

1. Теорема линейности и комплексной сопряженности.

Из описанного семейства преобразований к цифровой обработке сигналов и изображений имеет отношение дискретное преобразование Фурье, которое оперирует дискретной по времени выборкой периодического сигнала во временной области. Для того, чтобы быть представленным в виде суммы синусоид, сигнал должен быть периодическим. Но в качестве набора входных данных для ДПФ доступно только конечное число отсчетов (N).

Таким образом, если $\{X(m)\}$ означает последовательность $X(m)$ конечных действительных или комплексных чисел, где $m = \overline{0, N-1}$, то дискретное преобразование Фурье этой последовательности определяется как

$$C_x(k) = \frac{1}{N} \sum_{m=0}^{N-1} X(m)W^{km}, \text{ где } k = \overline{0, N-1}, W = e^{-i2\pi/N}, i = \sqrt{-1}$$

$$X(m) = \sum_{k=0}^{N-1} C_x(k)W^{-km} \quad \text{- пара преобразований Фурье}$$

Функции W^{km} являются N-периодическими, т.е. $W^{km} = W^{(k+N)m} = W^{k(m+N)}$.

Следовательно, последовательности $\{C_x(k)\}, \{X(m)\}$ также являются N-

периодическими, т.е.

$X(\pm m) = X(SN \pm m)$; Рассмотрим **основные свойства дискретного преобразования**

$C_x(\pm k) = C_x(SN \pm k)$. **Фурье:**

а) **теорема линейности:** дискретное преобразование Фурье является линейным, т.е. если $X(m) \leftrightarrow C_x(k), Y(m) \leftrightarrow C_y(k)$ и $Z(m) = aX(m) + bY(m)$, то $C_z(k) = aC_x(k) + bC_y(k)$;

б) **теорема комплексной сопряженности:** если $\{X(m)\} = \{X(0), X(1), \dots, X(N-1)\}$

– такая последовательность действительных чисел, что $N/2$ – целое число и

$X(m) \leftrightarrow C_x(k)$, то

$$C_x\left(\frac{N}{2} + l\right) = \overline{C_x\left(\frac{N}{2} - l\right)}, \forall l = \overline{0, N/2}. \tag{2.24}$$

Из пары преобразований Фурье следует:

что $C_x(k) = \frac{1}{N} \sum_{m=0}^{N-1} X(m)W^{km}$, где $W = e^{-i2\pi/N}$.

Тогда, подставляя в место $k - (N/2 + l)$, будем иметь

$$C_x\left(\frac{N}{2} + l\right) = \frac{1}{N} \sum_{m=0}^{N-1} X(m)W^{(N/2+l)m} = \frac{1}{N} \sum_{m=0}^{N-1} X(m)W^{-(N/2-l)m} W^{Nm} = \overline{C_x\left(\frac{N}{2} - l\right)},$$

т.к. $W^{Nm} = e^{-i2\pi m} \equiv 1$.

2. Ортогональность сигналов. Теорема Парсеваля.

Множество непрерывных функций действительного переменного $\{U_n(t)\} = \{U_0(t), U_1(t), \dots\}$ называется **ортогональным** на интервале $[t_0, t_0+T]$, если

$$\int_{t_0}^{t_0+T} U_m(t)U_n(t)dt = \begin{cases} c, & \forall m = n, \\ 0, & \forall m \neq n \end{cases}$$

При $c = 1$ множество $\{U_n(t)\}$ называется ортонормированным. Для вычисления сигнала через коэффициенты разложения используется:

$$x(t) = \sum_{n=0}^{\infty} a_n U_n(t)$$

Коэффициент разложения из указанного соотношения можно определить, если умножить обе части выражения слева на $U_m(t)$ и проинтегрировать в интервале $[t_0, t_0+T]$

$$\int_{t_0}^{t_0+T} x(t)U_m(t)dt = \sum_{n=0}^{\infty} a_n \int_{t_0}^{t_0+T} U_n(t)U_m(t)dt$$

$$a_n = \frac{1}{C} \int_{t_0}^{t_0+T} x(t)U_n(t)dt$$

В силу условий ортогональности будем иметь:

$$\vec{x} = U\vec{a}$$

В матричном виде форме:

где x - вектор - столбец, U - матрица ортогональных функций.

Умножим обе части на матрицу U^{-1} :

$$U^{-1}\vec{x} = U^{-1}U\vec{a} \quad a = \frac{1}{C}U^{-1}\vec{x}$$

так как $U^{-1}U = UU^{-1} = CI_d$, где I_d - единичная матрица, тогда

Теорема Парсеваля:

Для доказательства теоремы Парсеваля возведем обе части соотношения в квадрат:

$$x(t) = \sum_{n=0}^{\infty} a_n U_n(t) \quad \rightarrow \quad x(t)^2 = \sum_{n=0}^{\infty} U_n(t)^2 a_n^2 + \sum_{p=0}^{\infty} \sum_{q=0}^{\infty} a_p a_q U_p(t)U_q(t)$$

Проинтегрируем обе части:

$$\frac{1}{T} \int_T x(t)^2 dt = \sum_{n=0}^{\infty} a_n^2 \frac{1}{T} \int_T U_n(t)^2 dt + \sum_{p=0}^{\infty} \sum_{q=0}^{\infty} a_p a_q \int_T U_p(t)U_q(t) dt$$

По условию ортогональности:
$$\frac{1}{T} \int_T x(t)^2 dt = \frac{C}{T} \sum_{n=0}^{\infty} a_n^2$$

Если $x(t)$ - напряжение или ток, приложенный к концам резистивной нагрузки равной 1 Ом, то левая часть выражения представляет мощность

рассеиваемой энергии, а $\frac{C}{T} \sum_{n=0}^{\infty} a_n^2$ является распределением мощности в $x(t)$.

3. Теорема сдвига.

в) теорема сдвига: если $Z(m) \leftrightarrow C_z(k)$ и $Z(m) = X(m+h)$, $h = \overline{0, N-1}$, то

$$C_z(k) = W^{-kh} C_x(k). \quad (2.25)$$

Доказательство:

$$Z(m) \leftrightarrow C_z(k), \text{ т.е. } C_z(k) = \frac{1}{N} \sum_{m=0}^{N-1} Z(m) W^{km}, \quad k = \overline{0, N-1}.$$

С учетом подстановки $Z(m) = X(m+h)$, будем иметь $C_z(k) = \frac{1}{N} \sum_{m=0}^{N-1} X(m+h) W^{km}$.

Осуществляя замену переменных $m+h=r$, указанное соотношение будет

$$\text{иметь вид } C_z(k) = W^{-kh} \left\{ \frac{1}{N} \sum_{r=h}^{N+h-1} X(r) W^{kr} \right\}.$$

Так как

$$\sum_{m=p}^q X(m) W^{km} = \sum_{m=0}^{N-1} X(m) W^{km},$$

$\sum_{m=p}^q C_x(k) W^{-km} = \sum_{m=0}^{N-1} C_x(k) W^{-km}$, когда p и q удовлетворяют условию

$$|p-q| = N-1, \text{ то } C_z(k) = W^{-kh} C_x(k).$$

Аналогично при $Z(m) = X(m-h)$, $C_z(k) = W^{kh} C_x(k)$.

4. Теорема свертки

Свертка – это математический способ комбинирования двух сигналов для формирования третьего сигнала. Это один из самых важных методов ЦОС. Пользуясь стратегией импульсного разложения, системы описываются сигналом, называемым импульсной характеристикой. Свертка важна, так как она связывает три сигнала: входной сигнал, выходной сигнал и импульсную характеристику. Корреляция, так же, как свертка, использует два сигнала для получения третьего. Этот третий сигнал называется корреляционным сигналом двух входных сигналов.

Если $\{X(m)\}$ и $\{Y(m)\}$ – последовательности действительных чисел, для которых $X(m) \leftrightarrow C_x(k)$, $Y(m) \leftrightarrow C_y(k)$, и свертка этих последовательностей определяется как

$$Z(m) = \frac{1}{N} \sum_{h=0}^{N-1} X(h)Y(m-h), \quad m = \overline{0, N-1}, \quad (3.1)$$

То: $C_z(k) = C_x(k)C_y(k)$.

Доказательство: Вычисляя $Z(m)$, получим

$$C_z(k) = \frac{1}{N} \sum_{m=0}^{N-1} Z(m)W^{km}. \quad (3.2)$$

Подставляя в (3.2) соотношение свертки (3.1), получим

$$C_z(k) = \frac{1}{N^2} \sum_{m=0}^{N-1} \sum_{h=0}^{N-1} x(h)Y(m-h)W^{km} = \frac{1}{N} \sum_{h=0}^{N-1} x(h) \frac{1}{N} \sum_{m=0}^{N-1} Y(m-h)W^{km}$$

Согласно теореме сдвига, имеем

$$\frac{1}{N} \sum_{m=0}^{N-1} Y(m-h)W^{km} = W^{kh} C_y(k)$$

Таким образом,

$$C_z(k) = C_y(k) \frac{1}{N} \sum_{h=0}^{N-1} x(h)W^{kh} = C_x(k)C_y(k).$$

Эта теорема утверждает, что свертка временных последовательностей эквивалентна умножению их коэффициентов, полученных после дискретного преобразования Фурье.

5. Схема вычисления корреляции и свертки и на основе БПФ

Если $\{X(m)\}$ и $\{Y(m)\}$ – две N -периодические последовательности действительных чисел, то операции корреляции и свертки определяются соответственно как

$$\hat{Z}(m) = \frac{1}{N} \sum_{h=0}^{N-1} X(h)Y(m+h),$$

$$Z(m) = \frac{1}{N} \sum_{h=0}^{N-1} X(h)Y(m-h).$$

Если $N=4$, то

$$4\hat{Z}(0) = X(0)Y(0) + X(1)Y(1) + X(2)Y(2) + X(3)Y(3);$$

$$4\hat{Z}(1) = X(0)Y(1) + X(1)Y(2) + X(2)Y(3) + X(3)Y(0);$$

$$4\hat{Z}(2) = X(0)Y(2) + X(1)Y(3) + X(2)Y(0) + X(3)Y(1);$$

$$4\hat{Z}(3) = X(0)Y(3) + X(1)Y(0) + X(2)Y(1) + X(3)Y(2);$$

$$\begin{bmatrix} \hat{Z}(0) \\ \hat{Z}(1) \\ \hat{Z}(2) \\ \hat{Z}(3) \end{bmatrix} = \frac{1}{4} \begin{bmatrix} X(0) & X(1) & X(2) & X(3) \\ X(3) & X(0) & X(1) & X(2) \\ X(2) & X(3) & X(0) & X(1) \\ X(1) & X(2) & X(3) & X(0) \end{bmatrix} \begin{bmatrix} Y(0) \\ Y(1) \\ Y(2) \\ Y(3) \end{bmatrix}$$

