Учреждение образования
«Белорусский государственный университет информатики и радиоэлектроники»

Факультет вечернего, заочного и дистанционного обучения
Кафедра ЭВМ

Контрольная работа № 2
по дисциплине «Цифровая обработка сигналов и изображений»

студента 5 курса 500502 учебной группы
Авсеева С.П.

Минск 2010

Содержание

1. Практическая часть

1.1. Задание

Напишите программу, реализующую конкурентную нейронную сеть. Обучите конкурентную сеть на количество образов, превышающих количество нейронов сети.

1.2. Реализация

Класс, реализующий поведение нейрона.

```
class Neuron
  attr_accessor :id
  attr_accessor :weights
  attr_accessor :network
  attr_accessor :wins

  def initialize(id, network, weights = nil)
    @id = id
    @network = network
    @weights = weights || Array.new(network.number_of_synapses){rand}
    @wins = {}
    log_state('initial␣state', true)
  end

  def increase_link_with(input)
    @weights = @weights.add(input.data.sub(@weights).mul(@network.speed))
    @wins[input.filename] ||= 0
    @wins[input.filename] += 1
    log_state("win␣on␣#{input.filename}")
  end

  def reaction_on(input)
    wins = @wins.inject(0){|s,v| s+=v.last}
    wins = 1 if wins.zero?
    input.data.sub(@weights).dist * wins
  end

  protected
  def log_state(message = nil, truncate = false)
    File.open("neuron-#{id}.txt", truncate ? "w" : "a") do |f|
      f.puts(message) if message
      f.puts(@weights.inspect)
    end
  end
end
```

Класс, реализующий нейронную сеть.

```
require 'neuron'

class Network
  attr_accessor :neurons
  attr_accessor :speed
  attr_accessor :learn_steps
  attr_accessor :number_of_synapses


  def initialize(number_of_synapses, neurons_count)
    @number_of_synapses = number_of_synapses
    @neurons = Array.new(neurons_count){|i| Neuron.new(i, self) }
    @speed = 1.0 / neurons_count
    @learn_steps = learn_steps
  end

  def learn(input)
    winner = neurons.min{|a,b| a.reaction_on(input) <=> b.reaction_on(input)}
    winner.increase_link_with(input)
    winner
  end

  def recognize(input)
    neurons.min{|a,b| a.reaction_on(input) <=> b.reaction_on(input)}
  end
end
```

Тестовая программа

```
#!/usr/bin/env ruby

require 'image'
require 'array_ext'
require 'network'

image_width = image_height = 10
number_of_synapses = image_width * image_height
network = Network.new(number_of_synapses, 4)

images = Image.load(Dir['image-*.png'], image_width, image_height)

i = 0
prev_max_dist = 0
loop do
  i += 1
  $stderr.print "\обучениер...␣#{i}"
  max_dist = images.map do |image|
               winner = network.learn(image)
               image.data.sub(winner.weights).dist
             end.max
  break if (prev_max_dist - max_dist).abs < 0.001
  prev_max_dist = max_dist
end
puts "\n" + network.neurons.map{|n| "нейрон␣#{n.id}.␣#{n.wins.inspect}"}.join("\n")

puts "зашумление..."
images = Image.load(Dir['image-*.png'], image_width, image_height)
images.each do |image|
  (1..10).each do |level|
    new_name = image.filename.gsub('image', "noise-#{level}")
    image.add_noise(level).write(new_name)
  end
end

puts "распознавание..."

(1..10).each do |level|
  puts "\зашумлениен␣#{level*10}%"
  images = Image.load(Dir["noise-#{level}-*.png"], image_width, image_height)
  images.sort.each do |image|
    winner = network.recognize(image)
    puts "#{image.filename}.␣победитель:␣#{winner.id}.␣#{winner.wins.inspect}"
  end
end
```

