УДК621.397

ПОСТРОЕНИЕ И ОБУЧЕНИЕ НЕЙРОННОЙ СЕТИ РАСПОЗНАВАНИЯ ОБРАЗОВ ГРАФИЧЕСКИХ ПРИМИТИВОВ

НЕГРАЙ А.В, ЗДАНОВСКАЯ Н.А.

Одесская национальная академия связи им. А.С. Попова

DEVELOPING AND TRAINING OF NEURAL NETWORK FOR PATTERN RECOGNITION GRAPHIC PRIMITIVES

NEGRAY A.V, ZDANOVSKAYA N.A.

Odessa National Academy of Telecommunications n.a. A.S. Popov

Аннотация. *В статье приведены основные сведения по архитектуре нейронных сетей распознавания образов, а также продемонстрировано применение численных методов оптимизации для обучения однослойной искусственной нейронной сети распознавания образов графических примитивов.*

Abstract*. The main data are provided in article on architecture of neural networks for pattern recognition, and also application of numerical methods of optimization for training of an one-layer neural network for pattern recognition of graphic primitives is shown*

ВСТУПЛЕНИЕ

Обучение машин распознаванию образов и программные реализации распознавания находят широкое применение в различных областях науки и техники, в том числе и в системах искусственного интеллекта. Искусственная нейронная сеть распознавания образов представляет собой один из возможных вариантов реализации системы распознавания. Данный подход подсмотрен у природы и основывается на использовании математической модели нейрона. Нейрон представляет собой нервную клетку имеющую несколько входов-дендритов и один выход-аксон. Входы нейрона могут быть как тормозящие, так и возбуждающие, нейрон возбуждается в том случае, если число сигналов пришедших по возбуждающим входам превосходит число сигналов приходящих по тормозящим входам. Совокупность определённо взаимосвязанных, возможно и случайно, нейронов образует нейронную сеть. Использование данной структуры позволяет эффективно решать задачи распознавания образов, что и будет продемонстрировано ниже.

МАТЕМАТИЧЕСКАЯ МОДЕЛЬ НЕЙРОНА, ОСНОВНЫЕ СВЕДЕНИЯ О НЕЙРОННЫХ СЕТЯХ

Нейрон представляет собой единицу обработки информации в нейронной сети. Ниже на рис. 1 представлена математическая нелинейная модель нейрона [1]. В данной модели можно выделить основные части:1) набор синапсов (аналог дендритов реального нейрона), характеризующихся своим весом w_k, на которые поступают элементы входного вектора $x_1, x_2 \dots x_m$; 2) сумматор выполняет суммирование входных сигналов$x_1, x_2 \dots x_m$ взвешенных на веса нейронаw_k; 3) функция активации$\varphi(\cdot)$, или пороговая функция, выполняет ограничение амплитуды выходного сигнала нейрона. Нормализованный диапазон изменения амплитуд выходного сигнала находится в интервале [0 1] или [-1 1].

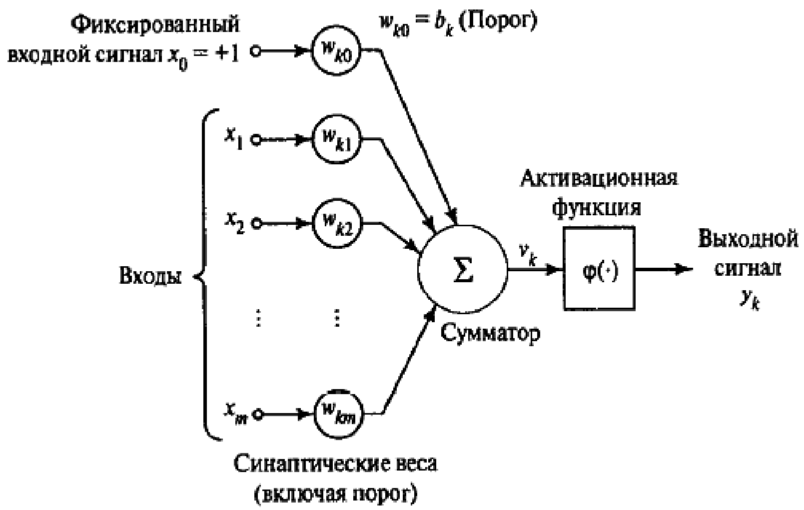

Рисунок 1– Нелинейная модель нейрона

Наиболее часто в качестве пороговой функции при создании искусственных нейронных сетей используют сигмоидальную функцию[1]. Примером сигмоидальной функции является логистическая функция, описываемая выражением:

$$\phi(v) = \frac{1}{1+\exp(-\alpha v)} \tag{1}$$

где α – параметр наклона сигмоидальной функции, v_k – индуцированное локальное поле нейрона:

$$v_k = \sum_{j=1}^{m} w_{kj} x_j + b_k \tag{2}$$

Описанная выше модель нейрона носит название модели Мак-Калока [1], впервые её предложившего. Нейроны могут быть объединены в единую структуру или сеть с использованием одной из следующих архитектур: однослойной сети прямого распространения и многослойной сети прямого распространения. Пример архитектуры однослойной сети приведен на рис. 2 [1].

