех обучающих примеров часть используется только для контроля.

Рассмотрим строение и математическое описание двухслойного персептрона.

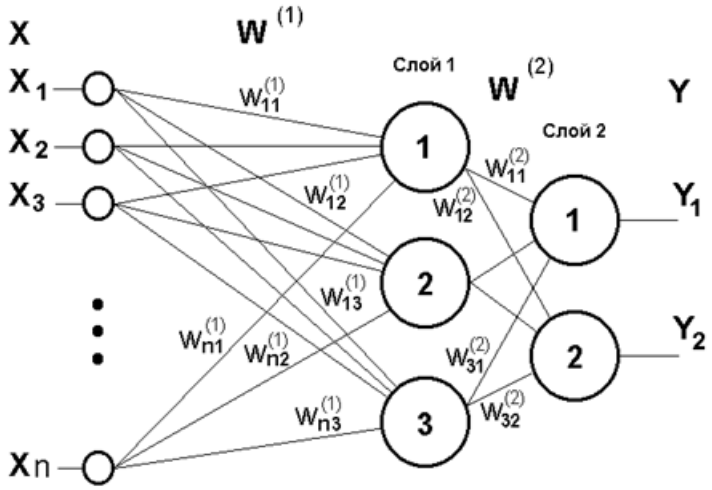

Рис. 1. Двухслойный персептрон [1]

Определим реакцию i – ого нейрона выходного (внешнего) слоя нейросети на входной вектор X_k, где $k = 1,2,3, \dots, n$. Для этого нам сначала нужно определить реакцию внутреннего слоя.

Итак, обозначим набор весов скрытого слоя как $W^{(1)}$ (см. рис. 1), тогда можем найти сетевую функцию, то есть произведение входного воздействия на соответствующий вес нейросети. Сетевая функция j – ого нейрона скрытого слоя для k – ого входного вектора вычисляется по формуле [1]:

$$net_j = \sum_{k=1}^{n} W_{jk}^{(1)} \mathrm{X}_k \,, \qquad (1)$$

где $j = 1,2, \dots, l$ (l – количество нейронов скрытого слоя).

Реакция j – ого нейрона на внутреннем слое определяется из выражения:

$$Z_j = g(net_j) = g\left(\sum_{k=1}^{n} W_{jk}^{(1)} \mathrm{X}_k\right), \qquad (2)$$

где $g(net_j)$ – передаточная функция нейронов данного слоя.

Если обозначить набор весов внешнего слоя как $W^{(2)}$, то сетевая функция i – ого нейрона выходного слоя вычисляется по формуле:

$$net_i = \sum_{j=1}^{l} W_{ij}^{(2)} Z_j = \sum_{j=1}^{l} W_{ij}^{(2)} g\left(\sum_{k=1}^{n} W_{jk}^{(1)} \mathrm{X}_k\right), \quad (3)$$

где $i = 1, \dots, p$ (p – количество нейронов внешнего слоя).

Исходя из этого, мы можем вычислить реакцию i – ого нейрона выходного слоя на входной вектор X_k при помощи следующего выражения:

$$Y_i = g(net_i) = g\left(\sum_{j=1}^{l} W_{ij}^{(2)} Z_j\right) = g\left(\sum_{j=1}^{l} W_{ij}^{(2)} g(\sum_{k=1}^{n} W_{jk}^{(1)} \mathrm{X}_k)\right). \quad (4)$$

Представленные формулы соответствуют прямому распространению сигналов в рассматриваемой нейросети.

Алгоритм обучения такого персептрона основан на минимизации ошибки, вычисленной методом градиентного спуска. Рассмотрим идею и математическое описание данного алгоритма.

Если представить два входных вектора персептрона на координатной плоскости (см. рис. 2), то его работа будет заключаться в том, чтобы определить, по какую сторону разделительной прямой лежит представленный для распознавания вектор. Разделительная прямая, называемая гиперплоскостью, представляет собой след плоскости в системе координат решений. Гиперплоскость является границей, разделяющей пространство решений нейросети.

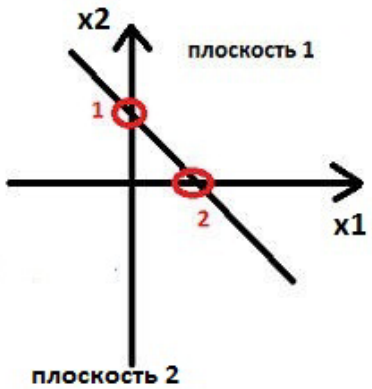

Рис. 2. Координатная плоскость [2]

Таким образом, линейный персептрон может классифицировать только такие образы, которые разделены при помощи гиперплоскости. В качестве примера в таблице истинности показаны значения выходного сигнала для логических функций «И» и «ИЛИ» в зависимости от комбинации входных сигналов. Соответствующие положения гиперплоскости представлены на рис. 3.