На вход программе в режиме обучения подавались образы представленные на рисунке 1.1, а в режиме распознавания образы с рисунков 1.2-1.11. Ниже представлен вывод после одного из запусков.

```
обучение... 17
нейрон 0. {”image-c.png”=>1, ”image-a.png”=>17}
нейрон 1. {”image-1.png”=>17}
нейрон 2. {”image-3.png”=>14, ”image-5.png”=>16}
нейрон 3. {”image-c.png”=>16, ”image-3.png”=>3, ”image-5.png”=>1}
зашумление...
распознавание...

зашумление 10%
noise-1-1.png. победитель: 1. {”image-1.png”=>17}
noise-1-3.png. победитель: 2. {”image-3.png”=>14, ”image-5.png”=>16}
noise-1-5.png. победитель: 2. {”image-3.png”=>14, ”image-5.png”=>16}
noise-1-a.png. победитель: 0. {”image-c.png”=>1, ”image-a.png”=>17}
noise-1-c.png. победитель: 3. {”image-c.png”=>16, ”image-3.png”=>3, ”image-5.png”=>1}

зашумление 20%
noise-2-1.png. победитель: 1. {”image-1.png”=>17}
noise-2-3.png. победитель: 3. {”image-c.png”=>16, ”image-3.png”=>3, ”image-5.png”=>1}
noise-2-5.png. победитель: 1. {”image-1.png”=>17}
noise-2-a.png. победитель: 0. {”image-c.png”=>1, ”image-a.png”=>17}
noise-2-c.png. победитель: 3. {”image-c.png”=>16, ”image-3.png”=>3, ”image-5.png”=>1}

зашумление 30%
noise-3-1.png. победитель: 1. {”image-1.png”=>17}
noise-3-3.png. победитель: 1. {”image-1.png”=>17}
noise-3-5.png. победитель: 1. {”image-1.png”=>17}
noise-3-a.png. победитель: 0. {”image-c.png”=>1, ”image-a.png”=>17}
noise-3-c.png. победитель: 3. {”image-c.png”=>16, ”image-3.png”=>3, ”image-5.png”=>1}

зашумление 40%
noise-4-1.png. победитель: 1. {”image-1.png”=>17}
noise-4-3.png. победитель: 1. {”image-1.png”=>17}
noise-4-5.png. победитель: 1. {”image-1.png”=>17}
noise-4-a.png. победитель: 0. {”image-c.png”=>1, ”image-a.png”=>17}
noise-4-c.png. победитель: 1. {”image-1.png”=>17}

зашумление 50%
noise-5-1.png. победитель: 1. {”image-1.png”=>17}
noise-5-3.png. победитель: 0. {”image-c.png”=>1, ”image-a.png”=>17}
noise-5-5.png. победитель: 1. {”image-1.png”=>17}
noise-5-a.png. победитель: 1. {”image-1.png”=>17}
noise-5-c.png. победитель: 1. {”image-1.png”=>17}

зашумление 60%
noise-6-1.png. победитель: 0. {”image-c.png”=>1, ”image-a.png”=>17}
noise-6-3.png. победитель: 1. {”image-1.png”=>17}
noise-6-5.png. победитель: 1. {”image-1.png”=>17}
noise-6-a.png. победитель: 1. {”image-1.png”=>17}
noise-6-c.png. победитель: 1. {”image-1.png”=>17}

зашумление 70%
noise-7-1.png. победитель: 0. {”image-c.png”=>1, ”image-a.png”=>17}
noise-7-3.png. победитель: 1. {”image-1.png”=>17}
noise-7-5.png. победитель: 0. {”image-c.png”=>1, ”image-a.png”=>17}
noise-7-a.png. победитель: 1. {”image-1.png”=>17}
noise-7-c.png. победитель: 1. {”image-1.png”=>17}
```

```
зашумление 80%
noise-8-1.png. победитель: 0. {”image-c.png”=>1, ”image-a.png”=>17}
noise-8-3.png. победитель: 0. {”image-c.png”=>1, ”image-a.png”=>17}
noise-8-5.png. победитель: 0. {”image-c.png”=>1, ”image-a.png”=>17}
noise-8-a.png. победитель: 1. {”image-1.png”=>17}
noise-8-c.png. победитель: 1. {”image-1.png”=>17}

зашумление 90%
noise-9-1.png. победитель: 0. {”image-c.png”=>1, ”image-a.png”=>17}
noise-9-3.png. победитель: 1. {”image-1.png”=>17}
noise-9-5.png. победитель: 0. {”image-c.png”=>1, ”image-a.png”=>17}
noise-9-a.png. победитель: 1. {”image-1.png”=>17}
noise-9-c.png. победитель: 1. {”image-1.png”=>17}

зашумление 100%
noise-10-1.png. победитель: 0. {”image-c.png”=>1, ”image-a.png”=>17}
noise-10-3.png. победитель: 1. {”image-1.png”=>17}
noise-10-5.png. победитель: 0. {”image-c.png”=>1, ”image-a.png”=>17}
noise-10-a.png. победитель: 1. {”image-1.png”=>17}
noise-10-c.png. победитель: 1. {”image-1.png”=>17}
```