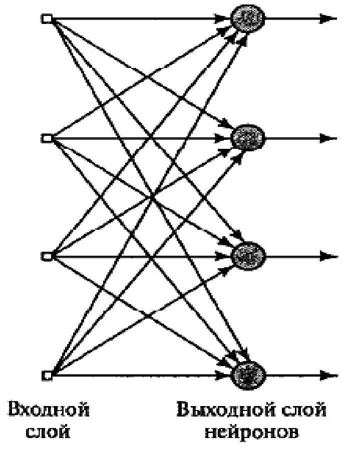

Рисунок 2– Однослойная сеть прямого распространения

Однослойная сеть содержит входной или рецепторный слой, информация от которого передаётся непосредственно на выходной, реагирующий слой вычислительных нейронов. При этом под единственным слоем понимается только выходной слой нейронов, никаких вычислений в рецепторном слое не производится. Другой класс нейронных сетей характеризуется наличием одного или нескольких скрытых слоёв нейронов называемых преобразующими. Функция промежуточного слоя заключается в посредничестве между внешним рецепторным слоем и выходным вычислительным слоем [1]. Пример многослойной сети, содержащей четыре нейрона в промежуточном, скрытом, слое и два нейрона в выходном слое, приведен на рис. 3. Данная сеть называется также полно связной, так как каждый из рецепторов входного слоя связан с каждым из промежуточных нейронов, а каждый из четырёх промежуточных связан со всеми выходными нейронами [1].

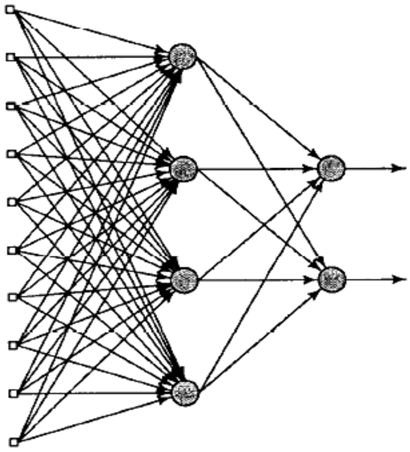

Рисунок 3– Многослойная сеть прямого распространения

Распознавание образа, т.е отнесение примера поданного на вход сети к одному из классов, заключается в возбуждении соответствующего нейрона выходного слоя, причём различным сигналам передаваемым с рецепторного слоя может соответствовать возбуждение одного и того же реагирующего нейрона выходного слоя. Одним из ключевых факторов распознавания является также правильное представление входных данных или признаков объектов, подаваемых на вход сети распознавания [2]. Особенно актуально это при распознавании объектов на изображении, так как один и тот же объект на разных изображениях может иметь различный размер, располагаться на разных участках изображения или быть повёрнутым на какой либо угол, а также иметь различную яркость [2]. Таким образом, необходимо выделять из входного сигнала информативные признаки, которые описывают наиболее существенную информацию, содержащуюся в наборе данных и при этом инвариантны к определённым изменениям входного сигнала [1]. При использовании таких признаков в нейронной сети не нужно хранить лишний объём информации, описывающей изменения объекта. При применении инвариантных признаков отличия между различными экземплярами одного и того же объекта могут быть вызваны только случайными факторами.

ОБУЧЕНИЕ НЕЙРОННОЙ СЕТИ С УЧИТЕЛЕМ

Для решения нейронной сетью задачи распознавания образов необходимо произвести её обучение. Обучение происходит путём итерационной корректировки свободных параметров нейронной сети, которыми являются значения её весов и порогов. Процесс обучения состоит в предъявлении сети учителем входной ситуации описываемой вектором $X = [x_1, x_2, x_3 ... x_m]$ и указанием, к какому из классов она относится, т.е. какой желаемый отклик нейрона сети должен соответствовать данному входному вектору X, и какой из нейронов выходного слоя должен перейти в возбуждённое состояние и обеспечить на своём выходе значение близкое к единице, при условии что остальные нейроны выходного слоя находятся в состоянии покоя и дают на своём выходе значение близкое к нулю [1]. Далее параметры сети, веса и пороги нейронов, корректируются с учётом обучающего вектора X и сигнала ошибки. Сигнал ошибки — это разность между желаемым сигналом и текущим откликом нейронной сети. Корректировка параметров выполняется итерационно с целью имитации нейронной сетью поведения учителя. После окончания обучения учителя можно отключить и предоставить сети возможность работать самостоятельно [1]. Процесс обучения можно проиллюстрировать следующей схемой:

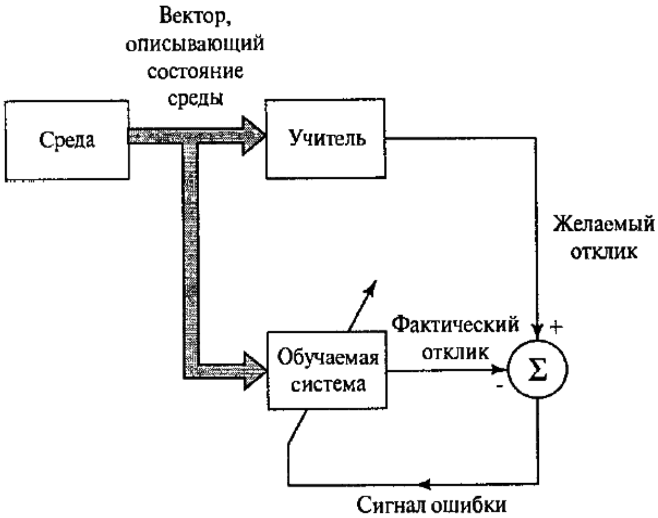

Рисунок4– Схема обучения с учителем

Обучения, таким образом, основывается на пошаговой коррекции ошибок. На вход нейрона выходного или скрытого слоя подаётся образ обучающей выборки, представленный вектором признаков $\mathrm{X} = [x_1, x_2 \ldots x_m]$,который можно также обозначить как $x(n)$. На выходе нейрона устанавливается выходной сигнал $y(n)$ вызванный входным сигналом. Далее производится сравнение полученного выходного сигнала $y(n)$ с желаемым откликом нейрона $d(n)$, указываемым учителем, в результате формируется сигнал ошибки $e(n) = d(n) - y(n)$, который в дальнейшем и управляет коррекцией весов и порогов нейронной сети, целью которой является приближение выходного сигнала $y(n)$ к желаемому отклику $d(n)$.