Вход 1	Вход 2	Выход
0	0	0
0	1	0
1	0	0
1	1	1

Вход 1	Вход 2	Выход
0	0	0
0	1	1
1	0	1
1	1	1

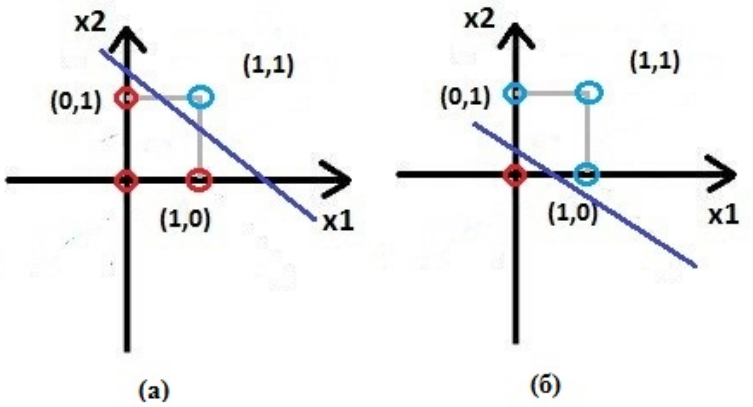

Рис. 3. Логические функции «И» (а) и «ИЛИ» (б) на координатной плоскости

Обучается линейный персептрон путем корректировки весов с учетом ошибки E на выходе сети. Предположим, на вход последовательно подаются векторы X_k, тогда выходная реакция Y_i сравнивается с желаемой реакцией T_i [3]:

$$E = \frac{1}{2}\sum_{i=1}^{p}(T_i - Y_i)^2 . \quad (5)$$

Цель тренировки сети состоит в достижении минимума ошибки E за счет надлежащей настройки весов. Если изменение веса обозначить как ΔW, то по методу градиентного спуска это изменение должно быть равно:

$$\Delta W = \eta\left(-\frac{\partial E}{\partial W}\right). \quad (6)$$

Коэффициент пропорциональности η называют скоростью обучения. Если учитывать величину корректировки ΔW как дополнительный вес, то разделяющая гиперплоскость будет проходить через начало координат, а изменение весового вектора W будет соответствовать вращению гиперплоскости вокруг точки начала координат, как это показано на рис. 4.

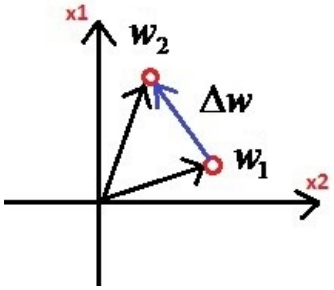

Рис. 4. Изменение положения гиперплоскости при корректировке весового вектора

Таким образом, изменение весовых коэффициентов нейронных связей выходного слоя $W_{ij}^{(2)}$ вычисляется по формуле [4]:

$$\Delta W_{ij}^{(2)} = -\eta \frac{\partial \mathrm{E}}{\partial W_{ij}^{(2)}} = \eta\left([T_i - Y_i]g'(net_i)Z_j\right) = \eta\left(\delta_i^{(2)} Z_j\right), \tag{7}$$

где $\delta_i^{(2)}$ – ошибка нейронов выходного слоя:

$$\delta_i^{(2)} = [T_i - Y_i]g'(net_i)\,. \tag{8}$$

Для изменения весов связей скрытого слоя $W_{jk}^{(1)}$ воспользуемся выражением:

$$\Delta W_{jk}^{(1)} = -\eta \frac{\partial \mathrm{E}}{\partial W_{jk}^{(1)}} = -\eta \frac{\partial \mathrm{E}}{\partial Z_j}\frac{\partial Z_j}{\partial W_{jk}^{(1)}}, \tag{9}$$

откуда следует:

$$\begin{aligned}\Delta W_{jk}^{(1)} &= \eta \sum_{i=1}^{l} [T_i - Y_i]g'(net_i)W_{ij}^{(2)}\, g'\left(net_j\right)\mathrm{X}_k \\ &= \eta \sum_{i=1}^{l} \delta_i^{(2)} W_{ij}^{(2)} g'\left(net_j\right)\mathrm{X}_k = \eta\left(\delta_j^{(1)}\mathrm{X}_k\right),\end{aligned} \tag{10}$$

где $\delta_j^{(1)}$ – ошибка нейронов внутреннего слоя:

$$\delta_j^{(1)} = g'(net_j) \sum_{i=1}^{l} \delta_i^{(2)} W_{ij}^{(2)} . \quad (11)$$

Таким образом, для выходного слоя ошибка вычисляется по формуле (8), для скрытого слоя ошибка вычисляется по формуле (11). То есть, ошибка каждого скрытого слоя вычисляется на основании ошибки следующего слоя. Это распространение ошибки в обратном направлении.