При $N=4$ для свертки будем иметь

$$4Z(0) = X(0)Y(0) + X(1)Y(-1) + X(2)Y(-2) + X(3)Y(-3);$$

$$4Z(1) = X(0)Y(1) + X(1)Y(0) + X(2)Y(-1) + X(3)Y(-2);$$

$$4Z(2) = X(0)Y(2) + X(1)Y(1) + X(2)Y(0) + X(3)Y(-1);$$

$$4Z(3) = X(0)Y(3) + X(1)Y(2) + X(2)Y(1) + X(3)Y(0).$$

Так как $[Y(k)]_{\text{mod } N} = Y(k-N)$ при $k > 0$,

$[Y(k)]_{\text{mod } N} = Y(k+N)$ при $k < 0$, то

$$\begin{bmatrix} Z(0) \\ Z(1) \\ Z(2) \\ Z(3) \end{bmatrix} = \frac{1}{4} \begin{bmatrix} X(0) & X(1) & X(2) & X(3) \\ X(1) & X(2) & X(3) & X(0) \\ X(2) & X(3) & X(0) & X(1) \\ X(3) & X(0) & X(1) & X(2) \end{bmatrix} \begin{bmatrix} Y(0) \\ Y(3) \\ Y(2) \\ Y(1) \end{bmatrix}.$$

В общем виде корреляцию двух последовательностей можно записать как

$$\begin{bmatrix} \hat{Z}(0) \\ \hat{Z}(1) \\ \hat{Z}(2) \\ \vdots \\ \hat{Z}(N-2) \\ \hat{Z}(N-1) \end{bmatrix} = \frac{1}{N} \begin{bmatrix} X(0) & X(1) & X(2) & \dots & X(N-1) \\ X(N-1) & X(0) & X(1) & \dots & X(N-2) \\ X(N-2) & X(N-1) & X(0) & \dots & X(N-3) \\ \vdots & \vdots & \vdots & \dots & \vdots \\ X(2) & X(3) & X(4) & \dots & X(1) \\ X(1) & X(2) & X(3) & \dots & X(0) \end{bmatrix} \begin{bmatrix} Y(0) \\ Y(1) \\ Y(2) \\ \vdots \\ Y(N-2) \\ Y(N-1) \end{bmatrix}$$

В свою очередь, соотношение свертки можно записать в общем виде как

$$\begin{bmatrix} Z(0) \\ Z(1) \\ Z(2) \\ \vdots \\ Z(N-2) \\ Z(N-1) \end{bmatrix} = \frac{1}{N} \begin{bmatrix} X(0) & X(1) & X(2) & \dots & X(N-1) \\ X(1) & X(2) & X(3) & \dots & X(0) \\ X(2) & X(3) & X(4) & \dots & X(1) \\ \vdots & \vdots & \vdots & \dots & \vdots \\ X(N-2) & X(N-1) & X(0) & \dots & X(N-3) \\ X(N-1) & X(0) & X(1) & \dots & X(N-2) \end{bmatrix} \begin{bmatrix} Y(0) \\ Y(N-1) \\ Y(N-2) \\ \vdots \\ Y(2) \\ Y(1) \end{bmatrix}$$

Если последовательности $\{X(m)\}$ и $\{Y(m)\}$ аналогичны друг другу, то

$$Z(m) = \frac{1}{N} \sum_{h=0}^{N-1} X(h)X(m+h), \text{ где } m = \overline{0, N-1}.$$

Это соотношение определяет автокорреляцию последовательности $\{X(m)\}$.

С использованием БПФ схема вычисления корреляции будет иметь вид рис.3.1.

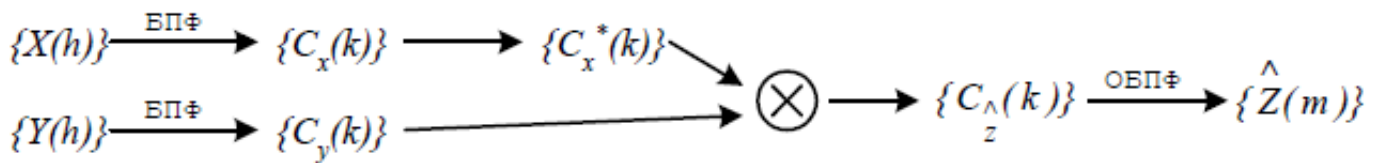


Рис. 3.1. Схема вычисления корреляции

В свою очередь, схему вычисления свертки можно представить как показано на ис.3.2.

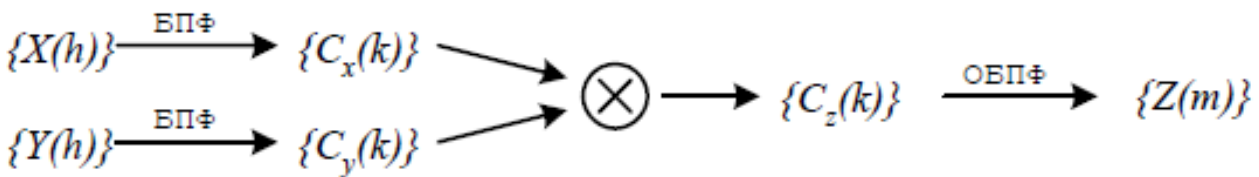


Рис. 3.2. Схема вычисления свертки

6. Виды разделяющих функций

А. Линейные разделяющие функции.

В этом случае в качестве $D(x)$ используется линейная комбинация признаков x_1, x_2, \dots, x_n

$$D_i(x) = \sum_{k=1}^n w_{ki} x_k + w_{i,n+1}$$

Решающая граница между областями w_i и w_j в пространстве признаков Ω имеет вид:

$$D_i(x) - D_j(x) = \sum_{k=1}^n w_k x_k + w_{n+1} = 0 \quad (22.1)$$

где $w_k = w_{ki} - w_{kj}$,
 $w_{n+1} = w_{i,n+1} - w_{j,n+1}$

Уравнение (22.1) представляет собой уравнение гиперплоскости в пространстве признаков Ω . Общая схема вычислителя линейной разделяющей функции представлена на рис. 22.1.

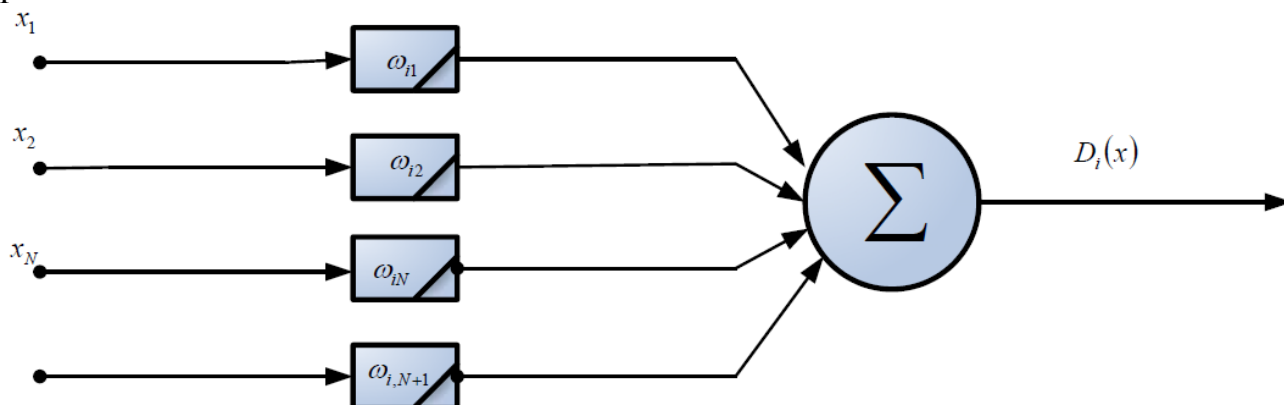


Рис. 22.1

Если $m = 2$, то согласно (22.1) $i, j = 1, 2$ и в качестве классификатора, использующего линейную разделяющую функцию, можно применить пороговый логический элемент.

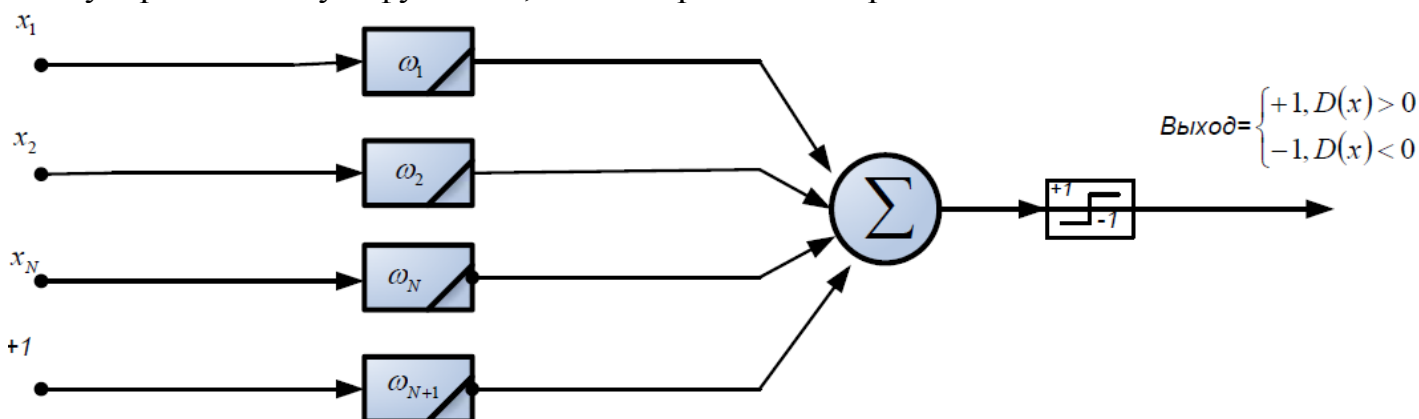


Рис. 22.2

Из этого рисунка получим, что если выходной сигнал $= +1$, т.е. при $D(x) > 0$, то $X \sim \omega_1$; если выходной сигнал $= -1$, т.е. $D(x) < 0$, то $X \sim \omega_2$. Когда число классов образов больше двух, т.е. $m > 2$ можно применить параллельное соединение нескольких пороговых ло-

гических элементов. При этом комбинации выходных сигналов M пороговых логических элементов достаточны для различения m классов при $2^M \geq m$.

С. Кусочно-линейная разделяющая функция.

Изложенная выше идея может быть распространена на классификацию по минимальным расстояниям до m множеств опорных векторов. Пусть R_1, R_2, \dots, R_m обозначают m множеств опорных векторов, относящихся соответственно к классам $\omega_1, \omega_2, \dots, \omega_m$ и пусть опорные векторы в R_j обозначены через $R_j^{(k)}$, т.е.

$$R_j^{(k)} \in R_j, \quad k = 1, \dots, U_j$$

где U_j - число опорных векторов множества R_j . Определим расстояние между вектором входных признаков и R_j следующим образом.

$$d(X, R_j) = \min_{k=1, \dots, U_j} |X - R_j^{(k)}|,$$

То есть расстояние между X и R_j равно наименьшему из расстояний между X и каждым вектором в R_j . Такой классификатор будет относить входной сигнал к классу образов, которому соответствует ближайшее множество векторов. Если расстояние между X и $R_j^{(k)}$ определить согласно (22.1) разделяющая функция в данном случае имеет вид

$$D_i(x) = \max_{k=1, \dots, U_j} \left\{ X^T R_i^{(k)} + X (R_i^{(k)})^T - (R_i^{(k)})^T R_i^{(k)} \right\}, \quad i = \overline{1, m}$$

Пусть $D_i^{(k)}(x) = X^T R_i^{(k)} + X (R_i^{(k)})^T - (R_i^{(k)})^T R_i^{(k)}$,

Тогда

$$D_i(x) = \max_{k=1, \dots, U_j} \left\{ D_i^{(k)}(x) \right\}$$

Следует отметить, что $D_i^{(k)}(x)$ является линейной комбинацией признаков. Поэтому указанный классификатор часто называют кусочно-линейным классификатором.