Таким образом, обучение нейронной сети сводится к минимизации сигнала ошибки $e(n)$ в пространстве весов и порогов w_k нейронов сети, а алгоритм обучения будет определять набор математических процедур, осуществляющих минимизацию сигнала ошибки и получение требуемого отклика каждого из нейронов сети при поступлении на её вход векторов обучающей выборки. Ещё одним фактором влияющим на работу сети является правильный подбор обучающей выборки, набора примеров на которых будет производиться обучение сети [2]. От того как подобраны примеры обучения будет зависеть насколько хорошо будет работать сеть после обучения. Чтобы обеспечить высокую вероятность правильного распознавания необходимо предвидеть свойства среды в которой будет работать сеть распознавания после обучения. Поскольку свойства среды носят случайный характер, то единственная правильная возможность обучения это выбирать примеры для обучения случайно и независимо [2]. Однако у этого подхода есть свой недостаток, который заключается в том, что не исключена возможность, когда для обучения будут использоваться преимущественно не типичные примеры, с которыми будет работать сеть после обучения. Это в свою очередь приведёт к тому, что наиболее часто встречаемые при работе ситуации могут неправильно распознаваться, а менее часто встречаемые будут распознаваться правильно. Следовательно, примеры должны подбираться таким образом, чтобы обеспечивать правильное распознавание ситуаций, которые наиболее часто буду встречаться сети при её работе [2]. Ошибки неправильного распознавания ситуаций встречающихся с малой вероятностью не будут критичными и обеспечат в целом высокую надёжность работы сети распознавания.

СЕТЬ РАСПОЗНАВАНИЯ ОБРАЗОВ ГЕОМЕТРИЧЕСКИХ ФИГУР

Машина распознавания образов на основе нейронной сети строится из двух частей: сети извлечения инвариантных признаков и обучаемой сети классификации (рис. 5).

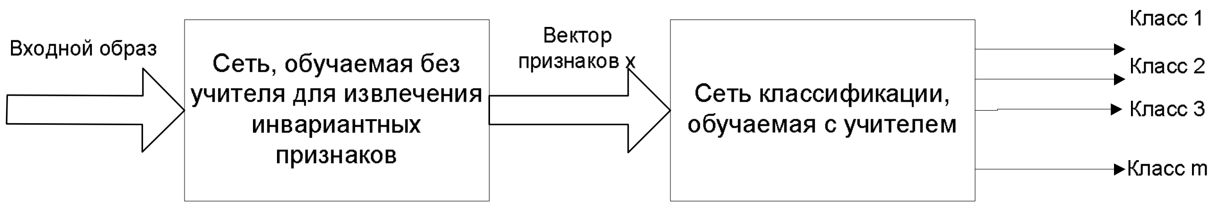

Рисунок 5– Структура сети распознавания образов

Сеть извлечения признаков может и не является нейронной сетью как таковой, а представлять собой математический и программный алгоритм, выделяющий из входного образа наиболее важные и информативные признаки объекта содержащегося во входном образе, которые инвариантны различным преобразованиям объекта в образе. Сеть извлечения признаков может быть интерпретирована как рецепторный слой сети. Виды требуемых инвариантностей выбираются исходя из конкретно решаемой задачи. Сеть классификации представляет собой однослойную или многослойную нейронную сеть обученную классифицировать ситуации, подаваемые на её вход, при этом количество нейронов выходного слоя должно совпадать с количеством ситуаций, которые обучена классифицировать сеть.

Рассмотрим задачу распознавания образов простых геометрических фигур: треугольника, квадрата, прямоугольника, окружности и эллипса.

Рисунок 6– Изображения распознаваемых геометрических фигур

Так как данные геометрические фигуры могут изменять своё положение на плоскости изображения, изменять свой размер, а также оказываться повёрнутыми на какой-либо угол, то необходимо выбрать признаки наиболее полно описывающие каждую из фигур в независимости от перечисленных изменений, т.е. признаки должны быть инвариантны к изменению масштаба фигуры, её положения и поворота на угол. Все выделяемые из исходного изображения признаки должны быть объединены в один вектор X. Перечисленным требованиям отвечает следующая система признаков:

Таблица 1 – Признаки геометрических фигур

Фигура	Признак 1	Признак 2	Признак 3	Признак 4	Признак 5	Признак 6
Треугольник	α	β	γ	0	0	0
Квадрат	α	β	γ	θ	δ	0
Прямоугольник	α	β	γ	θ	δ	0
Окружность	0	0	0	0	0	ε
Эллипс	0	0	0	0	0	ε

В табл.1 приняты следующие обозначения:

α, β, γ – углы треугольника;

$\alpha, \beta, \gamma, \theta$ – углы четырёхугольника, – отношение прилегающих сторон четырёхугольника;

ε-значение эксцентриситета кривой второго порядка, которыми являются окружность и эллипс.

Как следует из табл. 1 вектор признаков будет содержать шесть элементов $X = [x_1 x_2 x_3 x_4 x_5 x_6]$, несущих информацию о углах геометрической фигуры, отношении сторон четырёхугольника, а также о значении эксцентриситета в случае если фигура представляет собой кривую второго порядка, эллипс или окружность. Таким образом, сеть выделения признаков при подаче одной из фигур из рис.

6 на свой вход, на выходе должна предъявить вектор X с перечисленными выше шестью признаками. Алгоритм работы сети выделения признаков из табл. 1 приведен на рис. 7.

