Министерство образования и науки Российской Федерации

Нижегородский государственный университет им. Н.И. Лобачевского

Т. И. Чачхиани
М. Г. Серова

Алгоритм перцептрона

Практикум

Рекомендовано методической комиссией факультета ВМК для студентов ННГУ, обучающихся по направлениям подготовки 010300 «Фундаментальная информатика и информационные технологии» и 010400 «Прикладная математика и информатика»

Нижний Новгород
2015

УДК 519.92
ББК 22.18
Ч-28

Чачхиани Т.И., Серова М.Г. Алгоритм перцептрона: практикум. – Нижний Новгород: Нижегородский госуниверситет, 2015. – 25 с.

Рецензент: к.ф.-м. н., доцент **Кузенков О. А**.

Практикум входит в учебно-исследовательский комплекс лабораторных работ для курса «Распознавание образов». В нем содержится описание перцептрона и алгоритмов его обучения, приводится доказательство сходимости. Дается описание работы программного модуля. Практикум предназначается для студентов ННГУ, обучающихся по направлениям подготовки 010300 «Фундаментальная информатика и информационные технологии» и 010400 «Прикладная математика и информатика».

Ответственный за выпуск:
заместитель председателя методической комиссии
факультета ВМК ННГУ,
к. т. н., доцент **В.М. Сморкалова**

УДК 519.92
ББК 22.18

Введение

В жизни человека и животных обучение играет огромную роль. Основные навыки и представление об окружающем мире человек приобретает в процессе обучения и восприятия, в процессе общения с окружающей средой. При этом большое место занимает накопление и осмысление опыта.

Происхождение алгоритмов классификации образов, рассматриваемых в данной работе, было связано с попытками глобального осмысления деятельности мозга.

Американский ученый Ф.Розенблатт в 1957 году предложил перцептронный («узнающий») алгоритм [1], который представляет собой одну из первых моделей процессов запоминания и организации информации, реализуемых мозгом.

Перцептрон Розенблатта произвел ошеломляющее впечатление на современников. Он впервые указал на реальные возможности алгоритмизации интеллектуальной деятельности и привел к созданию нового направления исследований, получившего название «Распознавание образов» [2,3,4].

Целью данного практикума является ознакомление с алгоритмом перцептрона.

1. Перцептронный подход

В основе перцептрона лежит некоторая последовательность преобразования зрительного восприятия букв, цифр или геометрических фигур, проектируемых на сетчатку, состоящую из сенсорных элементов (фотоэлементов), до выходного сигнала реагирующего элемента. При этом осуществляется некоторая процедура его обучения, основанная на поощрении и наказании.

Рассмотрим основную модель перцептрона, обеспечивающую отнесение объекта к одному из двух заданных классов (рис 1). Сенсорные элементы сетчатки S случайным образом связаны с ассоциативными элементами второй сетчатки А. Каждый элемент второй сетчатки воспроизводит сигнал только в том случае, если достаточное число сенсорных элементов, соединенных с его входом, находится в возбужденном состоянии. Сенсорные элементы можно рассматривать в качестве устройств, с помощью которых вся система воспринимает из внешней среды объект, т.е. как некие измерительные устройства, а ассоциативные элементы – как входную часть системы. Реакция всей системы пропорциональна сумме, взятых с определенными весами, реакций элементов ассоциативной сетчатки. Таким образом, обозначив через x_i

реакцию i-го ассоциативного элемента и через w_i соответствующий вес, реакцию системы можно записать как

$$\mathrm{R} = \sum_{i=1}^{n+1} w_i x_i = \mathrm{w}'\mathrm{x} \qquad \textbf{(1)}$$

Выходные сигналы x_i элементов сетчатки A можно рассматривать как признаки распознаваемых объектов. При этом во все векторы признаков введена единица после последней компоненты, т.е. $\mathrm{x}=(x_1, x_2, \ldots, x_n, 1)'$ - пополненный вектор признаков, а $\mathrm{w}=(\mathrm{w}_1, \mathrm{w}_2, \ldots, \mathrm{w}_n, \mathrm{w}_{n+1})'$ - пополненный вектор весов.

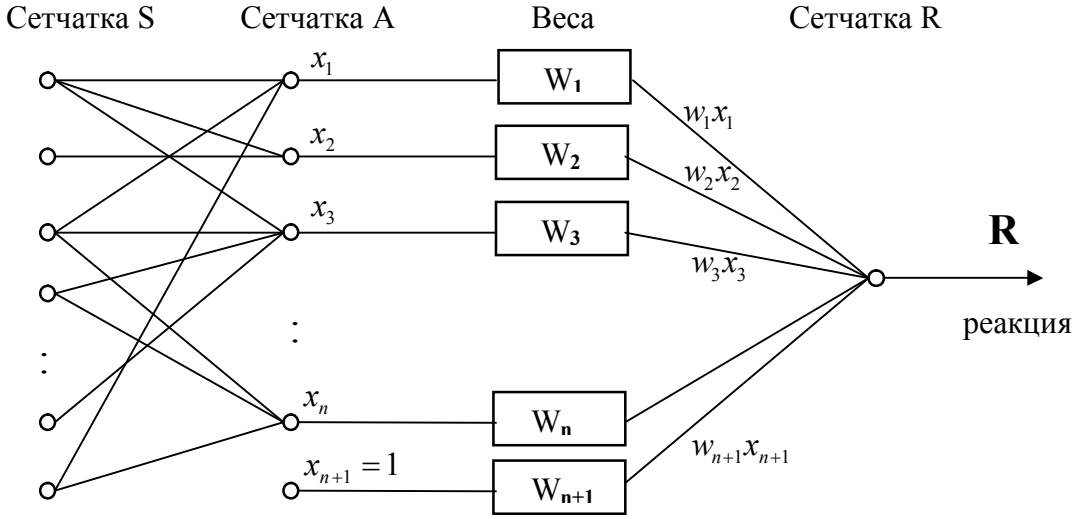

Рис 1. Основная модель перцептрона для двух классов

Поскольку одна и та же величина вводится в описание всех объектов, основные геометрические свойства классов не затрагиваются.

Процедура классификации происходит следующим образом. Если R>0, значит предъявленный системе образ принадлежит классу ω_1, если R<0, то образ относится к классу ω_2.

Можно видеть, что основная перцептронная модель представляет собой, за исключением сенсорной сетчатки, не что иное, как реализацию линейной решающей функции $\mathrm{d}(x) = \mathrm{w}'\mathrm{x}$. Схему на рис.1 легко распространить на случай разделения на несколько классов $\omega_1, \omega_2, .., \omega_м$ (рис.2).

В случае M классов необходимо в схему перцептрона ввести M блоков весов и M элементов в сетчатку R. Выходы элементов R-сетчатки

$$\mathrm{R}_i = \sum_{j=1}^{n} \mathrm{w}_j^{(i)} \mathrm{x}_j \qquad (2)$$

подаются на блок выбора $\max \mathrm{R}_i$. Объект x причисляется к классу ω_i, если $\mathrm{R}_i > \mathrm{R}_j$, для всех $i \neq j$. Основную модель можно также легко распространить на случай нелинейных решающих функций введением M соответствующих

нелинейных преобразователей после сетчатки A. Однако анализ можно без полной потери общности полностью ограничить линейными решающими функциями, т.к. нелинейные решающие функции можно рассматривать как линейные в расширенном пространстве признаков.