Д. Полиномиальная разделяющая функция.

Полиномиальная функция r -ой степени может быть представлена в виде

$$D_i(x) = w_{i1} f_1(x) + w_{i2} f_2(x) + \dots + w_{iL} f_L(x) + w_{i,L+1},$$

Где $f_i(x)$ является формой

$$x_{k_1}^{n_1} x_{k_2}^{n_2} \dots x_{k_r}^{n_r} \quad \text{при} \quad \begin{cases} k_1, k_2, \dots, k_r = 1, \dots, N \\ n_1, n_2, \dots, n_r = 0, 1 \end{cases}$$

Решающая граница между двумя классами также имеет форму полинома r -ой степени. В частности, если $r = 2$, решающая функция называется квадратичной. В этом случае:

$$f_j(x) = x_{k_1}^{n_1} x_{k_2}^{n_2} \text{ при } \begin{matrix} k_1, k_2 = 1, \dots, N \\ n_1, n_2 = 0, 1 \end{matrix}$$

Разделяющая функция будет иметь вид

$$D_i(x) = \sum_{k=1}^N w_{kk} x_k^2 + \sum_{j=1}^{N-1} \sum_{k=j+1}^N w_{kj} x_j x_k + \sum_{j=1}^N w_j x_j + w_{i,L+1},$$

где $L = 1/2N(N+3)$

В общем случае границей для квадратичных разделяющих функций является гиперboloид. В частных случаях это будут гиперсфера, гиперэллипсоид и гиперэллипсоидальный цилиндр. Общая схема вычислителя квадратичного разделения показана на рис. 22.3

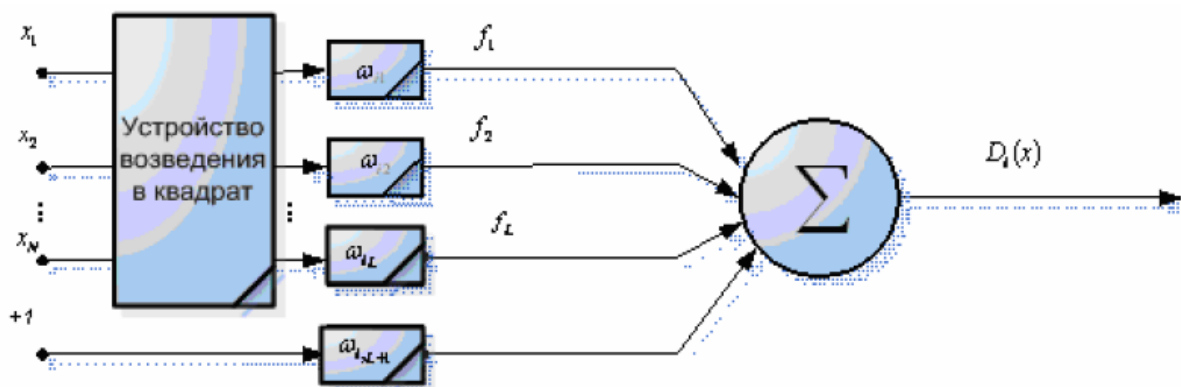


Рис. 22.3

7. Обучающий алгоритм перцептрона на 2 класса

Обучающий алгоритм для перцептрона сводится к простой схеме итеративного определения вектора весов W . Дадим краткое определение этой схемы, которую обычно называют алгоритмом перцептрона. Заданы два обучающих множества, представляющие классы ω_1 и ω_2 соответственно. Пусть $W(1)$ начальный вектор весов, который выбирается произвольно. В таком случае k -ый шаг обучения выглядит следующим образом.

Если $x(k) \in \omega_1$ и $W'(k)x(k) \leq 0$, то вектор весов $W(k)$ заменяется вектором

$$W(k+1) = W(k) + cx(k), (1)$$

где c - корректирующее приращение.

Если $x(k) \in \omega_2$ и $W'(k)x(k) \geq 0$, то $W(k)$ заменяется вектором

$$W(k+1) = W(k) - cx(k),$$

в противном случае $W(k)$ не изменяется, т.е. $W(k+1) = W(k)$

Иными словами, алгоритм вносит изменения в вектор весов W в том и только том случае, если образ предъявленный на k -м шаге был при выполнении этого шага неправильно классифицирован с помощью соответствующего вектора весов. При этом корректирующее приращение должно быть положительным. Очевидно, что алгоритм перцептрона является процедурой типа «подкрепление - наказание», причём подкреплением за правильную классификацию образа служит отсутствие наказания. Иными словами, если образ классифицирован правильно, то система подкрепляется тем, что вектор весов W не вносится никаких изменений.

С другой стороны если образ классифицирован неправильно и произведение $W'(k)x(k)$ оказывается меньше нуля, когда оно должно быть больше нуля, система наказывается увеличением значения вектора весов $W(k)$ на величину, пропорциональную $x(k)$. Точно так же, если произведение $W'(k)x(k)$ оказывается больше нуля, когда оно должно быть меньше нуля, система наказывается противоположным образом.

Сходимость алгоритма наступает при правильной классификации всех образов с помощью некоторого вектора весов. Если заданные классы линейно разделены, то алгоритм перцептрона сходится за конечное число итераций.

8. Классификатор по минимуму расстояния

Классификатор, работающий по минимуму расстояния имеет следующее решающее правило : данный образ Z принадлежит C_i , если Z ближе всего к \bar{Z}_i , $i=1,2,3$.

Пусть D_i обозначает расстояние образа Z от \bar{Z}_i . Тогда получим

$$D_i^2 = \|Z - \bar{Z}_i\|^2 = (Z - \bar{Z}_i)^T (Z - \bar{Z}_i)$$

Упрощение D_i^2 приводит к
$$D_i^2 = Z^T Z - Z^T \bar{Z}_i - \bar{Z}_i^T Z + \bar{Z}_i^T \bar{Z}_i = \|Z\|^2 - 2\left(\bar{Z}_i^T Z - \frac{1}{2}\|\bar{Z}_i\|^2\right) \quad (24.1)$$

Очевидно, что D_i^2 минимально, когда величина $\left\{\bar{Z}_i^T Z - \frac{1}{2}\|\bar{Z}_i\|^2\right\}$ максимальна. Поэтому вместо вычисления D_i^2 по формуле (24.1) проще потребовать, чтобы в классификаторе вычислялось значение $\left\{\bar{Z}_i^T Z - \frac{1}{2}\|\bar{Z}_i\|^2\right\}$.

Классификатор в этом случае описывается дискриминантными функциями.

$$\xi_i(Z) = \left\{\bar{Z}_i^T Z - \frac{1}{2}\|\bar{Z}_i\|^2\right\} \quad i=1,2,3.$$

Подстановка численных значений \bar{Z}_i и $\|\bar{Z}_i\|^2$ приводит к

$$\xi_1(Z) = -2Z_1 + 3Z_2 - 6,5$$

$$\xi_2(Z) = 6Z_1 + 5,6Z_2 - 33,68$$

$$\xi_3(Z) = 8Z_1 + 0,4Z_2 - 32,08$$

Таким образом, классификатор вычисляет три числа: $\xi_1(Z)$, $\xi_2(Z)$ и $\xi_3(Z)$, а затем сравнивает их. Классификатор относит Z к классу C_1 , если $\xi_1(Z)$ максимально , к классу C_2 , если $\xi_2(Z)$ максимально и к классу C_3 , если $\xi_3(Z)$ максимально.

9. Обобщенный алгоритм персептрона

Рассмотрим M классов $\omega_1, \omega_2, \dots, \omega_M$. Пусть на k -ой итерации процедуры обучения системе предъявляется образ $x(k)$, принадлежащий классу ω_i . Вычисляются значения M решающих функций.

$$d_j[x(k)] = W_j'(k)x(k); \quad j=1, 2, \dots, M.$$

Затем если выполняются условия

$$d_i[x(k)] > d_j[x(k)], \quad j = \overline{1, M} \quad i \neq j$$

то векторы весов не изменяются, т.е.

$$W_j(k+1) = W_j(k),$$

Допустим с другой стороны, что для некоторого l

$$d_i[x(k)] \leq d_l[x(k)]$$

В этом случае производятся следующие коррекции весов

$$W_i(k+1) = W_i(k) + cx(k)$$

$$W_l(k+1) = W_l(k) - cx(k)$$

$$W_j(k+1) = W_j(k)$$

$$j=1, 2, \dots, M; \quad j \neq i; \quad j \neq l$$

где c - положительная константа.

Если при рассмотрении этого случая классы разделимы, то этот алгоритм сходится за конечное число итераций при произвольных начальных векторах $W_i(1)$.

10 Сети встречного распространения с алгоритмами обучения

В отличие от сети с обратным распространением ошибки рассматриваемые сети предназначены для начального быстрого моделирования. Автор сети встречного распространения ошибки (ВР) Р. Хехт-Нильсен [1] удачно объединил в одной архитектуре преимущества возможности обобщения сети Т. Кохонена [2] и простоту обучения выходной звезды Д. Гроссберга [3], вследствие чего сеть ВР получила свойства, которых нет ни у одной из них в отдельности. Она принадлежит к классу сетей, обучающихся без учителя.

Сеть ВР работает с двоичными векторами, состоящими из нулей и единиц, или непрерывными. В результате обучения входные векторы ассоциируются с выходными и, когда сеть обучена, подача входных образов приводит к получению выходных. Правильный выход может быть получен и тогда, когда вход является неполным, несколько неверным. Гиперповерхность, получаемая в результате функционирования сети ВР, вследствие принципа непрерывности, дает возможность осуществлять прогнозирование. Естественно, что внутри гиперпараллелепипеда обучающих образов прогнозирование будет более точным, а при решении задачи экстраполяции ошибка будет значительно больше.