На изображении размером 640x480 пикселей содержится чёрная геометрическая фигура на белом фоне (рис. 6). Чёрный цвет фигуры позволяет определить номера строк и столбцов матрицы изображения, в которых содержится наименьшее количество чёрных элементов. Координаты крайних чёрных элементов, которые являются координатами крайних точек геометрических фигур, будут состоять из комбинации номер строки и номер столбца на пересечении которых расположен элемент чёрного цвета. Крайние точки обозначаются как a, b, c, d. В случае если координаты любых двух из четырёх точек отличаются очень незначительно или равны, то количество крайних точек становится равным трём, что соответствует треугольнику. На основе известных координат трёх точек можно определить длины отрезков их соединяющих, и далее значения углов между отрезками, воспользовавшись известными равенствами аналитической геометрии:

$$d = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}\ ; \tag{3}$$

$$\cos\alpha = \frac{(x_2 - x_1)(x_4 - x_3) + (y_2 - y_1)(y_4 - y_3)}{d_1 d_2}\ , \tag{4}$$

где – расстояние между точками с координатами (x_1, y_1) и (x_2, y_2), α-угол между отрезком длиной d_1 соединяющим точки с координатами (x_1, y_1) и (x_2, y_2) и отрезком длинной d_2 соединяющим точки с координатами (x_3, y_3) и (x_4, y_4).

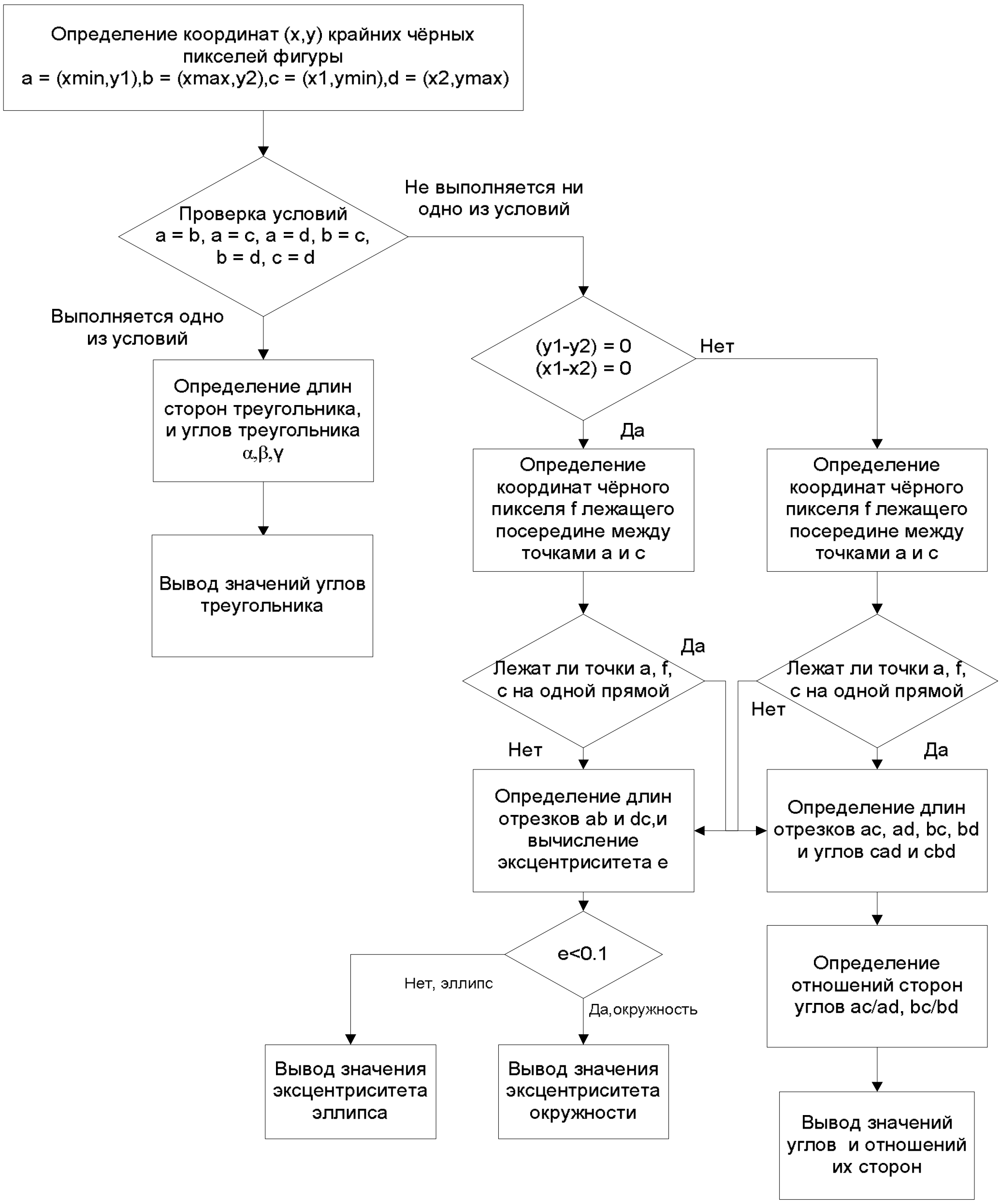

Рисунок 7– Алгоритм работы сети выделения признаков

Наличие четырёх крайних точек присуще не только четырёхугольнику, но и окружности с эллипсом. Следовательно, необходимо выполнить следующую дополнительную проверку: в случае если какие-либо две из четырёх точек лежат на одной прямой, то крайние точки принадлежат четырёхугольнику, в противном случае – кривой второго порядка. Данная проверка осуществляется вычислением определителя матрицы составленной из координат трёх точек:

$$\begin{pmatrix} x1 & y1 & 1 \\ x2 & y2 & 1 \\ x3 & y3 & 1 \end{pmatrix}.$$

В случае если определитель данной матрицы равен нулю или значению близкому к нулю, то точки с координатами (x1, y1), (x2, y2), (x3, y3) лежат на одной прямой и тогда фигура-четырёхугольник, в противном случае, если определитель больше нуля, то точки не лежат на одной прямой и принадлежат кривой второго порядка. Как следует из алгоритма приведенного на рис.7 для четырёхугольника выполняется вычисление его углов и отношения сторон одного из прямых углов. Для кривой второго порядка производится вычисление эксцентриситета ε, значение которого однозначно указывает на тип кривой второго порядка - окружность или эллипс:

$$\varepsilon = \sqrt{1 - \frac{b^2}{a^2}}\,, \tag{5}$$

гдеb – длина малой оси кривой второго порядка, a-длина большой оси кривой. Значение эксцентриситета близкое к нулю указывает на то, что кривая-окружность, в противном случае кривая-эллипс.