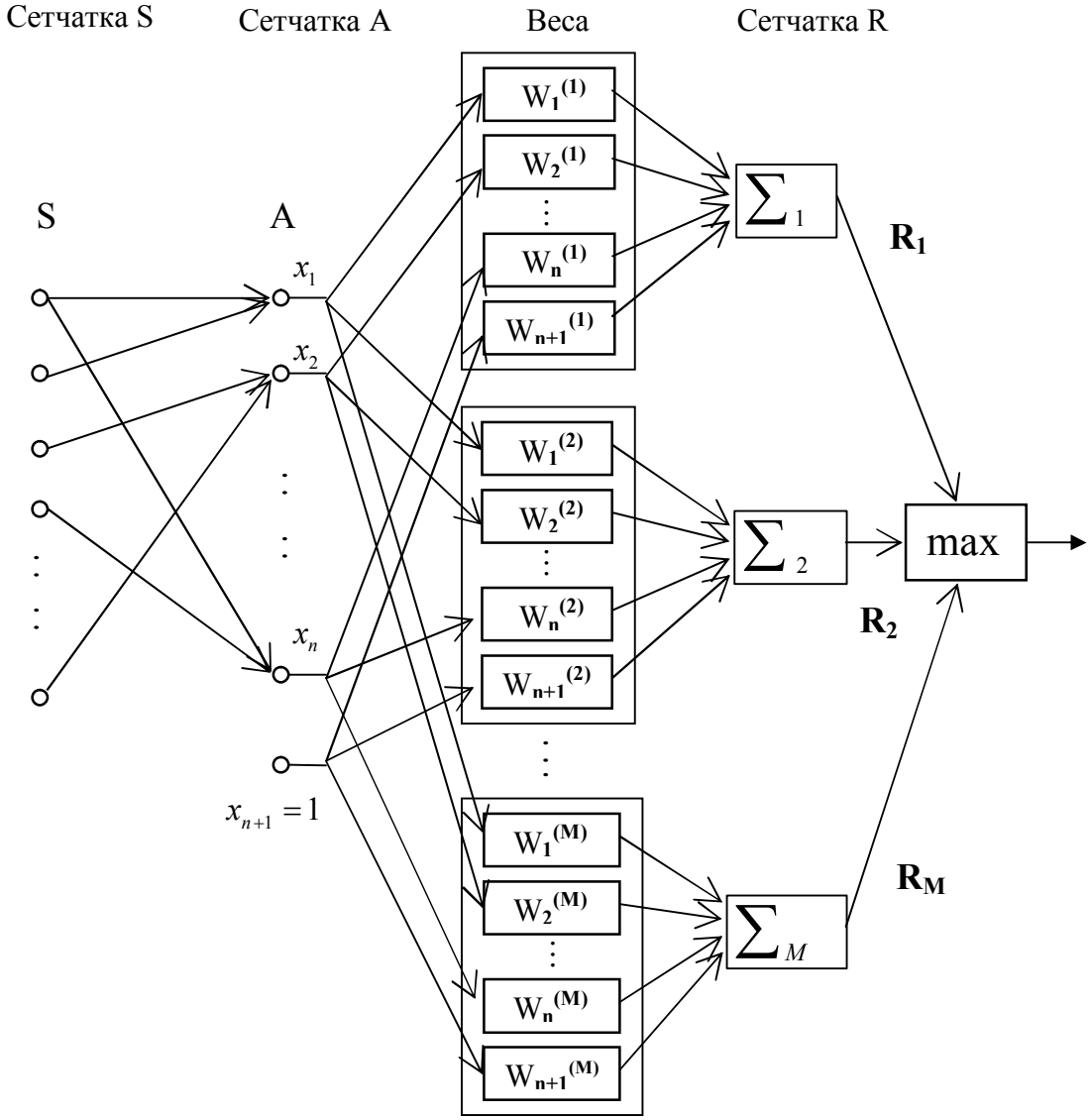

Рис. 2. Модель перцептрона для M классов

1.1 Принцип подкрепления-наказания

Обучающий алгоритм для перцептрона, приведенного на рис. 1, сводится к простой схеме итеративного определения вектора весов w. Дадим краткое описание этой схемы, которую называют алгоритмом перцептрона.

Заданы два обучающих множества объектов, представляющие классы ω_1 и ω_2 соответственно; пусть w(1) - начальный вектор весов, который выбирается произвольно, в этом случае *k*-ый шаг обучения выглядит следующим образом.

Если x(k) принадлежит ω_1 и $w'(k)\,x(k) \le 0$, то

$$w(k+1)= w(k) + C\, x(k)\ ,$$

где C – положительное число, называемое корректирующим множителем.

Если $x(k)$ принадлежит ω_2 и $w'(k)\, x(k) \geq 0$, то

$$w\,(k+1)= w(k) - C\, x(k)\ .$$

В противном случае $w(k)$ не изменяется, т.е.

$$w(k+1)= w(k).$$

Таким образом, вектор весов w изменяется только в том случае, если объект, предъявленный на *k*-ом шаге обучения, был классифицирован неправильно.

Очевидно, что алгоритм перцептрона является процедурой типа «подкрепления-наказания». Если объект классифицирован правильно, то система подкрепляется тем, что вектор весов не изменяется. С другой стороны, если объект классифицирован неправильно, то система наказывается изменением вектора весов w.

Сходимость алгоритма наступает при правильной классификации всех объектов из обучающих множеств.

1.2. Пространство признаков и пространство весов

Решающая функция в случае разбиения на два класса должна обладать следующим свойством: для всех объектов одного класса должно выполняться неравенство $d(x) > 0$ и для объектов второго класса - неравенство $d(x) < 0$. Допустим, что в каждый класс входят по два двумерных объекта $\{x_1^1, x_2^1\}$ и $\{x_1^2, x_2^2\}$, где верхние индексы обозначают классы ω_1 и ω_2 соответственно. Если классы линейно разделимы, задача сводится к отысканию вектора $w =(w_1, w_2, w_3)'$, для которого справедливы следующие неравенства:

$$\begin{aligned} &w_1 x_{11}^1 + w_2 x_{12}^1 + w_3 > 0,\\ &w_1 x_{21}^1 + w_2 x_{22}^1 + w_3 > 0,\\ &w_1 x_{11}^2 + w_2 x_{12}^2 + w_3 < 0,\\ &w_1 x_{21}^2 + w_2 x_{22}^2 + w_3 < 0. \end{aligned} \qquad (3)$$

Другими словами, вектор $\overline{w}$ является решением системы линейных неравенств, определяемого всеми объектами, входящими в состав обоих классов.

Умножив пополненные образы, принадлежащие классу ω_2 на -1, систему неравенств (2) можно переписать в виде

$$\begin{aligned} &w_1 x_{11}^1 + w_2 x_{12}^1 + w_3 > 0,\\ &w_1 x_{21}^1 + w_2 x_{22}^1 + w_3 > 0, \end{aligned} \qquad (4)$$

$$-w_1x_{11}^2 - w_2x_{12}^2 - w_3 > 0,$$
$$-w_1x_{21}^2 - w_2x_{22}^2 - w_3 > 0.$$

В этом случае задача сводится к отысканию вектора w, обеспечивающего положительность всех неравенств. Неравенства (3) и (4) требуют только, чтобы компоненты векторы w определяли разделяющую границу для классов ω_1 и ω_2. Для того чтобы получить более полное представление о геометрических свойствах вектора решений w, целесообразно обсудить различия между понятиями пространства признаков и пространства весов.

Пространство признаков представляет собой n-мерное евклидово пространство, содержащее вектора объектов, как это показано на рисунке 3,а. Координатные переменные обозначены x_1, x_2,…, x_n. В данном пространстве вектор w представляет набор коэффициентов, определяющих разделяющую поверхность.

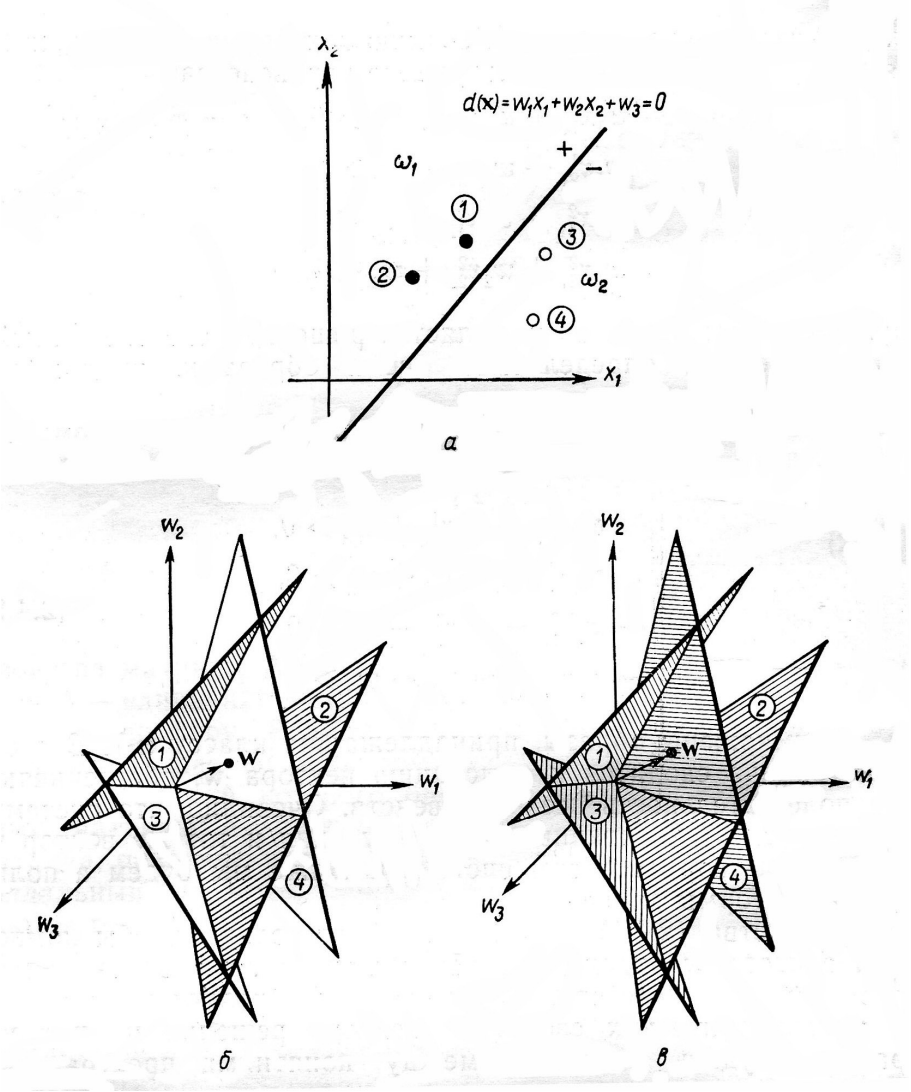

Рис. 3. а- пространство признаков; б - пространство весов для системы (3);
в - пространство весов для системы (4)

Пространство весов представляет собой (n+1) – мерное евклидово пространство, где $w_1, w_2, \ldots, w_{n+1}$ - координатные переменные. В этом пространстве каждое неравенство соответствует положительной или отрицательной зоне гиперплоскости, проходящей через начало координат. Каждому объекту соответствует своя гиперплоскость. Решение системы неравенств (3) является всякий вектор w, расположенный в положительных зонах всех плоскостей объектов класса ω_1 и в отрицательных зонах всех плоскостей объектов класса ω_2. Решением системы (4) служит всякий вектор w, расположенный в положительных зонах всех плоскостей объектов. Оба случая представлены на рисунках 3.б и 3.в, где заключенными в кружочки цифрами обозначены объекты и соответствующие им плоскости, находящиеся в пространстве весов. Штриховкой отмечены положительные стороны плоскостей объектов. Отметим, что в общих случаях вектор решений один и тот же, причем область решения ограничена конической поверхностью. В общем случае поверхность границы представляет собой выпуклый многогранный конус. Это конус решений систем (3) и (4).