Алгоритм обратного распространения ошибки можно описать следующими шагами:

1. Выполнить инициализацию весов связей небольшими случайными значениями и задать максимальную среднеквадратическую ошибку E.

2. Подать на вход сети входной вектор X_k.

3. Выполнить распространение сигналов в соответствии с прямыми связями.

4. Вычислить среднеквадратическую ошибку по формуле (5) и ошибку нейронов выходного слоя по формуле (8).

5. Вычислить ошибку нейронов внутреннего слоя, руководствуясь формулой (11).

6. Обновить веса связей каждого слоя в соответствии с формулами (7) и (10).

В качестве примера можно привести реальный эксперимент по распознаванию изображений цифр от 0 до 9 в среде MATLAB. Изображения цифр предварительно необходимо привести к бинарному виду и обрезать по размеру для усечения ненужной информации, как показано на рис. 5.

Смоделируем в программе MATLAB двухслойный перцептрон. Для этого следует использовать встроенную функцию *newff*. Эта функция создает двухслойный персептрон с заданной размерностью входов и выходов.

Рис. 5. Исходные изображения с цифрами

Будем обучать его изображениями с цифрами по описанному выше алгоритму, основанному на минимизации ошибки, вычисленной методом градиентного спуска. Для того,

чтобы обучить нейросеть, будем использовать встроенную функцию *train*, на вход которой подаем созданный персептрон. В качестве его параметра необходимо задать используемый алгоритм обучения. Так же зададим количество эпох, равное 100. Эпоха обучения - это один просмотр всех примеров обучающей выборки с одновременной коррекцией весов сети. Обучение проводим по исходным изображениям [5].

Для того, чтобы распознавать исходные изображения с помощью созданной нейросети, их необходимо разложить на составляющие с определенном весом каждый (см. рис. 6). В данной работе используется разложение на 35 квадратов (5 x 7). В левой части рис.6 представлено бинарное изображение цифры два, которое раскладывается на эти составляющие (квадраты). Чем больше черного цвета в выбранном квадрате, тем больше будет вес. На правом изображении представлено разложение изображения цифры два в соответствии с весами составляющих. После разложения на составляющие полученный массив подается на вход нейросети, которая имеет 35 входов и 10 выходов (числа от 0 до 9).

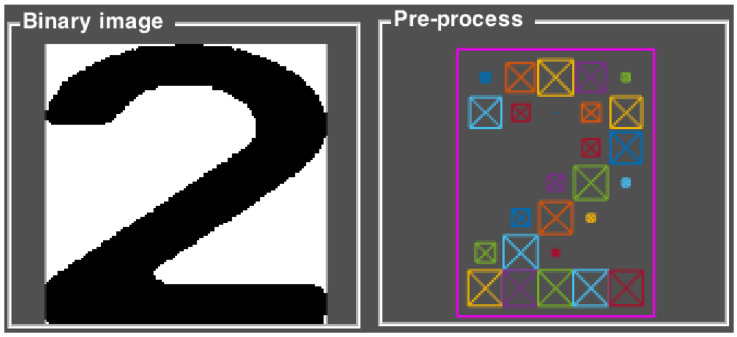

Рис. 6. Разложение числа на составляющие

Для наглядного предоставления результатов работы нейросети была написана программа в среде MATLAB (см. рис. 7), с помощью которой можно загружать исходные изображения и решать задачу распознавания. Загруженное изображение находится в поле под названием “loaded image”. Далее его необходимо перевести в бинарную форму. Результат этой операции можно видеть в поле “binary image”. Затем полученное изображение раскладывается на составляющие, результат показан в поле “pre-process”. В итоге, разложенное изображение поступает на вход нейросети, на выходе которой получаем одно из десяти возможных значений, соответствующее изображению искомого числа. Для сравнения, на рис. 8 приведен результат распознавания цифры без обучения.