Сеть классификации геометрических фигур, по признакам, перечисленным в таблице 1, состоит из пяти нейронов, по количеству классов фигур, являясь, таким образом, однослойной искусственной нейронной сетью. Каждый из нейронов представляется моделью Мак-Калока (рис.1) с шестью весами$w_1, w_2, w_3, w_4, w_5, w_6$, по количеству признаков выделяемых сетью извлечения и одним пороговым весом w_0на который подаётся фиксированный единичный сигнал. Нейроны обладают логистическими пороговыми функциями активации с параметром наклона α = 1:

$$\phi(v) = \frac{1}{1 + \exp(-v)}\,,$$

где v – локальное индуцированное поле.

Как видно из графика логистической пороговой функции с единичным наклоном(рис.8), функция принимает значения из закрытого интервала [0, 1], достигая нулевого значения при значении аргумента равном - 5,а единичного при аргументе равном 5.Наличие линейных и нелинейных участков поведения функции достаточно точно отражает инерционность процессов, происходящих в реальной нервной клетке при её возбуждении. Использование данной функции удобно также с той точки зрения, что она является всюду дифференцируемой. Это свойство используется в различных алгоритмах обучения нейронных сетей.

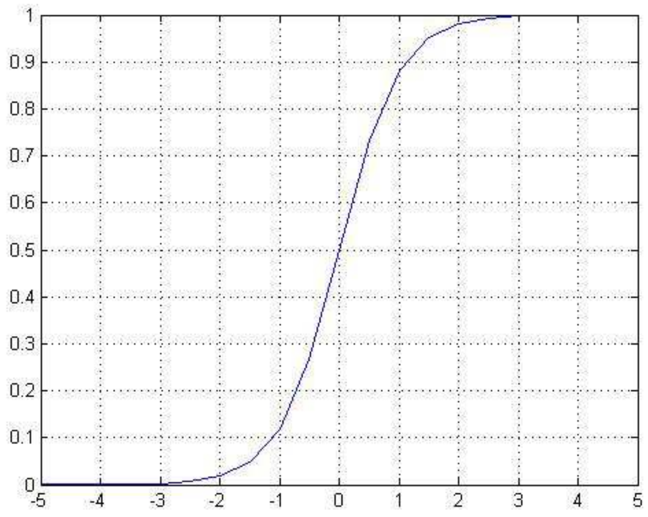

Рисунок 8– Логистическая функция с единичным наклоном

Обучение производится в несколько этапов.

1) Каждому из примеров обучения ставится в соответствие вектор $d(n)$ желаемого отклика сети, состоящий из откликов каждого из отдельных нейронов. Таким образом задаётся какой из нейронов сети должен отреагировать на данный пример и обеспечить на своём выходе отклик близкий к единице, отклики остальных нейронов должны стремится к нулю при этом.

2) Выполняется нормализация и масштабирование входного вектора признаков X, направленные на получение среднего значения элементов вектора равного нулю, а также на получение приблизительно равной ковариации входных элементов. Нормализация и масштабирование осуществляется с тем, чтобы все веса при обучении корректировались с приблизительно равной скоростью и их значения небыли бы велики, что осложнит процесс обучения. После этого этапа значения элементов входного вектора лежат в интервале [-1 1].

3) На вход однослойной сети классификации подаётся нормированный и масштабированный вектор признаков X первого примера обучения, каждый из элементов вектора подаётся на соответствующий вход каждого из нейронов выходного слоя.

4) Выбранным алгоритмом выполняется итерационная коррекция весов нейронов, для приближения значений их откликов к значению элементов вектора $d(n)$и минимизации сигналов ошибок $e(n) = d(n) - y(n)$. Далее процедура повторяется для остальных примеров обучающей выборки по этапам 2-4, пока не будут получены требуемые значения откликов вектора $d(n)$.

Рассмотрим третий и четвертый этапы более детально.

На вход сети классификации с выхода сети выделения признаков поступает вектор $X = [x_1 x_2 x_3 x_4 x_5 x_6]$, сформированный из входного образа, на котором изображен треугольник. Сеть классификации состоит из пяти нейронов, по количеству фигур которые она должна распознавать. Вектор желаемого отклика сети при подаче примера треугольника имеет вид: $d(n) = [1\,0\,0\,0\,0]$. Известный вектор признаков X и известный желаемый отклик $d(n)$ позволяют составить следующую систему условий для возбуждения только первого нейрона:

$$\begin{cases} \varphi(1\cdot w_{01} + x_1 w_{11} + x_2 w_{12} + x_3 w_{13} + x_4 w_{14} + x_5 w_{15} + x_6 w_{16}) = 1 \\ \varphi(1\cdot w_{02} + x_1 w_{21} + x_2 w_{22} + x_3 w_{23} + x_4 w_{24} + x_5 w_{25} + x_6 w_{26}) = 0 \\ \varphi(1\cdot w_{03} + x_1 w_{31} + x_2 w_{32} + x_3 w_{33} + x_4 w_{34} + x_5 w_{35} + x_6 w_{36}) = 0 \\ \varphi(1\cdot w_{04} + x_1 w_{41} + x_2 w_{42} + x_3 w_{43} + x_4 w_{44} + x_5 w_{45} + x_6 w_{46}) = 0 \\ \varphi(1\cdot w_{05} + x_1 w_{51} + x_2 w_{52} + x_3 w_{53} + x_4 w_{54} + x_5 w_{55} + x_6 w_{56}) = 0 \end{cases}, \quad (6)$$