Пусть $x_1, x_2, \ldots, x_n$ представляют обучающее множество пополненных объектов, относящихся к двум заданным классам, причем все объекты, принадлежащие классу ω_2, умножены на -1. тогда алгоритм перцептрона можно записать как

$$w(k+1) = \begin{cases} w(k), & \text{если } w'(k)\,x(k) > T, \\ w(k) + cx(k), & \text{если } w'(k)\,x(k) \le T, \end{cases} \qquad (5)$$

где C – положительный корректирующий множитель.

Таким образом, алгоритм обучения перцептрона состоит в том, чтобы найти такой вектор весов w*, при котором для всех объектов из классов ω_1 и ω_2 будет выполняться

$$w * x_i > 0, \qquad i = 1,2,\ldots N. \qquad (6)$$

Из записи алгоритма (5) видно, что в пространстве весов коррекция вектора w заключается в перемещении в сторону плоскости объекта точки, соответствующей вектору w. Причем величину шага коррекции определяет корректирующий множитель C: весовая точка может не дойти до плоскости объекта, попасть на нее или перейти на другую сторону плоскости объекта. В наиболее общей форме алгоритм обучения перцептрона можно определить как последовательное испытание множества плоскостей объектов. Для каждой такой плоскости выясняется, лежит ли весовая точка с положительной стороны. Если точка имеет правильное расположение, то переходят к рассмотрению следующей плоскости объекта из обучающего множества. Если же это расположение неправильное, то весовая точка перемещается перпендикулярно по направлению к ней (например, на другую сторону), а затем переходит к испытанию следующего объекта. Процесс продолжается до тех пор, пока

весовая точка не будет находиться с положительной стороны всех плоскостей объектов. Это соответствует тому, что весовой вектор придет в конус решений.

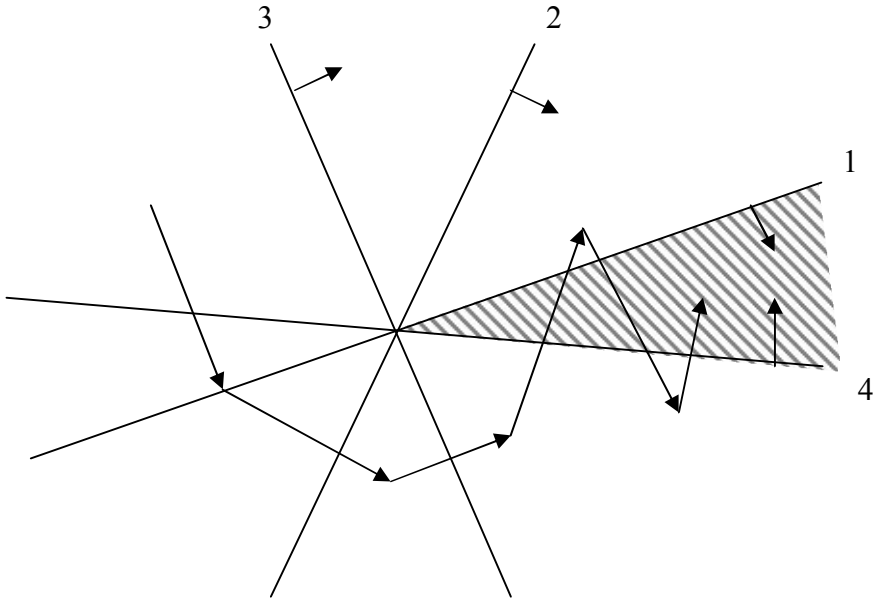

Рис. 4. Коррекция весового вектора

На рис. 4 показана геометрическая интерпретация коррекции весового вектора. Имеется 4 объекта, которым соответствуют плоскости в пространстве весов. Маленькие стрелки на плоскостях показывают положительную сторону. Объекты предъявляются циклически в следующем порядке 1-2-3-4, 1-2-3-4, … . В кружочках на плоскости стоят номера этих объектов. Требуется, чтобы весовая точка пришла в область решений (заштрихована). В данном простом примере это достигается после пяти исправлений весового вектора. Заметим, что исправление, сделанное для некоторого объекта, может, вообще говоря, полностью ликвидировать, сделанное для предыдущего объекта. Однако, в итоге будет сделано последнее исправление, которое оставит правильным распознавание других объектов.

1.3. Алгоритмы обучения перцептрона

Варьируя способ выбора корректирующего множителя C можно получить несколько модификаций алгоритма перцептрона. К наиболее распространенным алгоритмам обучения перцептрона относятся алгоритм фиксированного приращения, алгоритм абсолютной коррекции, алгоритм дробной коррекции.

В алгоритме фиксированного приращения корректирующий множитель C является константой, больше нуля. В частности, когда C = 1, каждый вес меняется с помощью прибавления соответствующей компоненты объекта.

Такое изменение веса исправляет или не исправляет ошибку, произошедшую на данном объекте, в зависимости от отношения величины $w'x$ и C.

В алгоритме абсолютной коррекции C выбирается достаточно большим, для того чтобы гарантировать правильную классификацию объекта после коррекции весов. Другими словами, если $w'(k)x(k) \le 0$, то C выбирается таким образом, чтобы

$$w'(k+1)\,x(k) = [\,w'(k) + C\,x(k)]'\,x(k) > 0$$

При этом в качестве C выбирается наименьшее целое число, превышающее

$$\frac{|w'(k)\,x(k)|}{x'(k)\,x(k)}.$$

В алгоритме дробной коррекции C выбирается так, чтобы величина $|w'(k)x(k) - w'(k+1)x(k)|$ была положительна и составляла некоторую долю r от величины $|w'(k)\,x(k)|$, т.е.

$$|w'(k)\,x(k) - w'(k+1)\,x(k)| = r\,|w'(k)\,x(k)|, \qquad (7)$$

где r есть положительная константа.

Подстановка $w(k+1) = w(k) + Cx(k)$ в (7) дает

$$C = r\,\frac{|w'(k)\,x(k)|}{x'(k)\,x(k)}.$$

Этот алгоритм, очевидно, требует, чтобы начальный вектор весов отличался от нулевого вектора. Дробная величина r представляет собой отношение расстояния, разделяющего прежний вектор весов w(k) и новый вектор весов w(k+1), к нормальному евклидову расстоянию от вектора весов w(k) до гиперплоскости объектов в пространстве весов. Если r >1, то образ классифицируется правильно после каждой коррекции весов. Можно показать, что при $0 < r < 2$ этот алгоритм сходится.

1.4. Доказательство сходимости

Пусть x_1, x_2, …, x_n представляют обучающее множество образов, относящихся к двум заданным классам, причем все образы, принадлежащие классу ω_2, умножены на -1. Постулируется, что в случае линейной разделимости классов алгоритм обучения

$$w(k+1) = \begin{cases} w(k), & \text{если } w'(k)\,x(k) > 0, \\ w(k) + cx(k), & \text{если } w'(k)\,x(k) \le 0, \end{cases} \qquad (8)$$

обеспечивает определение весового вектора решения w*, отличающегося тем, что

$$w^{*\prime}\,x_i > 0, \quad i = 1, 2, \ldots, N. \qquad (9)$$

Выражение (9) можно представить в несколько более общем виде, введя неотрицательную пороговую величину T, такую, что при линейной разделимости классов

$$w^{*\prime}\, x_i > T,\ \ i = 1,2,\ldots,N. \qquad (10)$$

При этих условиях алгоритм (8) принимает следующий вид:

$$w(k+1) = \begin{cases} w(k), & \text{если } w'(k)\, x(k) > T, \\ w(k) + x(k), & \text{если } w'(k)\, x(k) \le T, \end{cases} \qquad (11)$$

причем вектор $w(1)$ выбирается произвольным образом.

Пусть для простоты $C = 1$. это допущение не нарушает общности рассуждений, так как любое другое значение C может быть введено в векторы образов в качестве нормирующей константы. Из геометрического анализа пространства образов (рис.3.а) следует, что пороговая величина T создает с обеих сторон гиперплоскости $w'(k)\, x(k) = 0$ буферные области. Всякий образ, попадающий в эти области, классифицируется неправильно. Обращаясь к рис.4, следует отметить, что непосредственным результатом увеличения пороговой величины T является уменьшение объема конуса решений.

Предполагая возможность предъявления каждого образа необходимое количество раз, утверждается, что при линейной разделимости заданных классов алгоритм, представленный выражением (11), приведет за конечное число шагов к получению искомого результата. Доказательство существенно упроститься, если помимо применения введенных выше обозначений принимать во внимание только те индексы k, которым соответствует неправильная классификация образов. Тогда, изменив снова запись индексов, можно прийти к выражениям

$$w(k+1) = w(k) + x(k), \qquad (12)$$

и

$$w'(k)x(k) \le T, \qquad (13)$$

для всех значений индекса k в обучающей последовательности, при которых происходит коррекция.

Сходимость алгоритма на самом деле означает, что после некоторого конечного значения индекса k_m имеет место равенство

$$w(k_m) = w(k_m + 1) = w(k_m + 2) = \ldots .$$

После введения этих упрощений доказательство сходимости алгоритма состоит в следующем. Из (11) получаем

$$w(k+1) = w(1) + x_i(1) + x_i(2) + \ldots + x_i(k). \qquad (14)$$

Скалярное произведение вектора w^* с обеими частями уравнения (14) дает

$$w'(k+1)\, w^* = w'(1)w^* + x_i'(1)w^* + x_i'(2)w^* + \ldots + x_i'(k)\, w^*. \qquad (15)$$

Так как из условия (9) следует, что каждый член $x_i'(j)w^*$, $j = 1, \ldots, k$, больше пороговой величины T, то

$$w'(k+1)\, w^* \ge w'(1)\, w^* + kT. \qquad (16)$$

Неравенство Коши – Шварца ($\|a\|^2\|b\|^2 \ge (a'b)^2$) приводит к выражению

$$[\mathrm{w}'(k+1)\,\mathrm{w}^*]^2 \le \|\mathrm{w}'(k+1)\|^2\,\|\mathrm{w}^*\|^2, \qquad (17)$$

где $\|\mathrm{a}\|^2$ обозначает квадрат модуля вектора a.