Рис. 1.1: Образы для обучения

Рис. 1.2: Образы с зашумлением 10%

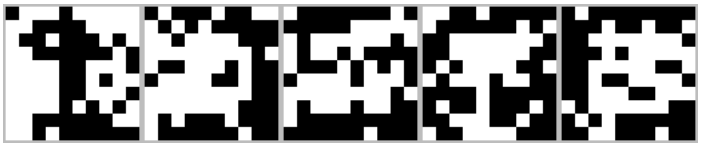

Рис. 1.3: Образы с зашумлением 20%

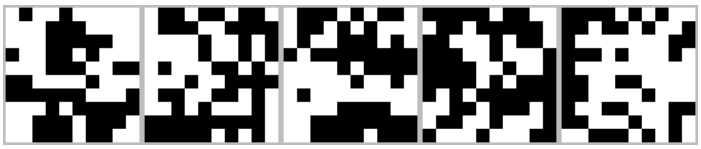

Рис. 1.4: Образы с зашумлением 30%

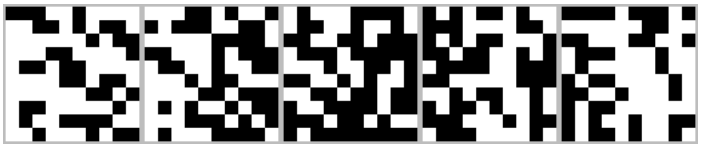

Рис. 1.5: Образы с зашумлением 40%

Рис. 1.6: Образы с зашумлением 50%

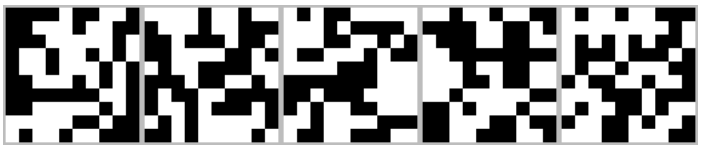

Рис. 1.7: Образы с зашумлением 60%

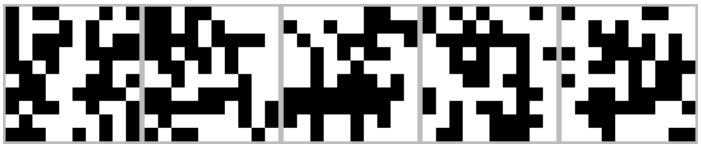

Рис. 1.8: Образы с зашумлением 70%

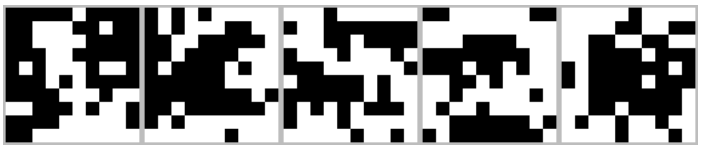

Рис. 1.9: Образы с зашумлением 80%

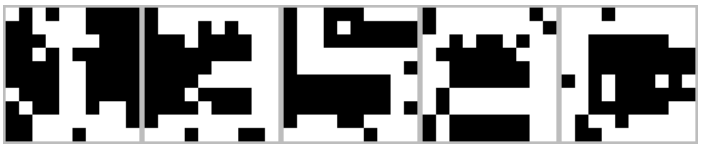

Рис. 1.10: Образы с зашумлением 90%

Рис. 1.11: Образы с зашумлением 100%

2. Теоретическая часть

2.1. Метод отображения по алгоритму наименьших квадратов (11)

Один из подходов для группировки классов образов заключается в том что классификатор должен в первую очередь отображать образы в пространство решений, в котором образы, принадлежащие C_i обязательно группируются вокруг заранее выбранной точки $V_i,\ i = 1, 2, ..., K$. Преобразование A, которое позволяет осуществлять это отображение из пространства признаков в пространство решений в общем случае выбирается таким, чтобы общая среднеквадратичная ошибка была минимальной. Для классификации некоторого образа этот образ сначала отображается в пространство решений, а затем классифицируется как принадлежащий C_{io}, если он отображён ближе к точке V_{io}. Классификаторы с минимальным среднеквадратичным расстоянием основываются на отображении по методу наименьших квадратов.