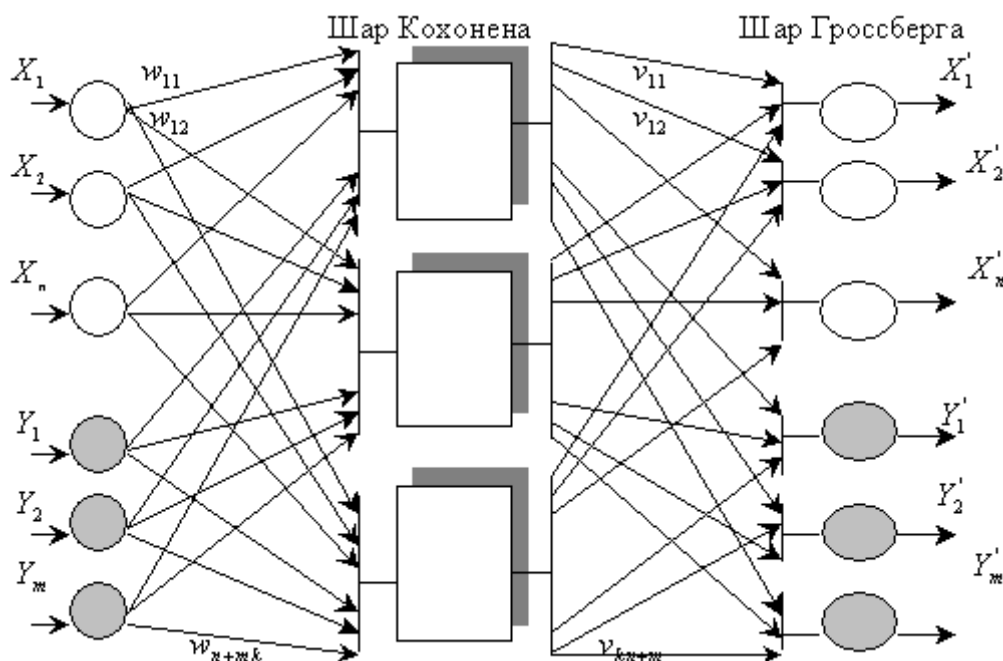


Рис. 1. Сеть встречного распространения ошибки

Подобно другим сетям сеть ВР работает в двух режимах: обучения и использования. В первом случае на входы подаем и вектор X и вектор Y , корректируем весовые коэффициенты; во втором режиме на вход уже обученной сети подаем X или Y , а с выхода снимаем и X' и Y' .

Общий метод функционирования сети: на вход подаем первый образ X , в каж-

дом нейроне слоя Кохонена рассчитывается активация $A_j = \sum_{i=1}^n w_{ij} x_i = W^T \cdot X$. Только на

выходе одного нейрона будет единица, а именно $OUT_j^X = \begin{cases} 1, & \text{якощо } A_j = \max_p A_p, \\ 0, & \text{в противном случае.} \end{cases}$ Очевидно, что веса инцидентные выигравшему нейрону максимально близки к компонентам входного вектора. Следующим шагом есть коррекция весовых коэффициентов, ребра

которых инцидентны выигравшему нейрону. Слой Гроссберга функционирует подобно слою Кохонена. Активация нейронов его слоя

$$B_j = \sum_{i=1}^m v_{ij} OUT_i^K$$

. На выходе нейронов слоя Гроссберга будут веса, инцидентные “выигравшему” нейрону слоя Кохонена

$OUT_j^G = B_j$. Далее корректируем веса, инцидентные этому нейрону и подаем на вход сети следующий образ. В результате многократного выполнения указанной процедуры весовые коэффициенты слоя Гроссберга должны совпадать или быть достаточно близкими к входным образам. Но все оказывается не так просто. Рассмотрим нюансы функционирования сети ВР в следующем алгоритме.

Алгоритм функционирования сети встречного распространения

Напомним, что исходные данные представлены в табл. 1 и один входной образ соответствует одной строке этой таблицы.

Таблица 1

X_1	X_2	...	X_n	Y_1	Y_2	...	Y_m
x_1^1	x_2^1	...	x_n^1	y_1^1	y_2^1	...	y_m^1
x_1^2	x_2^2	...	x_n^2	y_1^2	y_2^2	...	y_m^2
...
x_1^q	x_2^q	...	x_n^q	y_1^q	y_2^q	...	y_m^q

Шаг 1. Нормируем каждый элемент таблицы, так, например, нормированный аналог второго элемента первой строки рассчитываем по формуле

$$z_2^1 = \frac{x_2^1}{\sqrt{\sum_{j=1}^n (x_j^1)^2 + \sum_{j=1}^m (y_j^1)^2}}$$

и заменяем в табл. 1 каждый элемент на его нормированный аналог.

Шаг 2. Генерируем случайным образом весовые коэффициенты и обязательно их нормализуем, тем самым сокращая процесс обучения. Здесь существует много нюансов, о них можно прочитать в книге Уоссермена, а также частично см. замечание.

Шаг 3. Подаем на вход сети строку матрицы Z и рассчитываем скалярные произведения с векторами весов, которые связаны со всеми нейронами Кохонена.

Шаг 4. Среди всех скалярных произведений выбираем наибольшее и подстраиваем веса соответствующего нейрона согласно выражения $W_n = W_c + \alpha(X - W_c)$, где W_c - предыдущее значение веса, W_n - его новое значение, α - коэффициент обучения, вначале примерно равный 0,7 и постепенно уменьшающийся в процессе обучения. Выход “выигравшего” нейрона равен единице, всех остальных равен нулю.

Шаг 5. Выходной вектор слоя Кохонена подается на слой нейронов Гроссберга. В каждом нейроне слоя Гроссберга обычным образом рассчитывается активация

Шаг 6. Корректируем все веса слоя Гроссберга в соответствии с выражением:

$$V_{ij} = V_{jc} + \beta \cdot (Y_j - V_{jc}) \cdot X_i^K,$$

где Y_j - j -ая компонента реального вектора выхода, V_{ij}^* - старое значение весового коэффициента слоя Гроссберга, V_{ij}^n - новое значение,

$$x_i^k = \begin{cases} 1, & \text{если } i\text{-й нейрон слоя Кохонена "выиграл"}, \\ 0, & \text{в противном случае.} \end{cases}$$

Замечания

Некоторой оптимизацией предложенного алгоритма есть использование метода выпуклой комбинации, в одном из вариантов которого предлагается приравнять все

весовые коэффициенты одной и той же величине $w_{ij} = \frac{1}{\sqrt{n}}$. Кроме того, приравнять ка-

ждую компоненту входного вектора X : $x_i = \alpha \cdot x_i + \frac{1-\alpha}{\sqrt{n}}$. В начале обучения $\alpha = 0$ и входные векторы совпадают с весовыми коэффициентами. В процессе обучения α возрастает, постепенно приближаясь к единице. Этот метод приводит к правильному распределению входных векторов и является достаточно эффективным.

В другом подходе предлагается к истинным значениям входного вектора прибавлять шум (смещение).

В третьем подходе подстраивают все весовые коэффициенты, а не только “выигравшего” нейрона.

Возможен и такой вариант: если один из нейронов слоя Кохонена чаще становится победителем, чем другие, порог его срабатывания увеличивается

Все методы, рассмотренные выше относятся к функционированию сети в режиме аккредитации. Сеть встречного распространения может функционировать также в режиме интерполяции. Такой вариант соответствует тому, что подстраиваются весовые коэффициенты нескольких нейронов, имеющих наибольшую активацию. При этом выходы нейронов слоя Кохонена необходимо нормировать, используя, например, функцию SOFTMAX (см. книгу Уоссермена). Тогда значения выходов этих нейронов будут интерпретироваться как вероятности принадлежности входного вектора тому или иному классу.

Сеть встречного распространения рекомендуется для предварительного прогнозирования (для этого достаточно на входы обученной сети подать только вектор X и получить на выходе и X , и Y). По сравнению с сетью с обратным распространением ошибки сеть ВР обучается значительно быстрее, но точность результата меньше.

11 Ортогональные функции Хаара. Алгоритм быстрого преобразования Хаара.

Множество функций Хаара $har(n,m,t)$ (рис. 8.1), образующих периодическую, ортонормированную и полную систему функций, было предложено в 1910 году. Рекуррентное соотношение, позволяющее получить $har(n,m,t)$, имеет вид

$$har(0,0,t) = 1, \forall t \in [0,1);$$

$$har(r,m,t) = \begin{cases} 2^{r/2} & , \frac{m-1}{2^r} \leq t < \frac{m-1/2}{2^r}, \\ -2^{r/2} & , \frac{m-1/2}{2^r} \leq t < \frac{m}{2^r}, \\ 0 & , \text{при остальных } t \in [0,1), \end{cases}$$

где $0 \leq r < \log_2 N$, $1 \leq m \leq 2^r$

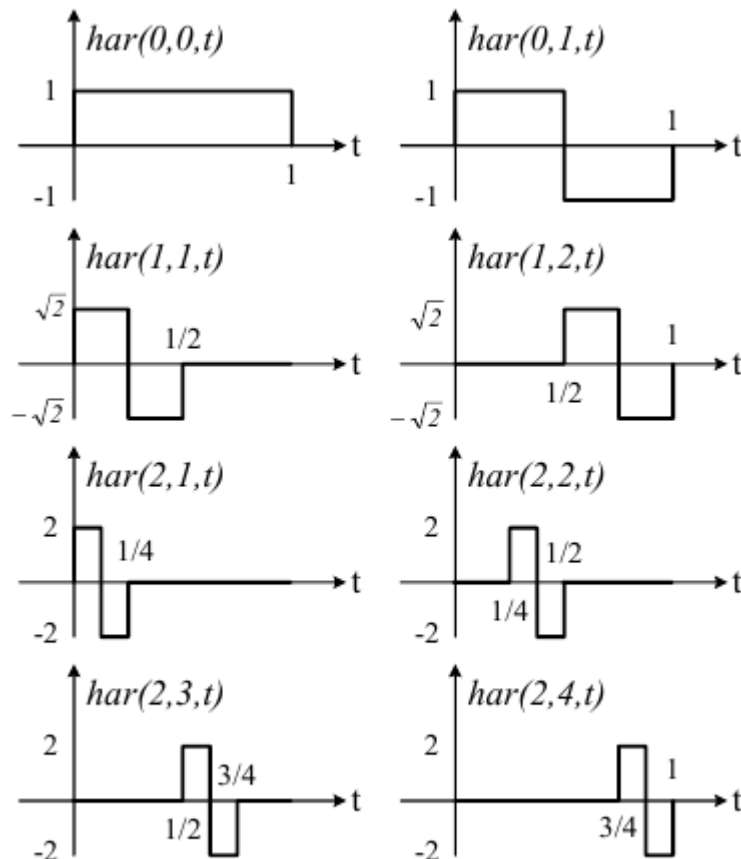


Рис. 8.1. Функции Хаара для $N=8$

Коэффициенты преобразования Хаара $Y(k)$, $k = \overline{0, N-1}$, соответствующие входной последовательности $\{X(m)\}$, получаются в результате преобразования

$$\vec{Y}(n) = \frac{1}{N} H^*(n) \vec{X}(n), \text{ где } H^*(n) \text{ – матрица Хаара размерностью } N \times N.$$

Для $N=8$

$$H^*(3) = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & -1 & -1 & -1 & -1 \\ \sqrt{2} & \sqrt{2} & -\sqrt{2} & -\sqrt{2} & & & & \\ & & & & \sqrt{2} & \sqrt{2} & -\sqrt{2} & -\sqrt{2} \\ 2 & -2 & & & & & & \\ & & 2 & -2 & & & & \\ & & & & 2 & -2 & & \\ & & & & & & 2 & -2 \end{bmatrix}.$$

Рассматривая $H^*(n)$, можно отметить, что $N/2$ коэффициентов преобразования Хаара соответствуют корреляции двух соседних точек в пространстве входных последовательностей, $N/4$ коэффициентов соответствуют связям четырех соседних точек и так далее до N/N коэффициентов, соответствующих всем N координатам пространства входных последовательностей. Это означает, что область преобразования в случае преобразования Хаара обладает свойством как локальной, так и глобальной чувствительности. При дискретном преобразовании Фурье и преобразовании Уолша-Адамара каждый коэффициент преобразования является функцией всех координат пространства входных последовательностей (свойство глобальной чувствительности), а в преобразовании Хаара это относится к первым двум коэффициентам.