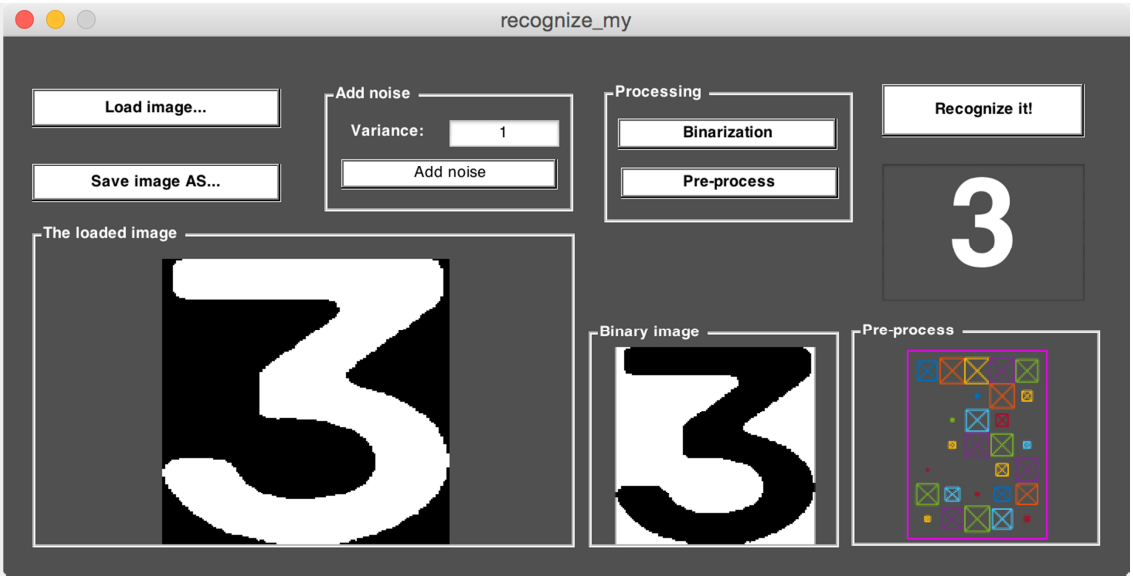

Рис. 7. Результат распознавания с обучением

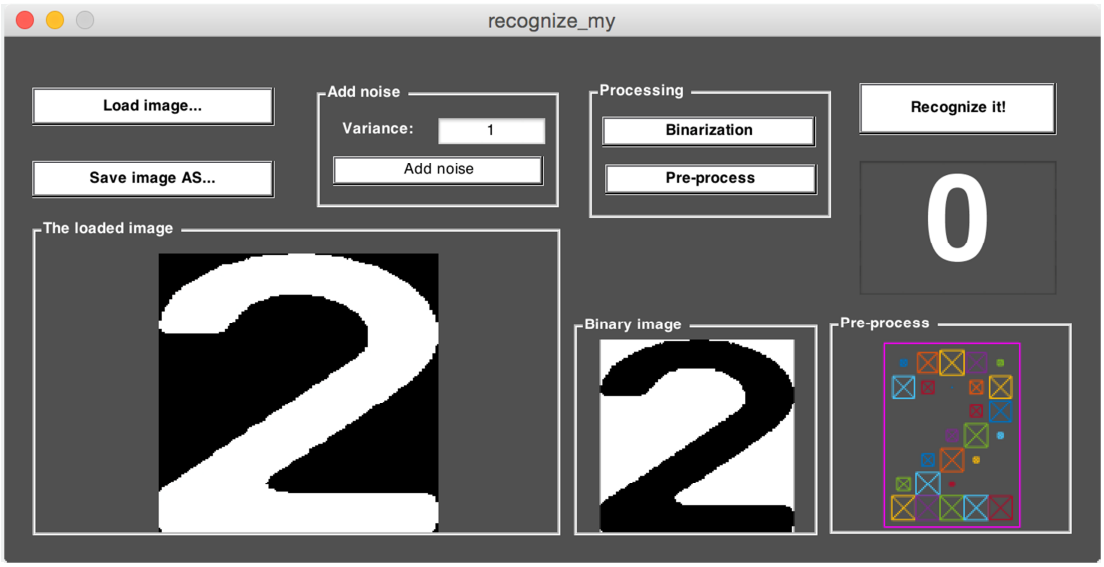

Рис. 8. Результат распознавания без обучения

Таким образом, можно сделать вывод, что нейросеть с обучением показывает хорошие результаты в распознавании изображений, в то время как без обучения нейросеть является практически бесполезной.

Список литературы

1. Осовский С. Нейронные сети для обработки информации. М.: Финансы и Статистика, 2002. 344 с.

2. Каллан Р. Основные концепции нейронных сетей. М.: Вильямс, 2001. 288 с.
3. Хайкин С. Нейронные сети: полный курс. М.: Вильямс, 2006. 1104 с.
4. Уоссермен Ф. Нейрокомпьютерная техника. М: Мир, 1992. 184 с.
5. Шварц Д. Т. Интерактивные методы решения задачи многокритериальной оптимизации. Обзор // Наука и образование. МГТУ им. Н.Э. Баумана. Электрон. журнал. 2013. № 4. Режим доступа: http://technomag.bmstu.ru/doc/547747.html (дата обращения 26.03.2015) .