Неравенство (17) можно переписать в виде

$$\left\|\mathrm{w}(k+1)\right\|^2 \ge \frac{\left[\mathrm{w}'(1)\mathrm{w}^*\right]^2}{\left\|\mathrm{w}^*\right\|^2}. \qquad (18)$$

После подстановки неравенства (16) в (18) получим неравенство

$$\left\|\mathrm{w}(k+1)\right\|^2 \ge \frac{\left[\mathrm{w}'(1)\mathrm{w}^* + kT\right]^2}{\left\|\mathrm{w}^*\right\|^2}. \qquad (19)$$

Другая ветвь рассуждений приводит к противоречию, касающемуся величины $\|\mathrm{w}(k+1)\|^2$. Из (11) заключаем, что

$$\|\mathrm{w}(j+1)\|^2 = \|\mathrm{w}(j)\|^2 + 2\mathrm{w}'(j)\mathrm{x}(j) + \|\mathrm{x}(j)\|^2 \qquad (20)$$

или

$$\|\mathrm{w}(j+1)\|^2 - \|\mathrm{w}(j)\|^2 = 2\mathrm{w}'(j) + \|\mathrm{x}(j)\|^2 \qquad (21)$$

Используя неравенство (12) и полагая $\mathrm{Q} = \max\|\mathrm{x}_i(j)\|^2$, придем к

$$\|\mathrm{w}(j+1)\|^2 - \|\,\mathrm{w}(j)\|^2 \le 2T + \mathrm{Q} \qquad (22)$$

Суммируя эти неравенства по всем $j = 1, 2, \ldots, k$, получим

$$\|\mathrm{w}(k+1)\|^2 \le \|\,\mathrm{w}(1)\|^2 + (2T + \mathrm{Q})\,k\,. \qquad (23)$$

Сопоставление неравенств (18) и (22) показывает, что при достаточно больших значениях k границы, устанавливаемые для величины $\|\mathrm{w}(k+1)\|^2$ соответствующими неравенствами, противоречат друг другу. Действительно, индекс k не может принимать значений, больших значения k_m, удовлетворяющего уравнению

$$\frac{\left[\mathrm{w}'(k+1)\mathrm{w}^* + kT\right]^2}{\left\|\mathrm{w}^*\right\|^2} = \|\,\mathrm{w}(1)\,\|^2 + (2T + \mathrm{Q})\,k_m. \qquad (24)$$

Согласно (23), k_m – конечная величина, из чего следует сходимость алгоритма перцептрона за конечное число шагов при условии линейной разделимости заданных классов. Это завершает доказательство сходимости алгоритма перцептрона.

1.5. Замечания

Частный случай при $T = 0$ доказывается несколько иначе. Неравенство (15) принимает вид

$$\mathrm{w}'(k+1)\,\mathrm{w}^* \ge \mathrm{w}'(1)\,\mathrm{w}^* + ka, \qquad (25)$$

где

$$a = \min\,[\mathrm{x}_i'(j)\,\mathrm{w}^*]. \qquad (26)$$

Так как согласно нашей гипотезе w* - вектор решения, то $a > 0$. Кроме того, поскольку $\mathrm{w}'(j)\mathrm{x}_i(j) \le 0$, неравенство (21) превращается в

$$\|w(j+1)\|^2 - \|w(j)\|^2 \le \|x_i(j)\|^2 \le Q. \qquad (27)$$

Остальная часть доказательства остается неизменной. Число шагов алгоритма, необходимое для его сходимости при $T = 0$, задается решением уравнения

$$\frac{\left[w'(1)w^* + k_m a\right]^2}{\|w^*\|^2} = \| w(1) \|^2 + Q k_m. \qquad (28)$$

Сходимость алгоритма перцептрона можно доказать целым рядом способов. Доказательство, приведенное выше, является одним из наиболее четких. Алгоритм перцептрона и его модификации сходятся в тех случаях, когда заданные классы можно разделить поверхностью выбранного типа. В тех случаях, когда разделимость отсутствует, эти алгоритмы зацикливаются и работают в таком режиме, пока их выполнение не прерывается извне. Поскольку при наличии разделимости невозможно заранее рассчитать число шагов, необходимое для сходимости алгоритма, редко можно с абсолютной уверенностью судить о том, означает или нет наличие длинной обучающей последовательности отсутствие линейной разделимости классов.

2. Описание работы программного модуля

Для представления работы алгоритма перцептрона разработан программный модуль «Персептрон».

После запуска программы на экране появляется главное окно программы (рис.5).

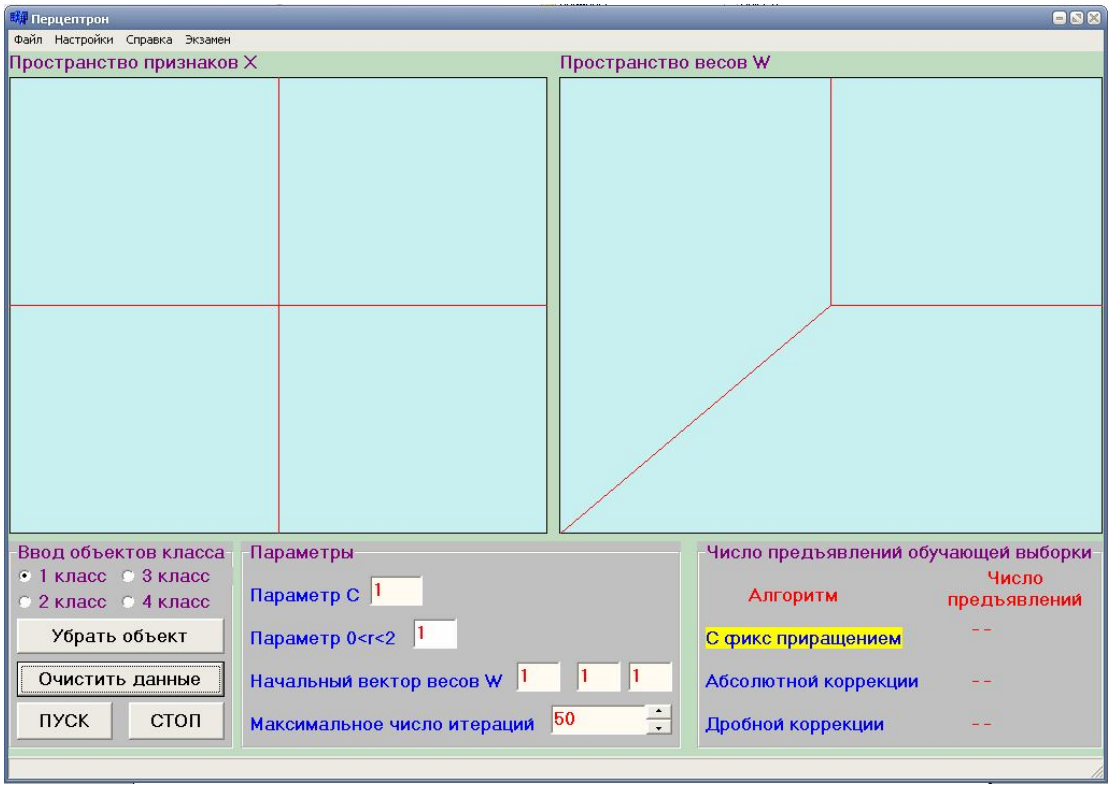

Рис. 5. Главное окно программы

В окне представлены две графических области: область пространства признаков X и область пространства весов W; а также блок, управляющий работой алгоритма, блок параметров и информационный блок.

2.1. Настройка работы алгоритма перцептрона

Перед началом работы алгоритма следует выбрать вид алгоритма коррекции (рис.6) и метод демонстрации (рис.7) с помощью соответствующих подпунктов пункта меню «Настройки».

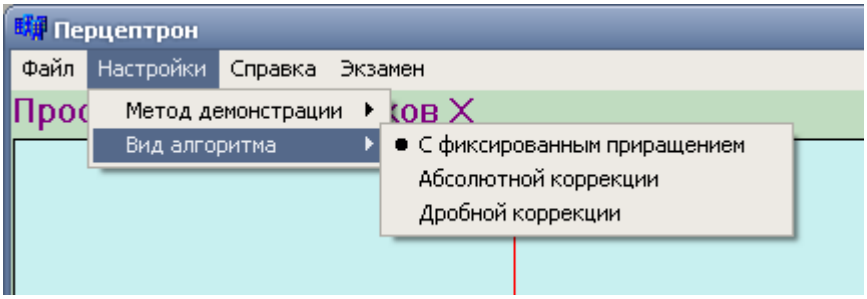

Рис. 6. Выбор алгоритма коррекции.

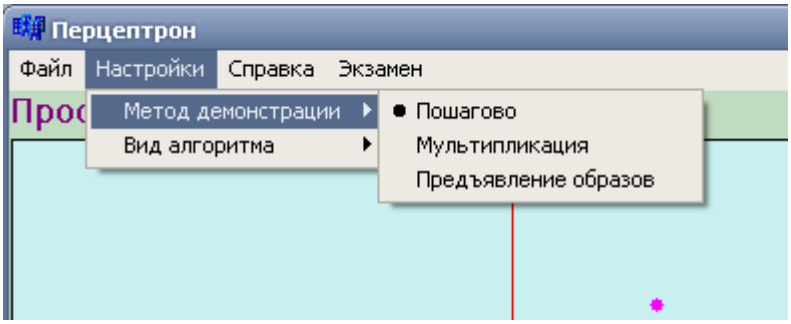

Рис. 7. Выбор метода демонстрации.