Рассмотрим множество M-мерных образов $Z_{ij},\ j = 1, 2, ..., N_i$, которые должны отображаться в определенную точку в K-мерном пространства, обозначаемую $V_i' = [v_1 v_2 ... v_k]$. Найдем преобразование A, которое отображает $\{Z_{ij}\}$ в V_i такм образом, чтобы общая среднеквадратичная ошибка, вызываемая отображением, была минимальной. Обозначим результат отображения образа Z_{ij} через L_{ij}. Тогда соответствующий вектор ошибки равен

$$\varepsilon_j = L_{ij} - V_i = AZ_{ij} - V_i \tag{2.1}$$

Из выражения (2.1) следует, что общая среднеквадратичная ошибка при отображении $\{Z_{ij}\}$ в V_i определяется как

$$\varepsilon = \frac{1}{N_i} \sum_{j=1}^{N_i} \|\varepsilon_j\|^2 \tag{2.2}$$

Подстановка (2.1) в (2.2) приводит к

$$\begin{aligned} \varepsilon =& \frac{1}{N_i} \sum_{j=1}^{N_i} \|\varepsilon_j\|^2 = \frac{1}{N_i} \sum_{j=1}^{N_i} (AZ_{ij} - V_i)' (AZ_{ij} - V_i) = \\ &= \frac{1}{N_i} \sum_{j=1}^{N_i} (AZ_{ij})' AZ_{ij} - 2 (AZ_{ij})' V_i + V_i' V_i = \\ &= \frac{1}{N_i} \sum_{j=1}^{N_i} \left\{ Z_{ij}' A' AZ_{ij} - 2Z_{ij}' A' V_i + \|V_i\|^2 \right\} \end{aligned} \tag{2.3}$$

Так как A должно быть выбрано так, чтобы ε было минимальным, то оно получается в результате решения уравнения $\nabla_A \varepsilon = 0$, что приводит к

$$\frac{1}{N_i}\sum_{j=1}^{N_i}\nabla_A\left\{Z'_{ij}A'AZ_{ij}\right\} - \frac{2}{N_i}\sum_{j=1}^{N_i}\nabla_A\left\{Z'_{ij}A'V_i\right\} + \nabla_A\left\{\|V_i\|^2\right\} = 0 \tag{2.4}$$

Так как

$$\begin{aligned}
&\nabla_A\left\{Z'_{ij}A'AZ_{ij}\right\} = 2A\left(Z_{ij}Z'_{ij}\right)\\
&\nabla_A\left\{Z'_{ij}A'V_i\right\} = V_iZ'_{ij}\\
&\nabla_A\left\{\|V_i\|^2\right\} = 0
\end{aligned} \tag{2.5}$$

то применение приведённых выше тождеств к выражению (2.4) приводит к

$$A\left[\frac{2}{N_i}\sum_{j=1}^{N_i}\left(Z_{ij}Z'_{ij}\right)\right] = \frac{2}{N_i}\sum_{j=1}^{N_i}V_iZ'_{ij} \tag{2.6}$$

Что позволяет определить A как

$$A = \left[\sum_{j=1}^{N_i}V_iZ'_{ij}\right]\left[\sum_{j=1}^{N_i}Z_{ij}Z'_{ij}\right]^{-1} \tag{2.7}$$

Вышеприведенные рассуждения и вывод взяты из книги «Ортогональные преобразования при обработке цифровых сигналов» [1].