Для осуществления преобразования Хаара требуется $2(N-1)$ операций сложения / вычитания и N операций умножения.

Алгоритм быстрого преобразования Хаара.

Для $N=8$ алгоритм преобразования Хаара типа Кули-Тьюки может быть представлен следующим образом. В исходной матрице преобразования Хаара необходимо переупорядочить столбцы $H(3)$, пользуясь последовательно двоичной инверсией при $N=8, 4, 2$.

Шаг 1. Переставим столбцы $H^*(3)$ в соответствии с двоичной инверсией их номеров при $N=8$, т.е. $\{0, 1, 2, 3, 4, 5, 6, 7\} \rightarrow \{0, 4, 2, 6, 1, 5, 3, 7\}$, что приведет к

$$H^*(3) = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 & 1 & -1 & 1 & -1 \\ \sqrt{2} & 0 & -\sqrt{2} & 0 & \sqrt{2} & 0 & -\sqrt{2} & 0 \\ 0 & \sqrt{2} & 0 & -\sqrt{2} & 0 & \sqrt{2} & 0 & -\sqrt{2} \\ \boxed{2} & \boxed{0} & \boxed{0} & \boxed{0} & \boxed{-2} & \boxed{0} & \boxed{0} & \boxed{0} \\ \boxed{0} & \boxed{0} & \boxed{2} & \boxed{0} & \boxed{0} & \boxed{0} & \boxed{-2} & \boxed{0} \\ \boxed{0} & \boxed{2} & \boxed{0} & \boxed{0} & \boxed{-2} & \boxed{0} & \boxed{0} & \boxed{0} \\ \boxed{0} & \boxed{0} & \boxed{0} & \boxed{2} & \boxed{0} & \boxed{0} & \boxed{0} & \boxed{-2} \end{bmatrix}.$$

0 1 2 3 0 1 2 3

Шаг 2. Переставим столбцы (4×4) матриц, заключенных в квадраты в соответствии с двоичной инверсией номеров столбцов при $N=4$, т.е. $\{0, 1, 2, 3\} \rightarrow \{0, 2, 1, 3\}$, что приведет к матрице

$$H^*(3) = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 & 1 & -1 & 1 & -1 \\ \boxed{\sqrt{2}} & \boxed{0} & \boxed{-\sqrt{2}} & \boxed{0} & \boxed{\sqrt{2}} & \boxed{0} & \boxed{-\sqrt{2}} & \boxed{0} \\ \boxed{0} & \boxed{\sqrt{2}} & \boxed{0} & \boxed{-\sqrt{2}} & \boxed{0} & \boxed{\sqrt{2}} & \boxed{0} & \boxed{-\sqrt{2}} \\ 2 & 0 & 0 & 0 & -2 & 0 & 0 & 0 \\ 0 & 2 & 0 & 0 & 0 & -2 & 0 & 0 \\ 0 & 0 & 2 & 0 & 0 & 0 & -2 & 0 \\ 0 & 0 & 0 & 2 & 0 & 0 & 0 & -2 \end{bmatrix}.$$

Шаг 3. Переставим столбцы (2×2) матриц, заключенных в квадраты в соответствии с двоичной инверсией номеров столбцов при $N=2$, т.е. $\{0,1\} \rightarrow \rightarrow \{0,1\}$, что приводит к матрице, совпадающей с $H^*(3)$.

Граф-схемы прямого и обратного быстрого преобразования Хаара представлены соответственно на рис 8.2 и 8.3.

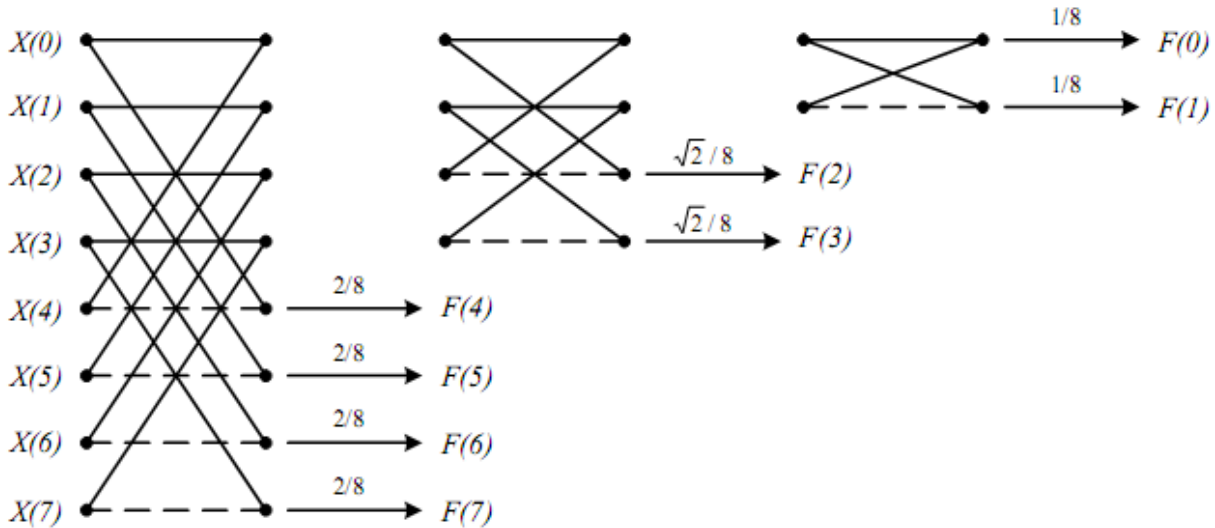


Рис. 8.2. Граф-схема прямого быстрого преобразования Хаара
Тогда спектр можно определить как

$$P_h(0) = F^2(0); \quad P_h(1) = F^2(1); \quad P_h(2) = \sum_{k=2}^3 F^2(k); \quad P_h(3) = \sum_{k=4}^7 F^2(k).$$

В общем виде можно записать $P_h(0) = F^2(0);$

$$P_h(r) = \sum_{k=2^{r-1}}^{2^r-1} F^2(k); \quad r = \overline{1, n}.$$

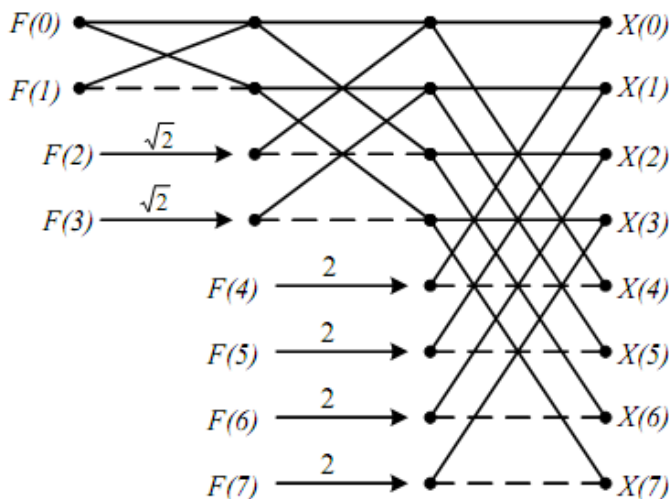
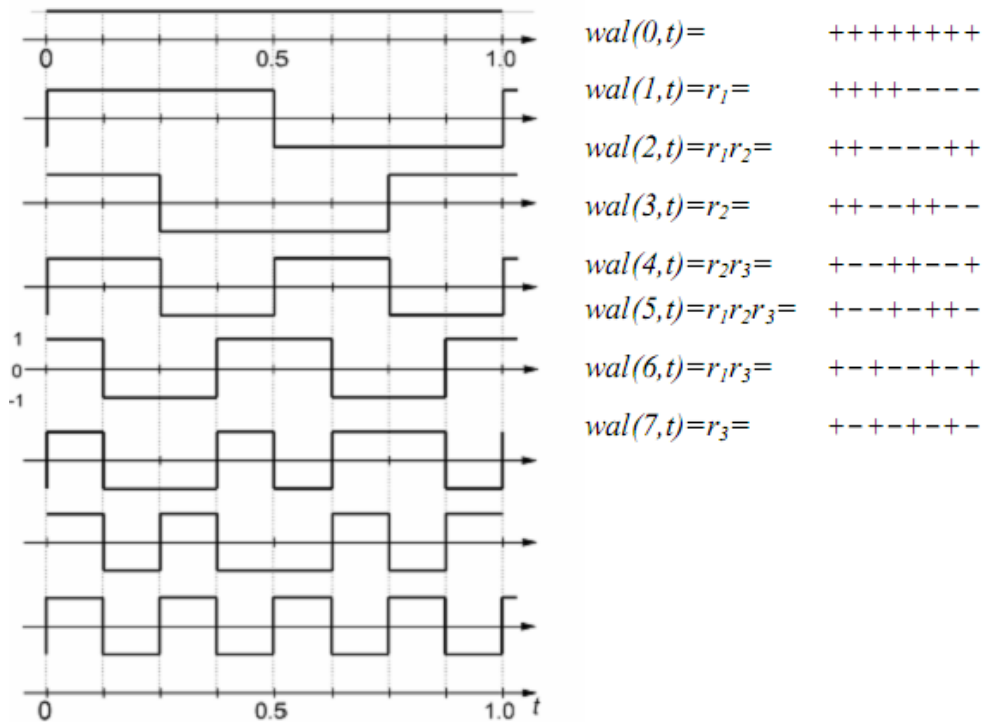


Рис. 8.3. Граф-схема быстрого обратного преобразования Хаара

12 Система ортогональных функций Уолша и ее свойства.

Функции Уолша, образующие полную ортонормированную систему, можно сформировать, образуя произведения соответствующих функций Радемахера. Первые восемь функций Уолша представлены на рис. 6.1.



Сопоставление этих функций с функциями Радемахера позволяет составить очевидные соотношения, согласно которым каждая функция Уолша $wal(n,t)$ с номером n , входящая в систему из $N = 2^r$ функций, является произведением степеней первых n функций Радемахера.