В режиме «Пошагово» после одного нажатия кнопки «ПУСК» можно видеть изменения полей пространств после одной итерации (предъявления всех объектов). В режиме «Предъявление образов» в этом случае можно видеть состояние полей пространств после предъявления одного объекта.

В режиме «Мультипликация» предъявление каждого объекта и переход на другую итерацию происходит автоматически с некоторой задержкой. Остановка происходит при завершении обучения или превышения количества итераций.

Другие параметры (c, r, начальный вектор весов, максимальное количество итераций) можно задать в области параметров в нижней части окна (или оставить значения по умолчанию).

2.2. Ввод объектов

Для пользователя подготовлены примеры обучающих выборок от 2-х до 4-х классов. Чтобы загрузить один из примеров, нужно воспользоваться пунктом меню «Файл – Открыть» и выбрать файл примера из одной из папок в папке «Примеры» (рис. 8).

После загрузки примера на пространстве признаков X появятся точки различной окраски. Точки, соответствующие объектам 1-го класса имеют красный цвет, точки 2-го класса – синий, 3-го – зеленый, 4-го – коричневый.

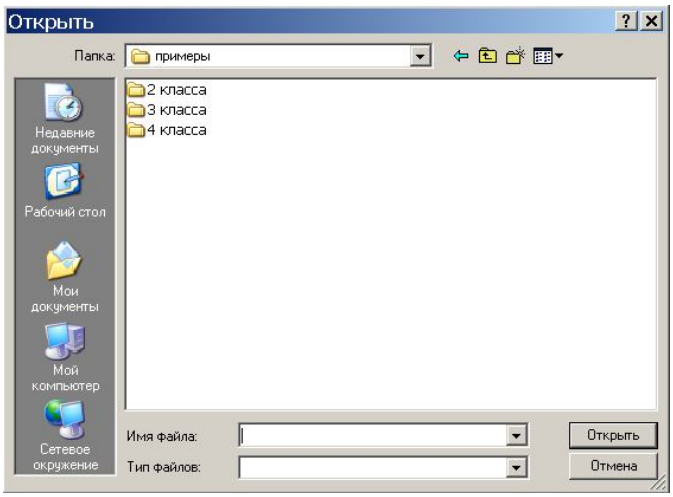

Рис. 8. Выбор файла примера

Ввод объектов классов можно выполнить также вручную мышкой в области пространства признаков после нажатия кнопки: «1 класс» для ввода объектов первого класса, «2 класс» - второго класса, и так далее. Для корректной работы алгоритма при вводе объектов (точек) нужно соблюдать следующие правила: точки должны быть визуально линейно разделимыми на плоскости пространства признаков и эта разделяющая классы прямая линия обязана проходить через точку пересечения осей (начало координат).

Стереть ранее введенные объекты классов можно после нажатия кнопки «Очистить» с помощью клика мыши над нужной точкой.

2.3. Работа алгоритма

После ввода настроек, параметров и объектов классов следует запустить работу алгоритма при помощи кнопки «Пуск».

В режиме демонстрации «пошагово» алгоритм будет останавливаться и ждать следующего нажатия кнопки «Пуск» после каждой итерации. В режиме мультипликации алгоритм выполняется без участия пользователя до его завершения. При этом в области пространства признаков на каждой итерации будет рисоваться разделяющая классы прямая, а в области пространства весов –

траектория движения вектора весов (рис. 9). Можно остановить работу программы, нажав кнопку «СТОП».

После окончания работы алгоритма в информационной области в правой нижней части окна будет представлено число предъявлений обучающей выборки напротив выбранного алгоритма коррекции.

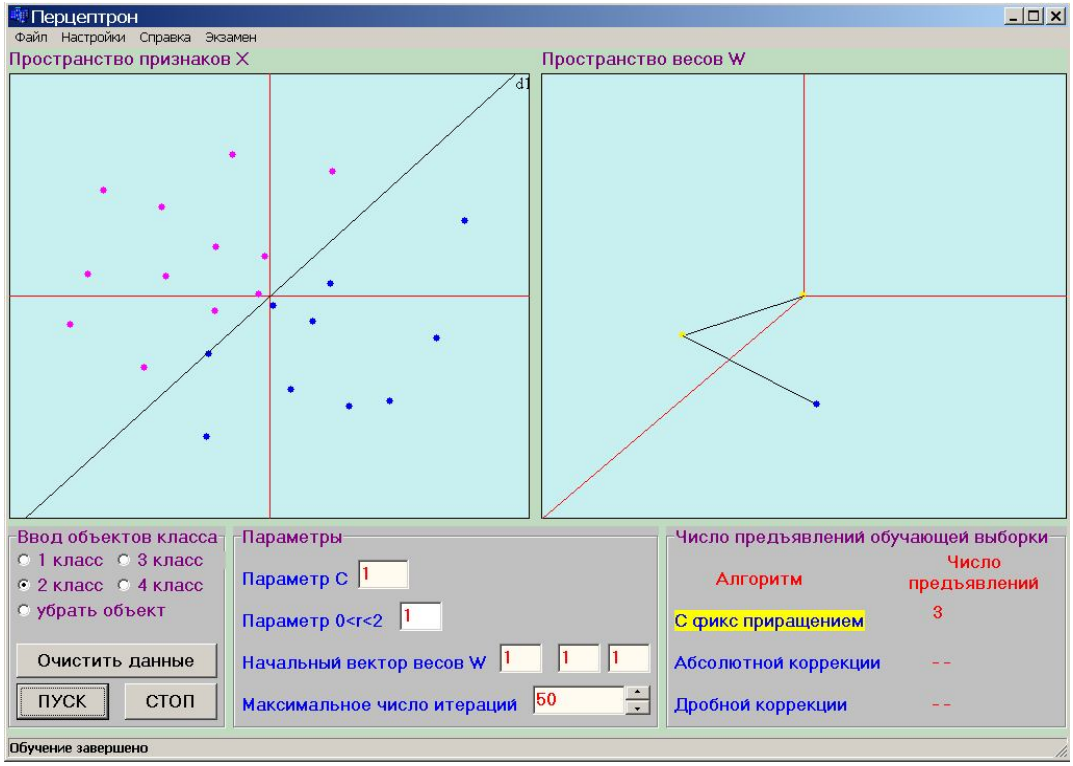

Рис. 9. Результат работы алгоритма

2.4. Экзамен

Для полученной решающей функции следует провести экзамен. Для этого нужно выбрать пункт меню «Экзамен» и следовать инструкции в правой части окна.

2.5. Справка

В пункте меню «Справка» пользователь найдет описание алгоритма перцептрона и подробное руководство по работе с программой.

2.6. Примеры

Рассмотрим сначала примеры для двух классов.

Пример 1. Алгоритм фиксированного приращения для двух классов

Откроем 1-й пример из папки «2 класса» (рис. 8). На поле пространства признаков появляется 6 объектов, по 3 в каждом классе (рис. 10).

Сначала выберем через пункт меню «Настройки - Вид алгоритма» алгоритм «С фиксированным приращением». Выбранный алгоритм будет отмечен в поле «Число предъявлений обучающей выборки» (рис. 10).

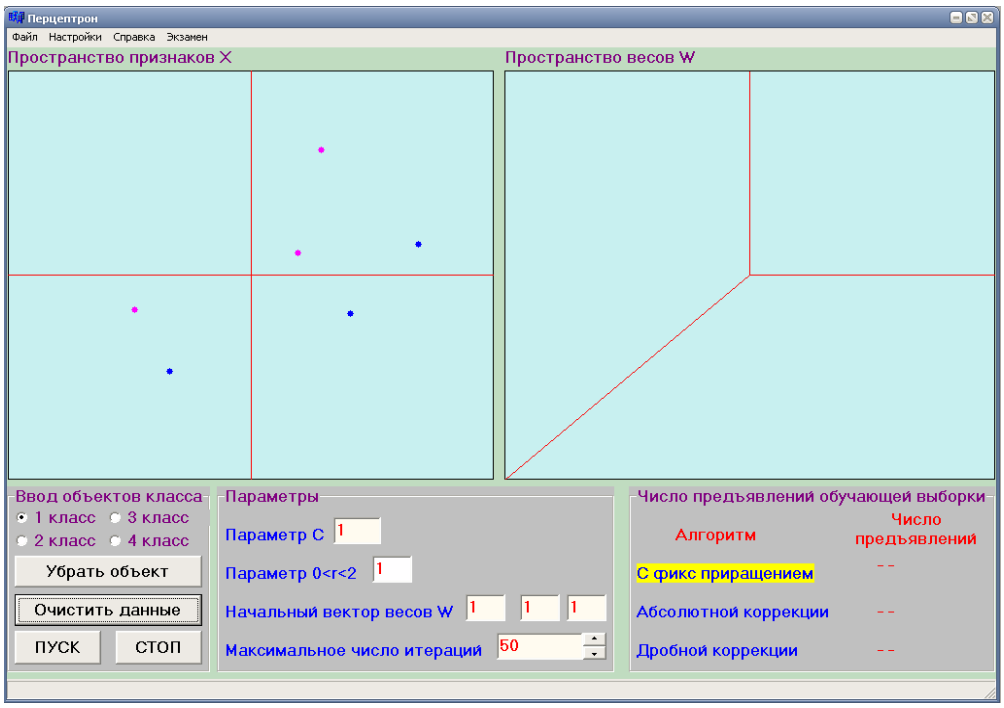

Рис. 10. Обучающая выборка из примера для 2-х классов

Далее можно изменить параметры демонстрации работы алгоритма с помощью пункта меню «Настройки - Метод демонстрации». С целью подробного рассмотрения выберем метод «Предъявление образов».