где $\varphi(\cdot)$- операция нахождения значения логистической пороговой функции от локального индуцированного поля. Учитывая описанные выше свойства логистической функции, систему можно переписать в виде:

$$\begin{cases} 1\cdot w_{01} + x_1 w_{11} + x_2 w_{12} + x_3 w_{13} + x_4 w_{14} + x_5 w_{15} + x_6 w_{16} = -5 \\ 1\cdot w_{02} + x_1 w_{21} + x_2 w_{22} + x_3 w_{23} + x_4 w_{24} + x_5 w_{25} + x_6 w_{26} = -5 \\ 1\cdot w_{03} + x_1 w_{31} + x_2 w_{32} + x_3 w_{33} + x_4 w_{34} + x_5 w_{35} + x_6 w_{36} = -5 \\ 1\cdot w_{04} + x_1 w_{41} + x_2 w_{42} + x_3 w_{43} + x_4 w_{44} + x_5 w_{45} + x_6 w_{46} = -5 \\ 1\cdot w_{05} + x_1 w_{51} + x_2 w_{52} + x_3 w_{53} + x_4 w_{54} + x_5 w_{55} + x_6 w_{56} = -5 \end{cases}, \quad (7)$$

где $w_{01} - w_{05}$ – пороги нейронов, на которые подаются постоянные единичные воздействия,$w_{11} - w_{16}$-веса первого нейрона, $w_{21} - w_{26}$ – веса второго нейрона, $w_{31} - w_{36}$ – веса третьего нейрона, $w_{41} - w_{46}$ и $w_{51} - w_{56}$ – веса четвёртого и пятого нейронов соответственно.

Как следует из приведенных систем, обучение сети сводится к решению системы линейных алгебраических уравнений вида $A \cdot x = b$, где – известная положительно определённая матрица составленная из строк заполненных элементами вектора признаков:

$$A = \begin{pmatrix} 1\, x_1\; x_2\; x_3\; x_4\; x_5\; x_6 \\ 1\, x_1\; x_2\; x_3\; x_4\; x_5\; x_6 \\ 1\, x_1\; x_2\; x_3\; x_4\; x_5\; x_6 \\ 1\, x_1\; x_2\; x_3\; x_4\; x_5\; x_6 \\ 1\, x_1\; x_2\; x_3\; x_4\; x_5\; x_6 \end{pmatrix},$$

b – вектор столбец, составленный из желаемых откликов для данного примера обучения:

$$b = \begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix},$$

x – искомая матрица весов и порогов нейронной сети:

$$x = \begin{pmatrix} w_{01} & w_{11} & w_{12} & w_{13} & w_{14} & w_{15} & w_{16} \\ w_{02} & w_{21} & w_{22} & w_{23} & w_{24} & w_{25} & w_{26} \\ w_{03} & w_{31} & w_{32} & w_{33} & w_{34} & w_{35} & w_{36} \\ w_{04} & w_{41} & w_{42} & w_{43} & w_{44} & w_{45} & w_{46} \\ w_{05} & w_{51} & w_{52} & w_{53} & w_{54} & w_{55} & w_{56} \end{pmatrix},$$

Нахождение искомой матрицы весов и порогов x из приведенной системы уравнений$A \cdot x - b = 0$, эквивалентно решению следующей задачи оптимизации [3]:

$$F(x) = \frac{1}{2} Ax, x - b, x \to \inf, x \in X \ , \tag{8}$$

$$F'(x) = A \cdot x - b = 0,$$

где $F(x)$ – целевая функция заданная на множестве X весов и порогов, минимум которой находится в точке $x = x_*$. Данную задачу можно решить воспользовавшись одним из эффективных численных методов оптимизации – методом сопряжённых градиентов [3]. Идея метода сопряженных градиентов состоит в следующем. Пусть задан базис $\{p_k\}_{k=1}^{n}$ на множестве X. Тогда для любой точки $x_0 \in X$ вектор $x_* - x_0$ можно разложить по базису [3]:

$$x_* - x_0 = \alpha_1 p_1 + \alpha_2 p_2 + \ldots + \alpha_n p_n;$$

$$x_* = x_0 + \alpha_1 p_1 + \alpha_2 p_2 + \ldots + \alpha_n p_n.$$

На каждой следующей итерации приближение вычисляется по выражению:

$$x_k = x_0 + \alpha_1 p_1 + \alpha_2 p_2 + \ldots + \alpha_n p_k \ . \tag{9}$$

В качестве начального приближения x_0 выбирается произвольный вектор весов и нейронов. На каждой итерации α_kвыбирается по правилу:

$$\alpha_k = argmin_{\alpha_k} F(x_{k-1} + \alpha_k p_k). \tag{10}$$

Базисные вектора $\{p_k\}$ вычисляются по выражениям [3]:

$$p_1 = -F'(x_0):$$

$$p_{k+1} = -F'(x_k) + \beta_k p_k. \tag{11}$$

Коэффициенты β_kвыбираются таким образом, чтобы векторы p_k и p_{k+1} были сопряжены относительно матрицы А:

$$\beta_k = \frac{F'(x_k), Ap_k}{Ap_k, p_k}. \tag{12}$$

Данный метод оптимизации эффективен и более устойчив для квадратных матриц А. Для получения квадратной матрицы из не квадратной, при обучении достаточно добавить дополнительно один или несколько примеров обученияс соответствующим добавлением элементов в вектор желаемых откликов $d(n)$, что приведёт к равенству количеств уравнений в системе с количеством неизвестных в каждом из уравнений системы. Основным достоинством метода сопряженных градиентов является то, что он решает квадратичную задачу оптимизации за конечное число шагов. Решив все пять задач оптимизации для примеров пятираспознаваемых фигур, приведём результаты обучения, веса и пороги, искусственной нейронной сети распознавания образов.