Принцип нахождения показателей этих степеней определяется двоичным представлением номера функции n . Примем следующие обозначения для разрядов, составляющих двоичное представление числа n : n_1 - первый разряд, n_2 - второй разряд, и так далее до n_r , то есть r -го разряда двоичного представления натурального числа n . При такой нумерации n_1 оказывается старшим разрядом числа n , а n_r - младшим. n_i может принимать одно из двух значений – нуль или единица. Будем считать, что $n_0 = 0$ по определению. Используя символ \oplus для обозначения операции поразрядного сложения по модулю 2, способ построения функций Уолша можно выразить аналитически для любого $N = 2^r$ в виде следующего соотношения:

$$wal(n, t) = \prod_{k=1}^r [r_k(t)]^{n_{r-k+1} \oplus n_{r-k}} \quad (6.1)$$

Поясним применение данной формулы на примере шестой функции Уолша (n=6), входящей в систему размером $N = 2^3 = 8$. Произведение состоит из трех множителей вида:

$$\text{при } k=1 \quad [r_1(t)]^{n_3 \oplus n_2},$$

$$\text{при } k=2 \quad [r_2(t)]^{n_2 \oplus n_1},$$

$$\text{при } k=3 \quad [r_3(t)]^{n_1 \oplus n_0}.$$

На основе двоичного представления числа n=6 несложно установить, что $n_1 = 1$, $n_2 = 1$, $n_3 = 0$.

Таким образом, $n_3 \oplus n_2 = 0 \oplus 1 = 1$, $n_2 \oplus n_1 = 1 \oplus 1 = 0$, $n_1 \oplus n_0 = 1 \oplus 0 = 1$ и по формуле:

$$wal(6, t) = r_1(t)r_2^0(t)r_3(t) = r_1(t)r_3(t).$$

Функции Радемахера перемножаются при использовании кода Грея. В некоторых практических приложениях, например в аналого-цифровых преобразованиях, желательно использовать коды, у которых все следующие друг за другом кодовые слова различаются только одной цифрой в некотором разряде. Коды, обладающие таким свойством, называются циклическими.

Приведем некоторые *свойства функций Уолша*.

1. Функции Уолша *ортонормированны* на интервале $0 \leq t \leq 1$:

$$\int_0^1 wal(k, t)wal(i, t)dt = \begin{cases} 1 & \text{при } k = i, \\ 0 & \text{при } k \neq i. \end{cases}$$

2. Функции Уолша обладают свойством *мультипликативности*, т.е. перемножение двух функций Уолша дает другую функцию Уолша, причем верно соотношение:

$$wal(k,t)wal(i,t) = wal(k \oplus i,t).$$

3. Функции Уолша обладают свойством *симметрии*, проявляющимся в том, что все выводы относительно i справедливы также и относительно t . Например, свойство мультипликативности с учетом свойства симметрии запишется в виде

$$wal(i,t_1)wal(i,t_2) = wal(i,t_1 \oplus t_2).$$

4. Умножение любой функции самой на себя дает функцию нулевого порядка $wal(0,t)$, так как в результате получаются только произведения вида $(+1)(+1)$ и $(-1)(-1)$. Таким образом,

$$wal(i,t)wal(i,t) = wal(0,t).$$

5. Очевидно также, что умножение $wal(i,t)$ на $wal(0,t)$ не изменяет функцию $wal(i,t)$.

$$Wal(a,t) = Wal(a,t+1) !!!!!!!$$

Способ нумерации функций в системе называется упорядочением. Функции Уолша, сформированные в соответствии с выражением (6.1), упорядочены по Уолшу.

В ряде практических задач целесообразно пользоваться иными способами упорядочения. Часто применяются функции Уолша, упорядоченные по Адамару [had(h,t)] и по Пэли [pal(p,t)].

Независимо от упорядочения функции Уолша, составляющие систему из $N = 2^r$ функций, всегда можно представить в виде произведения степеней первых r функций Радемахера. Принцип же нахождения показателей этих степеней индивидуален для каждого упорядочения.

Остановимся на упорядочении по Адамару. При $N=2^n$ матрица Адамара может быть получена с помощью соотношения

$$H(n) = \begin{bmatrix} H(n-1) & H(n-1) \\ H(n-1) & -H(n-1) \end{bmatrix};$$

$$H(0) = 1.$$

Матрица Адамара также может быть получена из ядра $H_1 = \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}$ с

помощью кронекеровского произведения, т.е.

$$H(2) = H_1 \otimes H_1 = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 \end{bmatrix}.$$

$$H(3) = H(1)*H(2)$$

13 Задачи анализа и синтеза сигналов.

Проблемы извлечения, передачи и обработки информации являются центральными для многих областей науки и техники, например, связь, автоматическое управление и регулирование, радиолокационные системы, распознавание образов. С физической точки зрения сигнал создается определенным процессом, протекающим во времени.

Важнейшими формами аналитического выражения сигнала является представление записи этого сигнала с помощью колебаний или спектра (временное или частотное представление).

Колебание $s(t)$ описывает сигнал как функцию от времени. Это основная форма аналитического выражения, поскольку t - естественная координата, с которой связаны все реальные явления. В свою очередь, сигнал можно рассматривать как совокупность элементарных функций $\eta_k(t)$, умноженных на коэффициент C_k и составляющих систему функций $\{\eta_k(t)\}$ определенного типа:

$$s(t) = \sum_{k=0}^{\infty} C_k \eta_k(t). \quad (1.1)$$

Система функций $\{\eta_k(t)\}$ носит название базисной системы, а представление сигнала в виде (1.1) называется разложением сигнала по системе базисных функций. Основными задачами теории сигналов являются:

1. анализ сигналов (изучение их свойств);
2. синтез сигналов (нахождение сигнала, обладающего заданными свойствами).

Для отображения исходной функции времени $x(t)$ в спектральной области служит, к примеру, прямое преобразование Фурье:

$$x(j\omega) = \int_0^{\infty} x(t) e^{-j\omega t} dt, \quad (1.2)$$

$$x(t) = \frac{1}{2\pi} \int_0^{\infty} x(j\omega) e^{j\omega t} d\omega. \quad (1.3)$$

Структуру, с помощью которой можно воспроизвести задачи анализа и синтеза, можно представить (рис. 1.1):

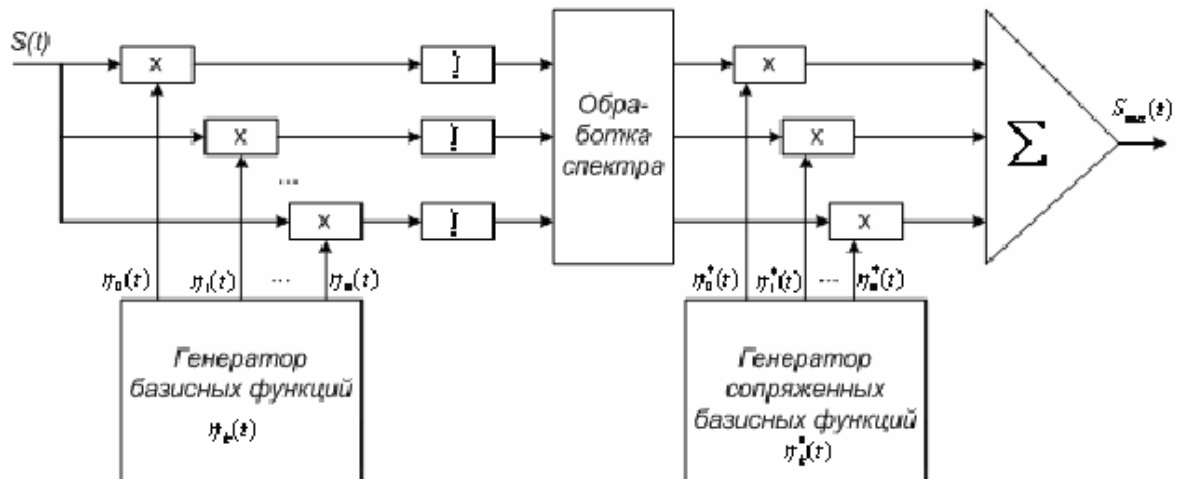


Рис. 1.1. Задачи анализа и синтеза сигналов

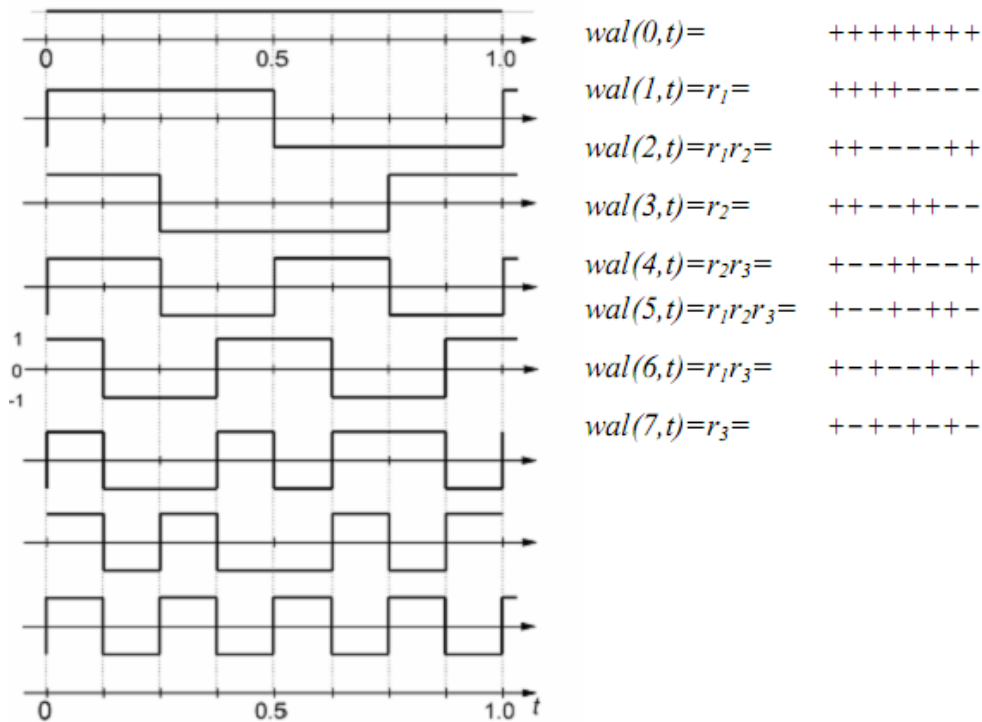
В математическом смысле задача спектрального анализа сигнала эквивалентна выполнению прямого преобразования Фурье по системе базисных функций $\eta_k(t)$, а задача синтеза – выполнению обратного преобразования Фурье по системе комплексно сопряженных $\eta_k^*(t)$. В результате спектрального анализа сигнала $s(t)$ получаем спектр $\{C_k\}$, а по спектру синтезируем новый сигнал с заданной точностью (так строятся устройства для фильтрации, сжатия данных). Реальные сигналы всегда ограничены во времени, т.к. мы не можем вести их наблюдение бесконечно долго и, как правило, являются в какой-то мере случайными.

Случайным называется сигнал, значение которого в каждый момент времени является случайной величиной.