Для запуска программы обучения перцептрона нажмем кнопку ПУСК.

Рассмотрим изменения после запуска: в поле «Пространство признаков X» отобразился первый объект и текущее решающее правило (рис. 11).

Соответственно, в поле «Пространство весов W» отобразилось новое положение вектора весов в конусе решений.

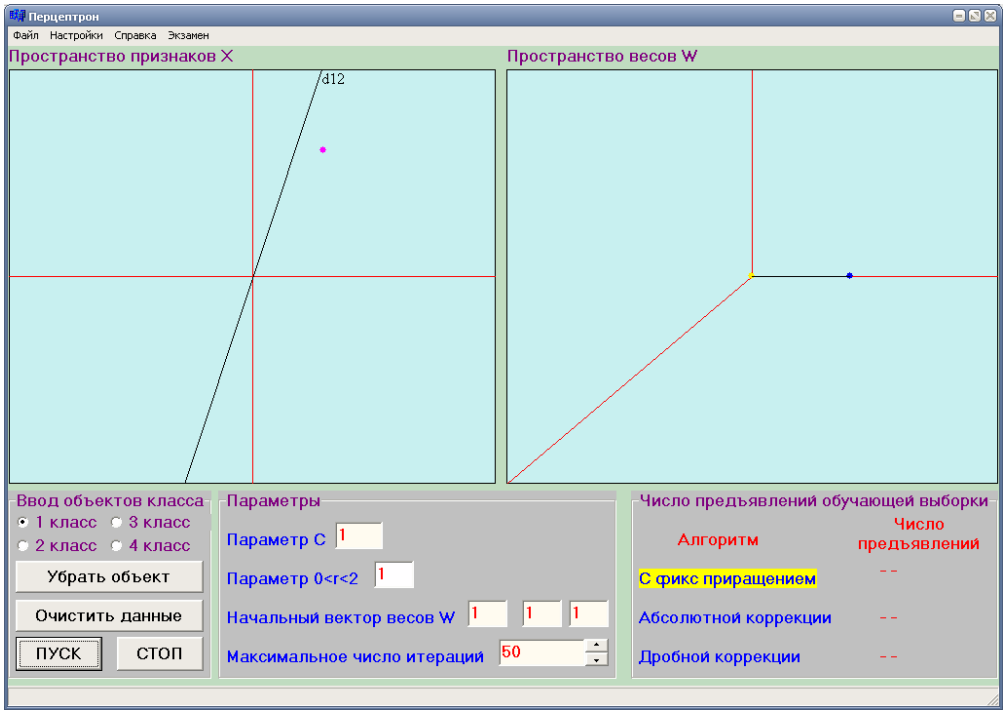

Рис. 11. Предъявление первого образа в примере для двух классов

Рассмотрим изменения после предъявления всех объектов: увеличилось число предъявлений в поле «Число предъявлений обучающей выборки» (рис.12). Так как при предъявлении объектов была сделана коррекция, то перцептрон на данном этапе еще не обучен, и в строчке внизу окна выведена подсказка о коррекции.

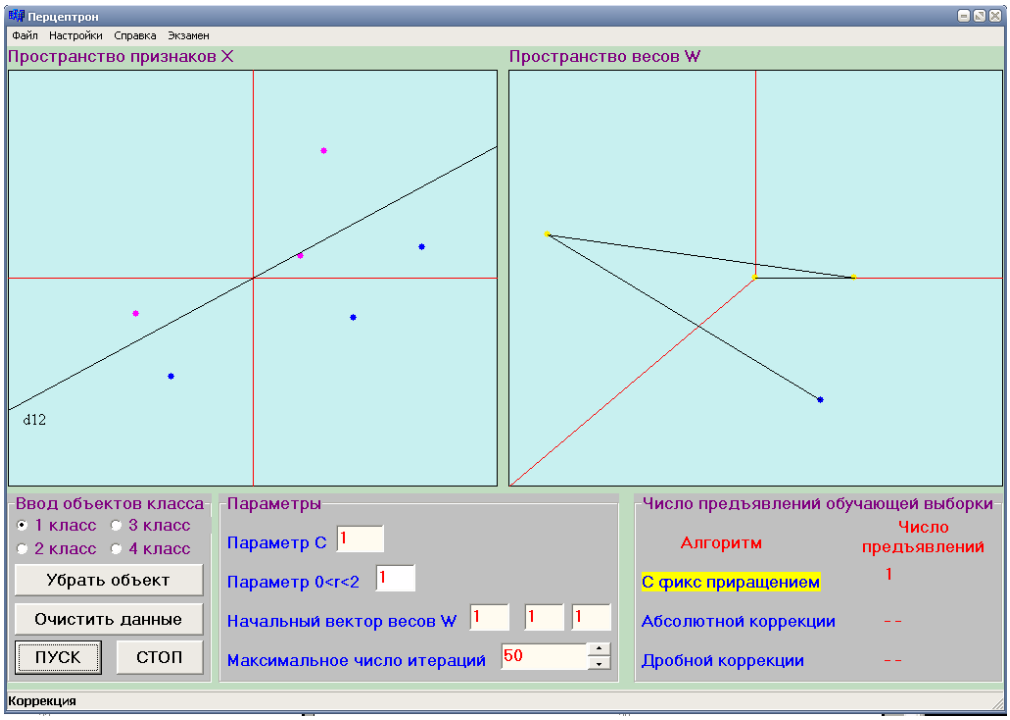

Рис. 12. Первая итерация в примере для двух классов

Продолжаем работу перцептрона, пока не появится сообщение об окончании обучения. В поле «Пространство признаков X» при этом отобразится обучающая выборка и решающая функция для обученного перцептрона (рис. 13).

В поле «Пространство весов W» отобразится положение вектора весов, пришедшего в конус решений.

Внизу окна можно увидеть сообщение о завершении обучения. Перцептрон обучился за 7 итераций. Информация об этом фиксируется в области «Число предъявлений обучающей выборки» для сравнения результатов работы различных видов алгоритмов.

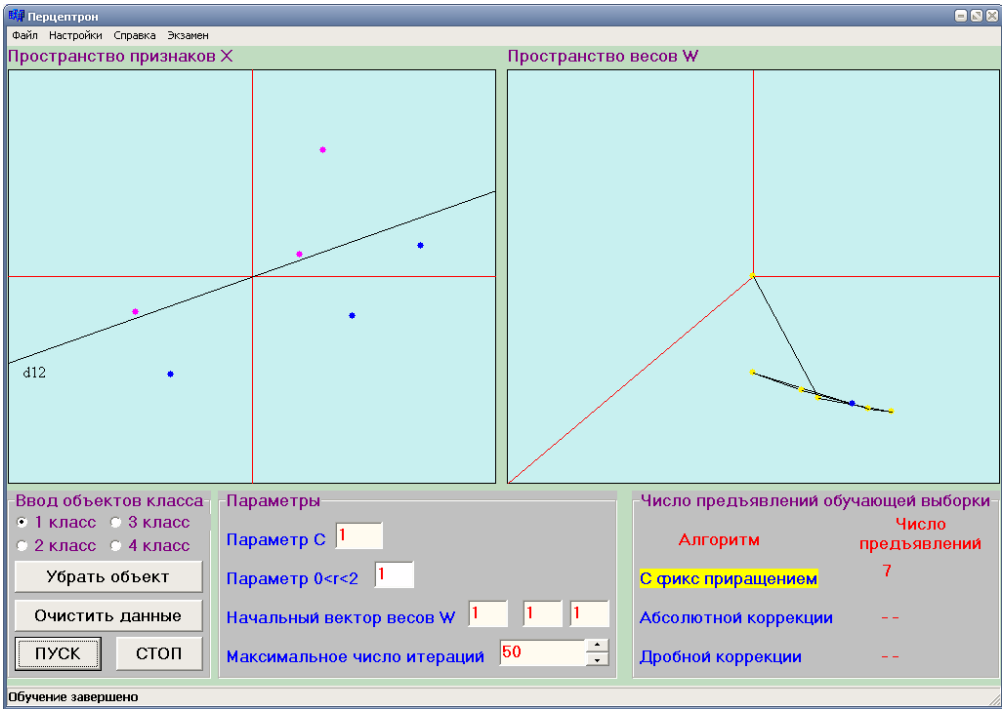

Рис. 13. Обучение завершено с алгоритмом фиксированного приращения

Пример 2. Алгоритм абсолютной коррекции для двух классов

Изменим вид алгоритма на алгоритм абсолютной коррекции и рассмотрим за сколько итераций обучится перцептрон (рис. 14). Количество итераций увеличилось до 8-ми.