Таблица 2–Значения порогов и весов обученной сети распознавания

Пороги и веса	Треугольник	Квадрат	Прямоугольник	Окружность	Эллипс
w_{k0}	–4,99	–4,9899	–4,9899	6,14298	–6,14298
w_{k1}	7,23892	56,9185	–56,9185	–8,44221	0,87435
w_{k2}	7,23892	56,9185	–56,9185	–8,44221	0,87435
w_{k3}	2,06273	–116,953	116,277	–1,54699	0,16022
w_{k4}	–16,81	25,693	–14,51	6,2951	–0,651936
w_{k5}	2,06273	2,06273	116,277	–1,54699	0,16022
w_{k6}	$-2{,}0768\cdot 10^{-6}$	$9{,}4934\ 10^{-6}$	$7{,}79302\cdot 10^{-6}$	–17,9564	17,9564

Приведём также графические зависимости относительных норм ошибок, невязок, от номеров итераций для каждого класса примеров, за которые они были достигнуты, при заданной точности метода сопряженных градиентов равной $1\cdot 10^{-5}$.

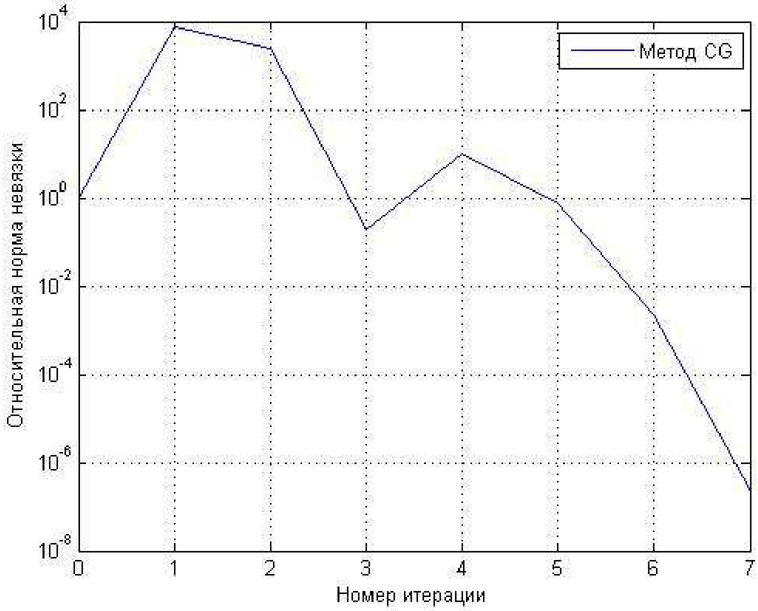

Рисунок 9 – Зависимость относительной нормы ошибки от номера итерации при обучении распознаванию треугольника

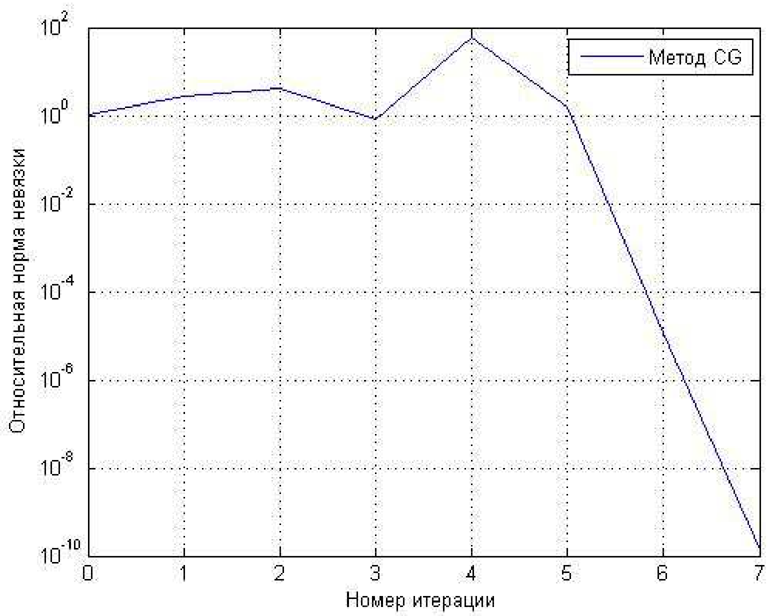

Рисунок 10 – Зависимость относительной нормы ошибки от номера итерации при обучении распознаванию квадрата

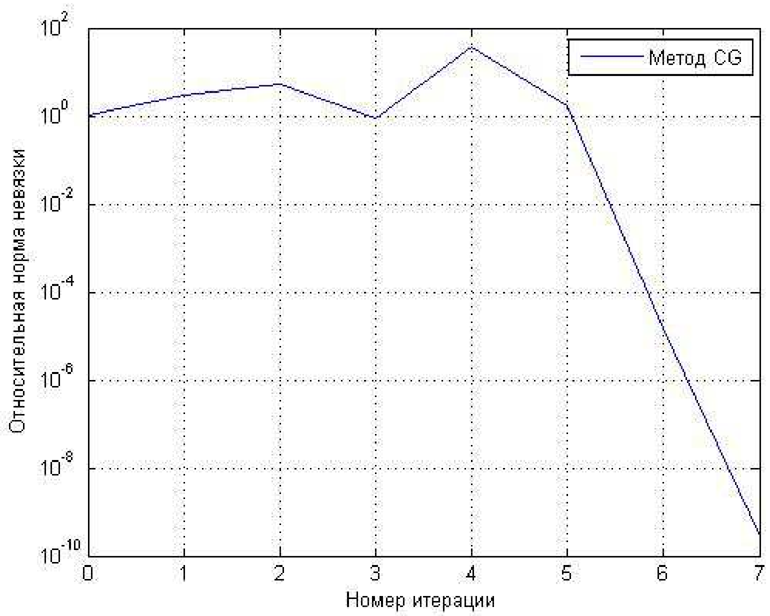

Рисунок 11 – Зависимость относительной нормы ошибки от номера итерации при обучении распознаванию прямоугольника