Сигнал, известный в любой момент времени, называется *детерминированным*.

14 Система ортогональных функций Уолша и ее свойства.

Функции Уолша, образующие полную ортонормированную систему, можно сформировать, образуя произведения соответствующих функций Радемахера. Первые восемь функций Уолша представлены на рис. 6.1.



Сопоставление этих функций с функциями Радемахера позволяет составить очевидные соотношения, согласно которым каждая функция Уолша $wal(n,t)$ с номером n , входящая в систему из $N = 2^r$ функций, является произведением степеней первых n функций Радемахера.

Принцип нахождения показателей этих степеней определяется двоичным представлением номера функции n . Примем следующие обозначения для разрядов, составляющих двоичное представление числа n : n_1 - первый разряд, n_2 - второй разряд, и так далее до n_r , то есть r -го разряда двоичного представления натурального числа n . При такой нумерации n_1 оказывается старшим разрядом числа n , а n_r - младшим. n_i может принимать одно из двух значений – нуль или единица. Будем считать, что $n_0 = 0$ по определению. Используя символ \oplus для обозначения операции поразрядного сложения по модулю 2, способ построения функций Уолша можно выразить аналитически для любого $N = 2^r$ в виде следующего соотношения:

$$wal(n, t) = \prod_{k=1}^r [r_k(t)]^{n_{r-k+1} \oplus n_{r-k}} \quad (6.1)$$

Поясним применение данной формулы на примере шестой функции Уолша (n=6), входящей в систему размером $N = 2^3 = 8$. Произведение состоит из трех множителей вида:

$$\text{при } k=1 [r_1(t)]^{n_3 \oplus n_2},$$

$$\text{при } k=2 [r_2(t)]^{n_2 \oplus n_1},$$

$$\text{при } k=3 [r_3(t)]^{n_1 \oplus n_0}.$$

На основе двоичного представления числа n=6 несложно установить, что $n_1 = 1$, $n_2 = 1$, $n_3 = 0$.

Таким образом, $n_3 \oplus n_2 = 0 \oplus 1 = 1$, $n_2 \oplus n_1 = 1 \oplus 1 = 0$, $n_1 \oplus n_0 = 1 \oplus 0 = 1$ и по формуле:

$$wal(6, t) = r_1(t)r_2^0(t)r_3(t) = r_1(t)r_3(t).$$

Функции Радемахера перемножаются при использовании кода Грея. В некоторых практических приложениях, например в аналого-цифровых преобразованиях, желательно использовать коды, у которых все следующие друг за другом кодовые слова различаются только одной цифрой в некотором разряде. Коды, обладающие таким свойством, называются циклическими.

Приведем некоторые *свойства функций Уолша*.

1. Функции Уолша *ортонормированны* на интервале $0 \leq t \leq 1$:

$$\int_0^1 wal(k, t)wal(i, t)dt = \begin{cases} 1 & \text{при } k = i, \\ 0 & \text{при } k \neq i. \end{cases}$$

2. Функции Уолша обладают свойством *мультипликативности*, т.е. перемножение двух функций Уолша дает другую функцию Уолша, причем верно соотношение:

$$wal(k,t)wal(i,t) = wal(k \oplus i,t).$$

3. Функции Уолша обладают свойством *симметрии*, проявляющимся в том, что все выводы относительно i справедливы также и относительно t . Например, свойство мультипликативности с учетом свойства симметрии запишется в виде

$$wal(i,t_1)wal(i,t_2) = wal(i,t_1 \oplus t_2).$$

4. Умножение любой функции самой на себя дает функцию нулевого порядка $wal(0,t)$, так как в результате получаются только произведения вида $(+1)(+1)$ и $(-1)(-1)$. Таким образом,

$$wal(i,t)wal(i,t) = wal(0,t).$$

5. Очевидно также, что умножение $wal(i,t)$ на $wal(0,t)$ не изменяет функцию $wal(i,t)$.

$$Wal(a,t) = Wal(a,t+1) !!!!!!!$$

Способ нумерации функций в системе называется упорядочением. Функции Уолша, сформированные в соответствии с выражением (6.1), упорядочены по Уолшу.

В ряде практических задач целесообразно пользоваться иными способами упорядочения. Часто применяются функции Уолша, упорядоченные по Адамару [had(h,t)] и по Пэли [pal(p,t)].

Независимо от упорядочения функции Уолша, составляющие систему из $N = 2^r$ функций, всегда можно представить в виде произведения степеней первых r функций Радемахера. Принцип же нахождения показателей этих степеней индивидуален для каждого упорядочения.

Остановимся на упорядочении по Адамару. При $N=2^n$ матрица Адамара может быть получена с помощью соотношения

$$H(n) = \begin{bmatrix} H(n-1) & H(n-1) \\ H(n-1) & -H(n-1) \end{bmatrix};$$

$$H(0) = 1.$$

Матрица Адамара также может быть получена из ядра $H_1 = \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}$ с

помощью кронекеровского произведения, т.е.

$$H(2) = H_1 \otimes H_1 = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 \end{bmatrix}.$$

$$H(3) = H(1)*H(2)$$

15 Многослойный перцептрон с обратным распространением ошибки.

Многослойный перцептрон является сетью с прямым распространением сигнала (без обратных связей), обучаемой с учителем. Такая сеть способна аппроксимировать любую непрерывную функцию или границу между классами со сколь угодно высокой точностью. Для этого достаточно одного скрытого слоя нейронов с сигмоидной функцией активации (2) – (4), т.е. многослойный перцептрон обычно состоит из 3 слоев: первого распределительного, второго скрытого и третьего выходного (рис. 2.1).

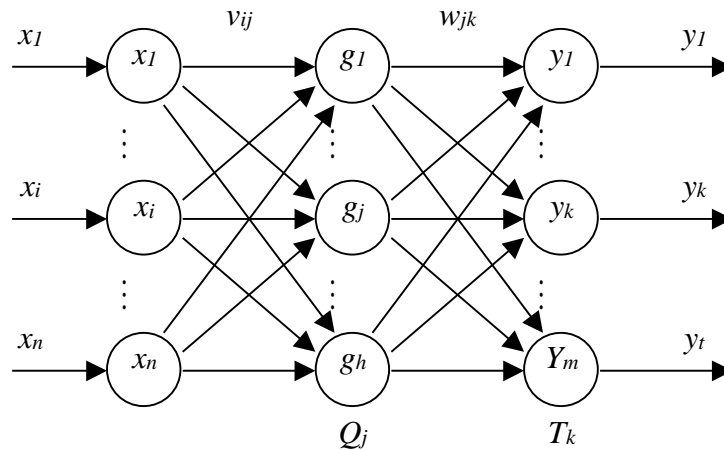


Рис. 2.1. Многослойный перцептрон

Такая сеть имеет n входов и n нейронов распределительного слоя, h нейронов скрытого слоя и m выходных нейронов. Используются две матрицы весов: скрытого слоя v размером $n \times h$ и выходного слоя w размером $h \times m$. Кроме этого, с каждым слоем нейронов связан массив порогов: Q – для скрытого слоя, T – для выходного. Эти данные представляют собой знания сети, настраиваемые в процессе обучения и определяющие ее поведение. Перцептрон функционирует по следующим формулам:

$$g_j = f\left(\sum_{i=1}^n v_{ij} x_i + Q_j\right), \quad (2.1) \quad y_k = f\left(\sum_{j=1}^h w_{jk} g_j + T_k\right). \quad (2.2)$$

В качестве функции активации используется одна из функций (2) – (4). Вид функции определяет диапазон чисел, в котором работает сеть. В дальнейшем будет использоваться сигмоидная функция (2), имеющая область значений от 0 до 1.

Обучение с учителем ставит перед сетью задачу обобщить p примеров, заданных парами векторов (x^r, y^r) , $r = \overline{1, p}$. Вектор $x^r = (x_1^r, x_2^r, \dots, x_i^r, \dots, x_n^r)$ в случае задачи классификации задает входной образ (вектор признаков), а вектор $y^r = (y_1^r, y_2^r, \dots, y_k^r, \dots, y_m^r)$, задающий эталонный выход, должен кодировать номер класса. При этом есть множество вариантов кодирования. Оптимальным представляется кодирование, когда номер класса определяется позицией единичной компоненты в векторе y^r , а все остальные компоненты равны 0. Каждый выходной нейрон соответствует одному классу. Такой способ позволяет при классификации определять вероятность каждого класса по величине на выходе соответствующего нейрона (чем ближе к единице, тем вероятность больше).

Обучение перцептрона проводится с помощью алгоритма обратного распространения ошибки, который минимизирует среднеквадратичную ошибку нейронной сети. Для этого с целью настройки синаптических связей используется метод градиентного спуска в пространстве весовых коэффициентов и порогов нейронной сети. Рассмотрим алгоритм обратного распространения ошибки.

1. Происходит начальная инициализация знаний сети. Простейший вариант такой инициализации – присвоить всем весам и порогам случайные значения из диапазона $[-1, 1]$.

2. Для каждой пары векторов (x^r, y^r) выполняется следующее:

2.1. Для входного вектора рассчитываются выходы нейронов скрытого слоя и выходы сети по формулам (2.1), (2.2).

2.2. Происходит коррекция знаний сети, при этом главное значение имеет отклонение реально полученного выхода сети y от идеального вектора y^r . Согласно методу градиентного спуска, изменение весовых коэффициентов и порогов нейронной сети происходит по следующим формулам:

$$w_{jk}(t+1) = w_{jk}(t) + \alpha \frac{\partial E}{\partial w_{jk}(t)}, \quad (2.3) \quad T_k(t+1) = T_k(t) + \alpha \frac{\partial E}{\partial T_k(t)}, \quad (2.4)$$

где E - среднеквадратичная ошибка нейронной сети для одного образа, а α – параметр, определяющий скорость обучения. Формулы записаны в терминах выходного слоя, аналогично выглядят формулы для скрытого слоя. Среднеквадратичная ошибка сети вычисляется как

$$E = \frac{1}{2} \sum_{k=1}^m (y_k^r - y_k)^2. \quad (2.5)$$

Ошибка k -го нейрона выходного слоя определяется как

$$d_k = \frac{\partial E}{\partial y_k} = y_k^r - y_k. \quad (2.6)$$

Выразим производные из формул (2.3), (2.4) через легко вычисляемые величины. Определим взвешенную сумму, аргумент функции активации как

$$S_k = \sum_{j=1}^h w_{jk} g_j + T_k. \quad (2.7)$$

Из соотношения (3.3) $\frac{\partial E}{\partial w_{jk}}$ можно представить как:

$$\frac{\partial E}{\partial w_{jk}} = \frac{\partial E}{\partial y_k} \frac{\partial y_k}{\partial S_k} \frac{\partial S_k}{\partial w_{jk}}, \quad (2.8)$$

где $\frac{\partial E}{\partial y_k} = d_k = y_k^r - y_k$ – ошибка k -го нейрона; $\frac{\partial y_k}{\partial S_k} = f'(S)$ – производная функции активации;