Рассмотрим пример. Пусть множество Z_{ij} имеет вид

$$Z_{i1} = \begin{bmatrix}4\\4\end{bmatrix};\ Z_{i2} = \begin{bmatrix}3\\4\end{bmatrix};\ Z_{i3} = \begin{bmatrix}3\\5\end{bmatrix};\ Z_{i4} = \begin{bmatrix}3\\6\end{bmatrix};\ Z_{i5} = \begin{bmatrix}4\\6\end{bmatrix},$$

что соответствует $N_i = V_i5$. Пусть $V_i = \begin{bmatrix}1\\1\end{bmatrix}$. Тогда

$$\sum_{j=1}^{5}V_iZ'_{ij} = \begin{bmatrix}17 & 25\\17 & 25\end{bmatrix};\ \sum_{j=1}^{5}Z_{ij}Z'_{ij} = \begin{bmatrix}59 & 85\\85 & 129\end{bmatrix}. \tag{2.8}$$

Подстановка выражений (2.8) в уравнение (2.7) даёт

$$A = \begin{bmatrix}0.176 & 0.078\\0.176 & 0.078\end{bmatrix} \tag{2.9}$$

Вычислим $L_{ij} = AZ_{ij}$

$$L_{i1} = \begin{bmatrix} 1.016 \\ 1.016 \end{bmatrix};\ L_{i2} = \begin{bmatrix} 0.84 \\ 0.84 \end{bmatrix};\ L_{i3} = \begin{bmatrix} 0.918 \\ 0.918 \end{bmatrix};\ L_{i4} = \begin{bmatrix} 0.996 \\ 0.996 \end{bmatrix};\ L_{i5} = \begin{bmatrix} 1.172 \\ 1.172 \end{bmatrix}.$$

Множество образов $\{Z_{ij}\}$ и $\{L_{ij}\}$ показано на рис. 2.1.

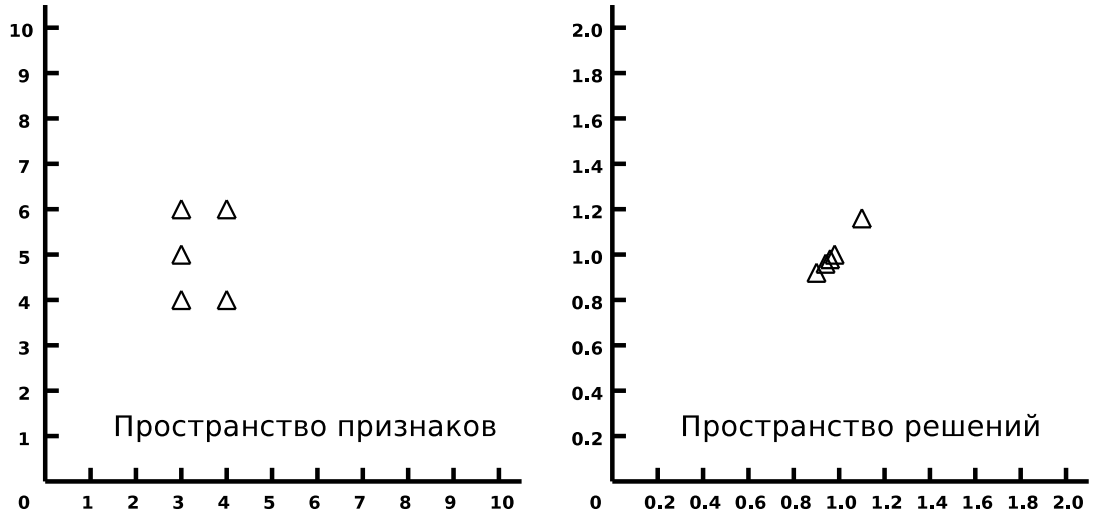

Рис. 2.1: Отображение по методу наименьших квадратов

2.2. Задача двухклассового распознавания (24)

Предположим, что мы желаем обучить классификатор автоматически классифицировать образ Z , принадлежащий либо классу C_1, либо классу C_2. Предположим также что обучающее множество (т.е. множество истинная классификация которого известна) состоит из следующего множества двумерных образов Z_{ij} , где Z_{ij} обозначает j-ый образ, принадлежащий $C_i,\ i = 1, 2$ [1]:

$$\begin{aligned} C_1 : Z_{11} &= \begin{bmatrix} 5 \\ 5 \end{bmatrix};\ Z_{12} = \begin{bmatrix} 6 \\ 5 \end{bmatrix};\ Z_{13} = \begin{bmatrix} 6 \\ 6 \end{bmatrix};\ Z_{14} = \begin{bmatrix} 6 \\ 7 \end{bmatrix};\ Z_{15} = \begin{bmatrix} 7 \\ 5 \end{bmatrix}; \\ C_2 : Z_{21} &= \begin{bmatrix} 0 \\ 3 \end{bmatrix};\ Z_{22} = \begin{bmatrix} -1 \\ 3 \end{bmatrix};\ Z_{23} = \begin{bmatrix} -2 \\ 3 \end{bmatrix};\ Z_{24} = \begin{bmatrix} -3 \\ 3 \end{bmatrix};\ Z_{25} = \begin{bmatrix} -4 \\ 3 \end{bmatrix}; \end{aligned} \tag{2.10}$$