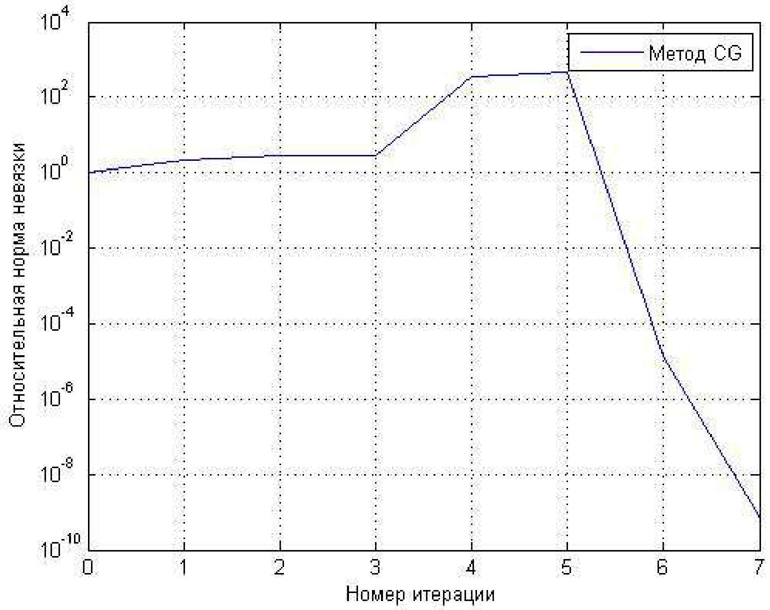

Рисунок 12 – Зависимость относительной нормы ошибки от номера итерации при обучении распознаванию окружности

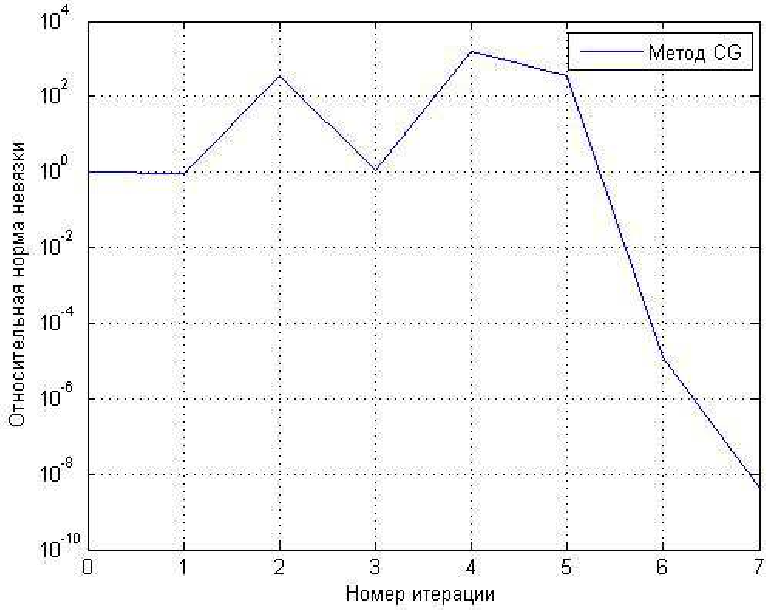

Рисунок 13 – Зависимость относительной нормы ошибки от номера итерации при обучении распознаванию эллипса

Как следует из приведенных графических зависимостей применение численного метода сопряженных градиентов для обучения нейронной сети, позволяет достичь заданной точности за 7 итераций для каждого из примеров обучения, что в десятки и сотни раз меньше, чем при использовании классических алгоритмов обучения нейронных сетей ,например метода обратного распространения ошибки и его модификаций , а также сократить тем самым количество операций умножения требуемых для них, уменьшить вычислительную мощность и как следует время затрачиваемое на обучения.

ВЫВОДЫ

В данной работе освещены основные сведения о построении и обучении нейронных сетей распознаванию образов. На примере сети распознавания образов простейших геометрических фигур показаны принципы построения и алгоритма работы сети выделения инвариантных признаков распознаваемых объектов, а также продемонстрировано применение численного метода оптимизации, метода сопряженных градиентов, для обучения однослойной нейронной сети классификации, состоящей из пяти нейронов. Полученные результаты демонстрируют преимущество применения численных методов оптимизаций для решении задачи обучения распознаванию образов, что отражается в значительном сокращении необходимых количеств математических операций, и достижении требуемых результатов обучения за конечное количество итераций.

ЛИТЕРАТУРА

1 Хайкин Саймон. Нейронные сети: полный курс /. – [2-е издание]: Издательский дом «Вильямс», 2006.– 1104 с.: ил. – Парал. тит. англ.

2 Вапник В.Н. Теория распознавания образов (статистические проблемы обучения), / В.Н. Вапник, А.Я. Червоненкис. – М.: Издательство Наука, Главная редакция физико-математической литературы, 1974. – 416 с.

3 Васильев Ф.П. Численные методы решения экстремальных задач: учеб. пособ. для вузов / Ф.П. Васильев. – [2-е изд. перераб. и доп.] . – М.: Наука. Гл. ред. физ.-мат. лит.,1988. – 552 с.