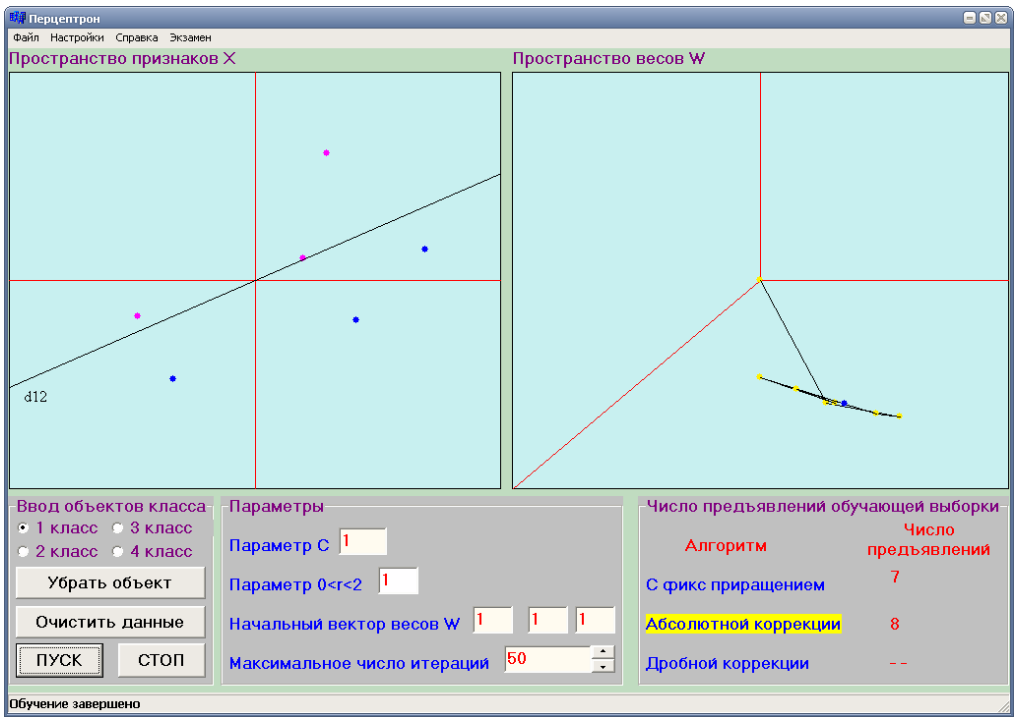

Рис. 14. Обучение завершено с алгоритмом абсолютной коррекции

Пример 3. Алгоритм дробной коррекции для двух классов

Изменим вид алгоритма на алгоритм дробной коррекции. Для исследования возможностей этого алгоритма сначала установим параметр r = 0,1 (рис. 15).

Рис. 15. Изменение параметра r

В ходе работы алгоритма появилось сообщение о превышении максимального числа итераций (рис. 16). Перцептрон не смог обучиться за установленные 50 итераций.

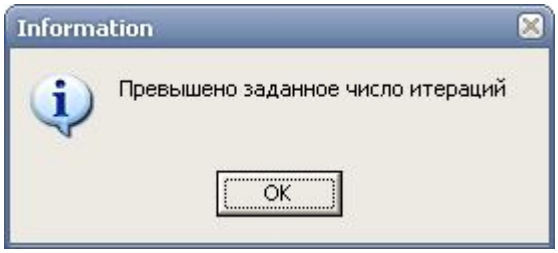

Рис. 16. Сообщение

Превышение числа итераций отображено в поле «Число предъявлений обучающей выборки» напротив выбранного алгоритма и в строке внизу формы (рис 17).

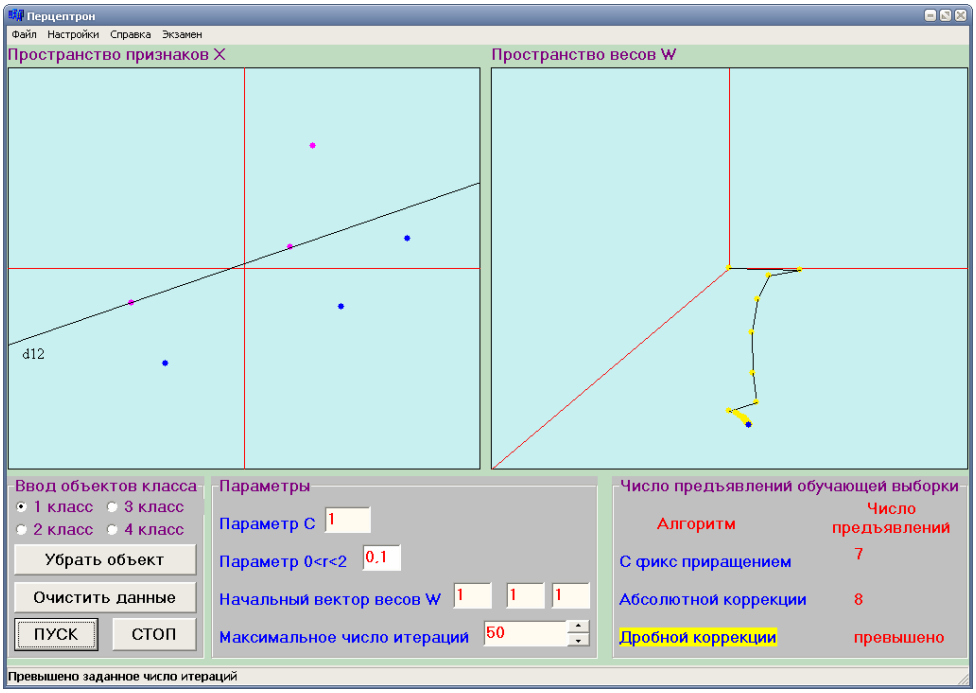

Рис. 17. Окно программы при превышении числа итераций

Для того чтоб перцептрон обучился, следует увеличить максимальное число итераций в поле «Параметры». Или, в случае алгоритма дробной коррекции, можно изменить значение параметра r в сторону увеличения, например, до величины «1,3». После такой коррекции параметра r перцептрон обучится за 3 итерации (рис. 18).

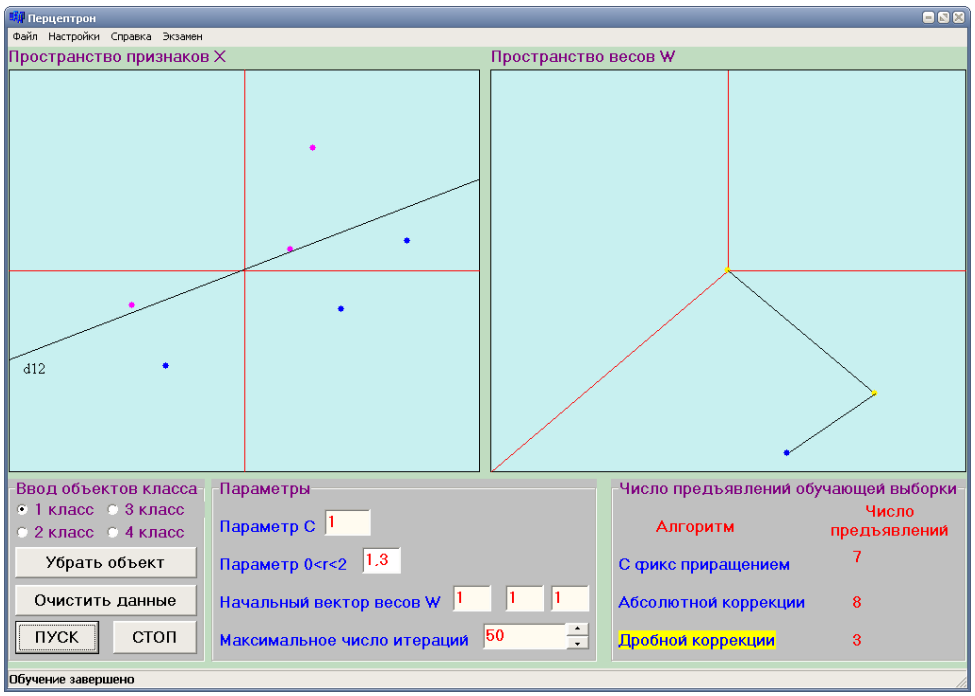

Рис. 18. Обучение с алгоритмом дробной коррекции при параметре r =1,3

Пример 4. Ввод объектов вручную

Для ввода собственного примера нужно воспользоваться областью «Ввод объектов класса» (рис. 19).

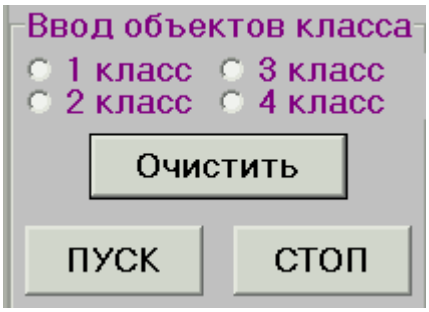

Рис. 19. Поле «Ввод объектов класса»

Поставим отметку в поле класса, который нужно ввести, и нанесем точки на поле «Пространство признаков X» с помощью левой кнопки мыши. В данном примере введем объекты 1-го и 2-го классов.

Для того чтобы ввести другой пример нужно нажать кнопку «Очистить» и ввести новые объекты классов.

После того как выбраны метод демонстрации и вид алгоритма перцептрона, можно запустить алгоритм и проанализировать результаты его работы (рис 20).