Образы Z_{ij}, принадлежащие классам C_1 и C_2, располагаются в двуменом пространстве признаков, как показано на рис. 2.2.

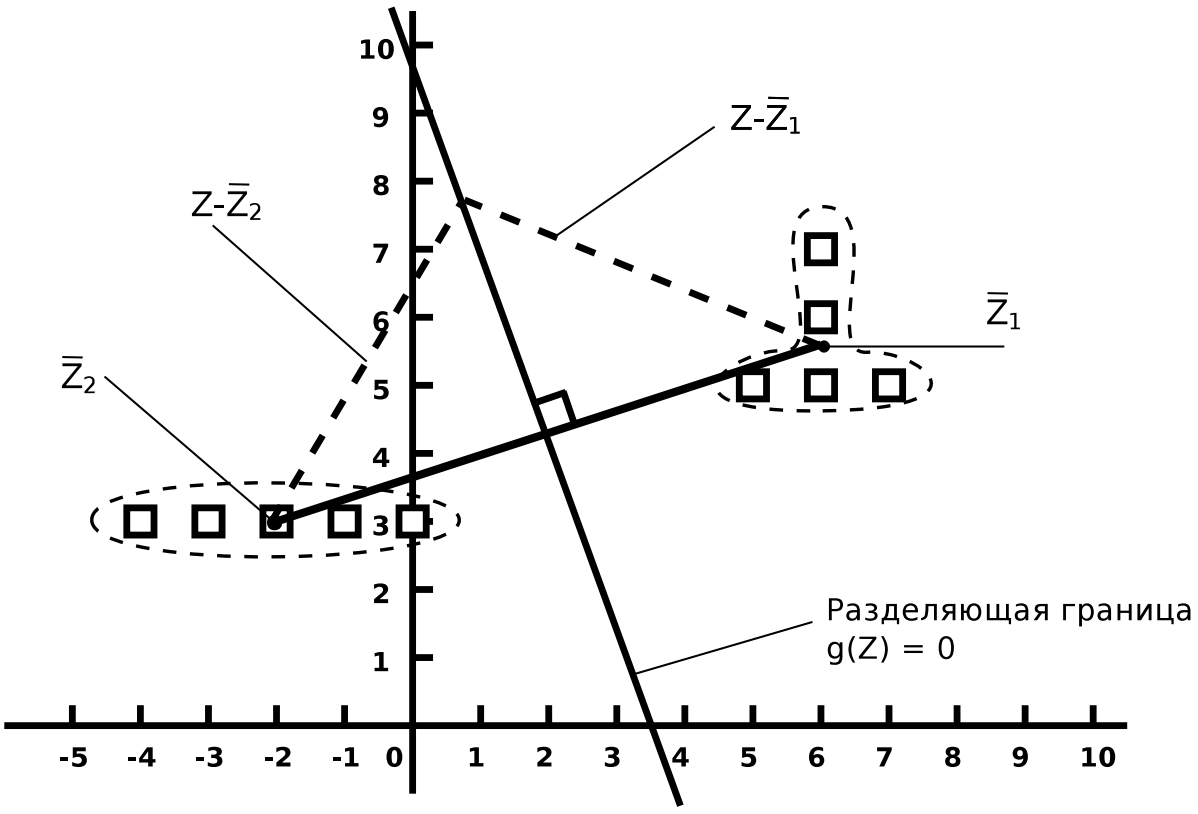

Рис. 2.2: Двумерное пространство признаков

Пусть $\overline{Z_1}$ и $\overline{Z_2}$ — средние векторы образов, связанные с C_1 и C_2 соответ-

ственно. Тогда

$$\overline{Z_i} = \frac{1}{5}\sum_{j=1}^{5} Z_{ij},\ i = 1, 2, \tag{2.11}$$

что дает $\overline{Z_1} = \begin{bmatrix} 6 \\ 5,6 \end{bmatrix}$ и $\overline{Z_2} = \begin{bmatrix} -2 \\ 3 \end{bmatrix}$

Из рис. 2.2 видно что наиболее целесообразной решающей границуй (т.е. линией на плоскости), разделяющей классы C_1 и C_2, является срединный перпендикуляр к прямой, соединяющей $\overline{Z_1}$ и $\overline{Z_2}$. Следующим шагом является описание предлагаемой решающей границы с помощью уравнения.

Рассмотрим любую точку Z, принадлежащую решающей границе, как показано на рис. 2.2. Так как решающая граница является срединным перпендикуляром к прямой, соединяющей $\overline{Z_1}$ и $\overline{Z_2}$, то $\|Z - \overline{Z_1}\|^2 = \|Z - \overline{Z_2}\|^2$; это в более простом виде можно записать как

$$\left(\overline{Z_1} - \overline{Z_2}\right)' Z = \frac{1}{2}\left\{\|\overline{Z_1}\|^2 - \|\overline{Z_2}\|^2\right\}. \tag{2.12}$$

Подставляя $\overline{Z'} = [z_1 z_2]$ и выражение (2.11) в выражение (2.12), получаем уравнение для решающей границы в виде

$$8z_1 + 2,6z_2 = 27,18. \tag{2.13}$$

Величина $0,5\left(\|\overline{Z_1}\|^2 - \|\overline{Z_2}\|^2\right) = 27,18$ называется порогом классификатора. Уравнение (2.13) даёт всю информацию, необходимую для создания классификатора. Основной характеристикой, представляющей классификатор, является дискриминантная функция $g(Z)$, которая определяется как

$$g(Z) = 8z_1 + 2,6z_2 - 27,18. \tag{2.14}$$

Благодаря дискриминантной функции можно вывести решающее правило: если $g(Z) > 0$, то $Z \in C_1$ и если $g(Z) < 0$, то $Z \in C_2$.

Обучающее множество образов используется для создания классификатора, который получается после вычисления дискриминантной функции $g(Z)$. Получив дискриминантную функцию $g(Z)$, говорят, что классфикатор обучен, т.е. способен классфицировать образы с помощью соответствующего решающего правила. Элемент пороговой логики (рис. 2.3) является классификатором, работающим по критерию минимума расстояния, так как решающее правило может быть сформулировано также сле-

дующим образом: если Z ближе к $\overline{Z_1}$, то $Z \in C_1$ и если Z ближе к $\overline{Z_2}$, то $Z \in C_2$.

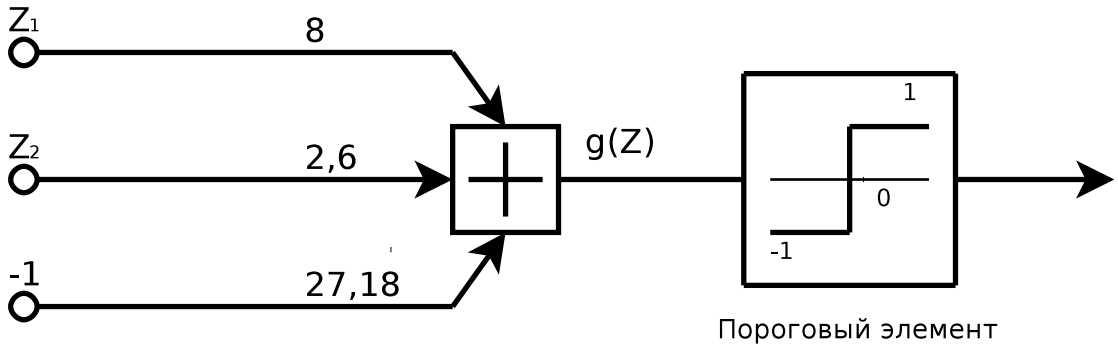

Рис. 2.3: Элемент линейной пороговой логики

Список литературы

[1] Ахмед Н., Рао К.Р. *Ортогональные преобразования при обработке цифровых сигналов*: Пер. с англ./Под ред. И. Б. Фоменко. — М.: Связь, 1980. — 248 с., ил.