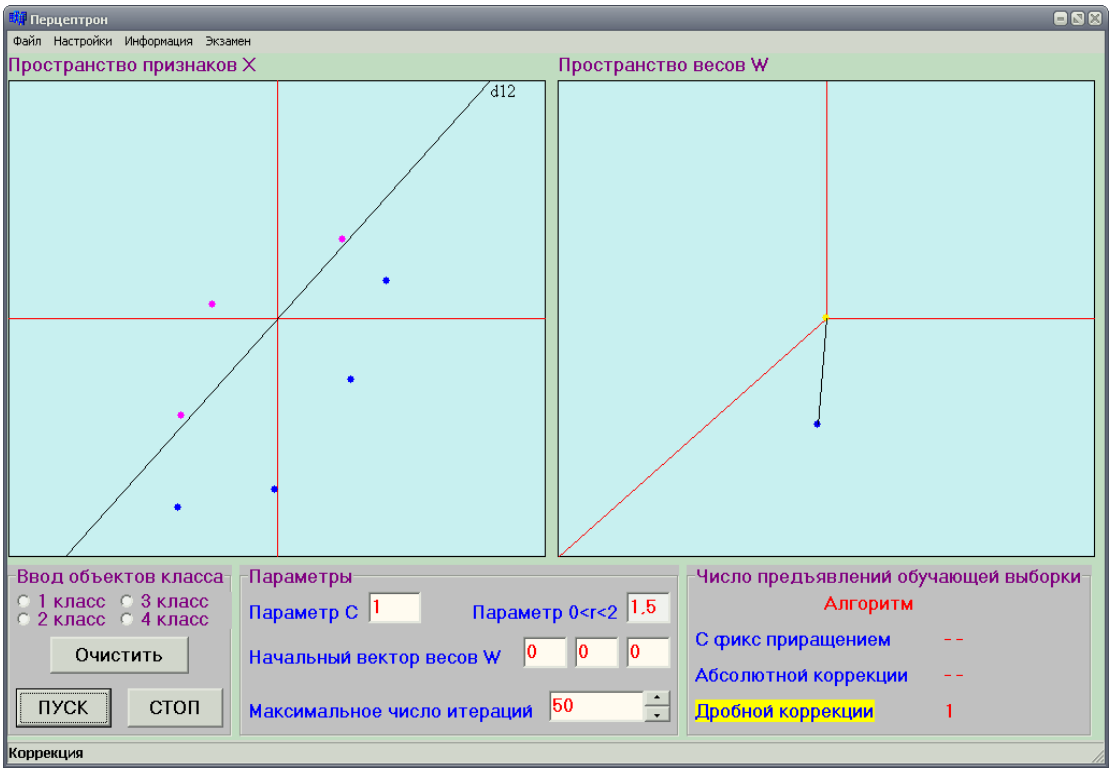

Рис. 20. Первая итерация для примера 4

В поле «Пространство признаков» отображена полученная решающая функция. В поле «Пространство признаков» синей точкой обозначено новое значение вектора весов, желтыми точками - предыдущие значения. После предъявления всей выборки количество итераций отобразится в поле «Число предъявлений обучающей выборки».

Пример 5. Больше двух классов

Если вручную введено больше двух классов, то после нажатия кнопки «Пуск» будут представлены следующие построения (рис. 21). В области «Пространства признаков X» цветными линиями обозначены границы между классами, проведенные для классов попарно. В поле «Результат» можно увидеть, как перцептрон разделил пространство признаков на области, принадлежащие различным классам.

Если перцептрон обучился, то появится сообщение «Обучение завершено» и мы получим решающую функцию для наших классов и положение вектора весов в конусе решений.

Если обучение не удалось выполнить за данное количество итераций, то появится сообщение «Превышено заданное число итераций». Тогда возможны два варианта дальнейшей работы: или необходимо изменить параметры, или классы линейно не разделимы.

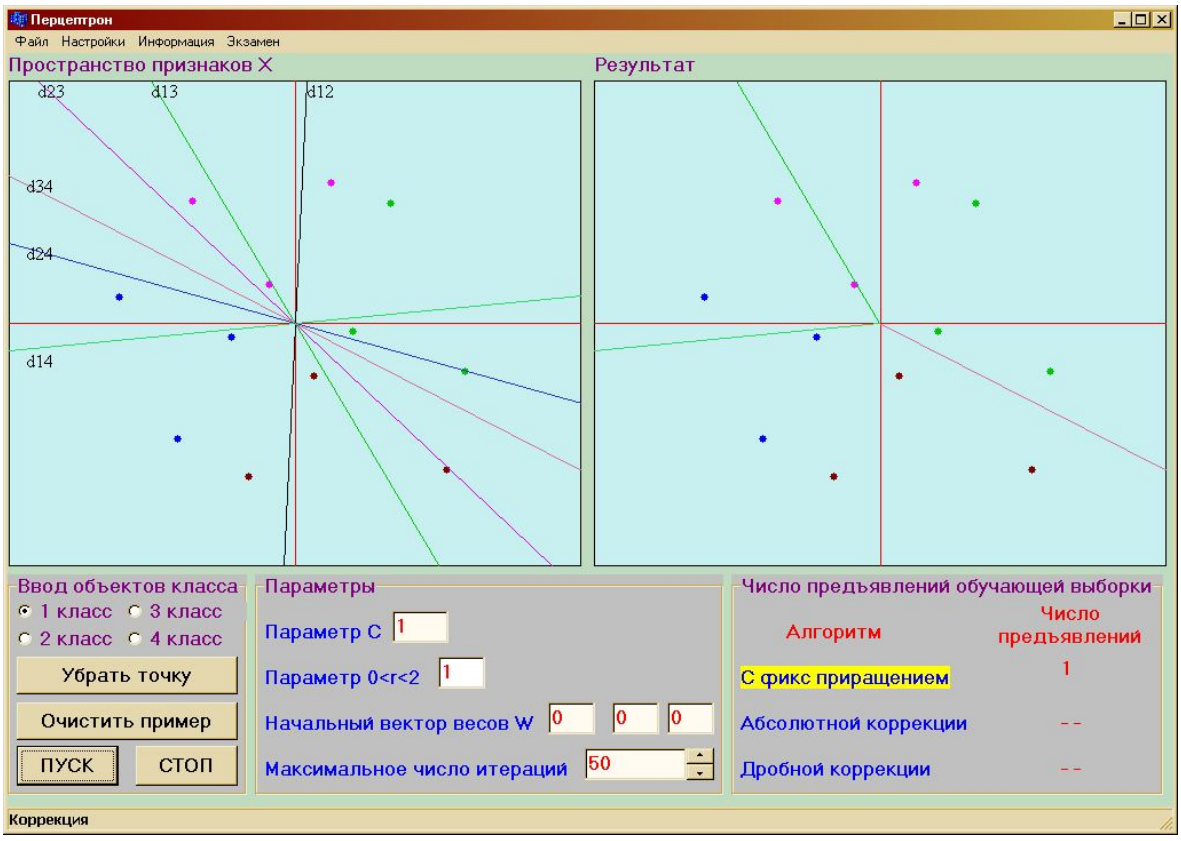

Рис. 21. Первая итерация для примера четырех классов

При изменении настроек или параметров для одного вида алгоритма число итераций, за которое происходит обучение перцептрона, для других видов не изменяется. Поэтому можно сравнивать скорость сходимости алгоритмов.

Экзамен

После получения решающей функции следует провести экзамен. Для этого выбираем пункт меню «Экзамен» и следуем инструкциям в правой части окна. После ввода экзаменационных объектов происходит их отнесение к различным классам в соответствии с решающей функцией.

На рис.22 можно увидеть результаты экзамена для решающей функции из примера 2.

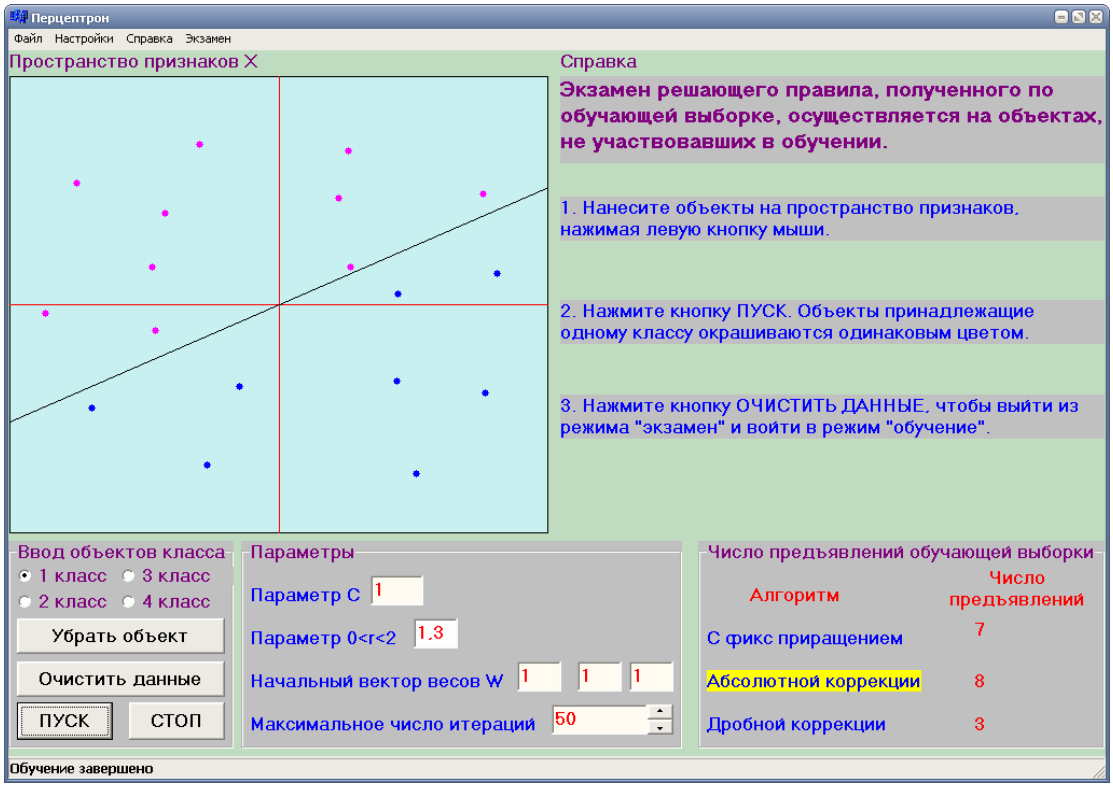

Рис. 22. Экзамен решающего правила для примера 2

ЛИТЕРАТУРА

1. Розенблатт Ф. Принципы нейродинамики. Перцептрон и теория механизмов. - М.: Мир, 1966.
2. Нильсон Н. Обучающиеся машины. – М.: Мир, 1967. - 180 с.
3. Дуда Р., Харт П. Распознавание образов и анализ сцен. – М.: Мир, 1976. – 512 с.
4. Ту Дж., Гонсалес Р. Принципы распознавания образов. – М. : Мир, 1978. - 416 с.

СОДЕРЖАНИЕ