$\frac{\partial S_k}{\partial w_{jk}} = g_j$ – значение j -го нейрона предыдущего слоя. Получаем

$$\frac{\partial E}{\partial w_{jk}} = d_k f'(S_k) g_j. \quad (2.9)$$

Аналогично (2.8), с учетом того, что $\frac{\partial S_k}{\partial T_k} = 1$, получаем

$$\frac{\partial E}{\partial T_k} = \frac{\partial E}{\partial y_k} \frac{\partial y_k}{\partial S_k} \frac{\partial S_k}{\partial T_k} = d_k f'(S_k). \quad (2.10)$$

Веса и пороги скрытого слоя также корректируются по формулам, аналогичным (2.3), (2.4), с учетом (2.9) и (2.10). При этом главной трудностью является определение ошибки нейрона скрытого слоя. Эту ошибку явно определить по формуле, аналогичной (2.6), невозможно, однако существует возможность рассчитать ее через ошибки нейронов выходного слоя (отсюда произошло название алгоритма обратного распространения ошибки):

$$e_j = \frac{\partial E}{\partial g_j} = \sum_{k=1}^m \frac{\partial E}{\partial y_k} \frac{\partial y_k}{\partial S_k} \frac{\partial S_k}{\partial g_j} = \sum_{k=1}^m d_k f'(S_k) w_{jk}. \quad (2.11)$$

Производные от функций активации тоже легко рассчитываются, например, для сигмоидной функции получаем

$$f'(S_k) = \left(\frac{1}{1 + e^{-S}} \right)' = f(S_k)(1 - f(S_k)) = y_k(1 - y_k). \quad (2.12)$$

Аналогично

$$(th(S_k))' = 1 - y_k^2;$$

$$\left(\frac{2}{1 + e^{-x}} - 1 \right)' = \frac{(1 - y_k^2)}{2}.$$

Таким образом, можно записать окончательные выражения (2.3), (2.4) для двух слоев, использующих сигмоидную функцию:

$$w_{jk} = w_{jk} + \alpha y_k (1 - y_k) d_k g_j, \quad (2.13)$$

$$T_k = T_k + \alpha y_k (1 - y_k) d_k, \quad (2.14)$$

$$v_{ij} = v_{ij} + \beta g_j (1 - g_j) e_j x_i, \quad (2.15)$$

$$Q_j = Q_j + \beta g_j (1 - g_j) e_j. \quad (2.16)$$

В эти формулы вводится дополнительный параметр β – скорость обучения скрытого слоя, который может отличаться от аналогичного параметра для выходного слоя. Рекомендуется изменять скорости обучения обратно пропорционально количеству шагов алгоритма обучения, однако это не всегда оправдывает себя на практике.

3. После того как коррекция знаний произведена для каждой пары векторов, можно оценить степень успешности обучения сети для определения момента завершения алгоритма. Для этого можно использовать в качестве критерия максимальную по модулю ошибку на выходе d_k , полученную на шаге 2. Условием прекращения обучения в этом случае будет

$$\max |d_k| < D, k = \overline{1, m}, r = \overline{1, p}, \quad (2.17)$$

где D – достаточно маленькая константа – величина максимальной ошибки, которую требуется достичь в процессе обучения. Если условие (2.17) не выполняется, то шаг 2 повторяется.

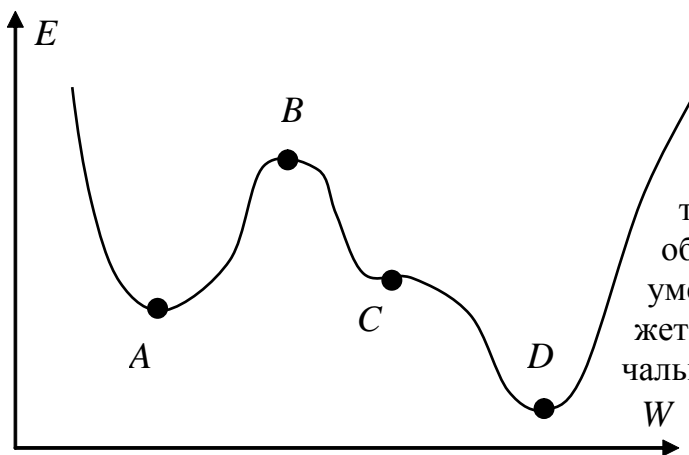
Способность персептрона-классификатора разделять образы в пространстве признаков прежде всего зависит от его скрытого слоя. Именно на этот слой возлагается задача сделать множество классов линейно разделимым для успешной работы выходного слоя. Очевидно, что чем больше нейронов в скрытом слое, тем большее количество примеров этот слой может разделять. Кроме этого, увеличение числа признаков входных образов также способствует успешному их разделению в пространстве признаков. Однако увеличение этих параметров приводит к росту ошибок сети и времени обучения. Увеличение размерности входов n приводит к росту ошибки аппроксимации сети, возникающей из-за обобщения данных. Увеличение числа нейронов h скрытого слоя приводит к росту ошибки, связанной со сложностью модели. Персептрону легче провести функцию через эталонные точки, однако при этом обобщающая способность сети ухудшается. Он хуже предсказывает поведение аппроксимируемой функции на образах, не входящих в обучающую выборку. Такое состояние сети называется *переобучением*.

Оптимальное соотношение между этими параметрами оценивают как

$$h \sim \sqrt{\frac{p}{n}}. \quad (2.18)$$

Эксперименты показывают, что обучение максимально успешно проходит на множестве классов, хорошо (желательно линейно) разделенных в пространстве признаков. Это достигается удачным подбором информативных признаков. Если классы в пространстве признаков хорошо кластеризуются, т.е. образы каждого класса составляют компактную группу, достаточно удаленную от других групп, то есть возможность уменьшить размер обучающей выборки p (используются только центры кластеров) и затем уменьшить число нейронов h . Это приводит к ускорению обучения и улучшает работу классификатора.

Другой проблемой является то, что алгоритм градиентного спуска не гарантирует нахождение глобального минимума среднеквадратичной ошибки сети (2.5), а гарантируется определение только локального минимума. Проблемы, возникающие в процессе градиентного спуска, можно проанализировать на примере функции ошибки, схематически изображенной на рис. 2.2.



Существуют многочисленные способы оптимизации метода градиентного спуска, призванные улучшить поведение алгоритма в подобных критических точках. Эффективной модификацией является введение момента, накапливающего влияние градиента на веса со временем. Тогда величина Δw (2.3) в момент времени t будет вычисляться как

$$\Delta w(t) = \alpha \frac{\partial E}{\partial w} + \mu \Delta w(t - 1), \quad (2.19)$$

где μ – параметр, определяющий величину влияния момента. С использованием (2.19) скорость изменения весов возрастает на участках с постоянным знаком производной. В окрестностях минимума скорость резко падает за счет колебания знака.

Достоинства алгоритма – большая скорость в точках перегиба, возможность по инерции преодолевать небольшие локальные минимумы. Недостатки – еще один параметр, величину которого следует подбирать и настраивать.

Этот и другие алгоритмы оптимизации обучения персептрона позволяют улучшить работу сети в условиях плохой сходимости. Однако они усложняют процесс обучения, не гарантируя в то же время полного успеха во всех случаях. Успех обучения классификатора зависит от самого алгоритма обучения и качества обучающей выборки.

16 Основные операторы обработки изображений.

Сегментация означает выделение областей, однородных по своим яркостным, цветовым или текстурным свойствам, либо посредством выделения их границ, либо путем разметки внутренних точек. В свою очередь, можно выделить два класса, которыми определяются методы выделения границ: пространственное дифференцирование и высокочастотная фильтрация. Причем общим для указанных методов является стремление рассматривать границу (край), как область резкого перепада функции яркости изображения.

Метод пространственного дифференцирования основан на предположении о том, что граничные (краевые) точки имеют большую величину модуля градиента функции $f(x,y)$. Градиент изображения $f(x,y)$ в точке (x,y) определяется как двумерный вектор:

$$\overline{G[f(x,y)]} = \begin{bmatrix} G_x \\ G_y \end{bmatrix} = \begin{bmatrix} \frac{\partial f}{\partial x} \\ \frac{\partial f}{\partial y} \end{bmatrix} \quad (17.1)$$

Из векторного анализа известно, что вектор G указывает направление максимального изменения функции f в точке (x,y) . Величина этого вектора определяется как:

$$G[f(x,y)] = [G_x^2 + G_y^2]^{1/2} = [(\frac{\partial f}{\partial x})^2 + (\frac{\partial f}{\partial y})^2]^{1/2} \quad (17.2)$$

На практике, как правило, градиент аппроксимируется абсолютными значениями

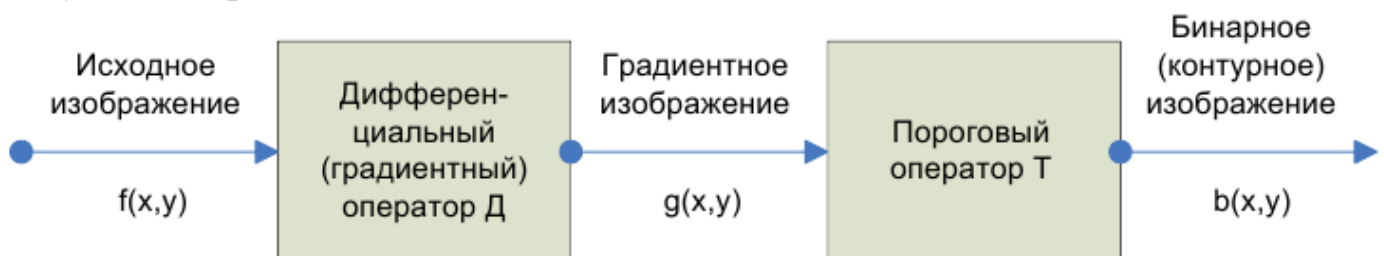
$$G[f(x,y)] \approx |G_x| + |G_y| \quad (17.3)$$

Как видно из указанных соотношений, вычисление градиента основано на нахождении первых производных $\partial f/\partial x$, $\partial f/\partial y$. Для цифрового изображения это можно сделать несколькими путями. Один из подходов состоит в использовании разности между соседними пикселями.

$$G_x = \partial f/\partial x = f(x,y) - f(x-1,y) \quad (17.4)$$

$$G_y = \partial f/\partial y = f(x,y) - f(x,y-1)$$

Структура метода пространственного дифференцирования может быть представлена следующим образом



Получаемое после преобразования D поле $g(x,y)$ называется градиентным изображением или изображением с усиленными границами. Обработка градиентного изображения осуществляется с помощью порогового оператора согласно правилу:

$$b(x, y) = \begin{cases} 1 & \text{при } g(x, y) \geq T \\ 0 & \text{при } g(x, y) < T \end{cases}$$

где T – величина постоянного или меняющегося по определенному правилу порога.

Вычислительная реализация данного метода сводится к синтезу численных алгоритмов оценки частных производных в некоторой точке изображения и выполнению далее несложных арифметических операций.

Пусть через $f(m, n)$ и $g(m, n)$, где $m, n=0, N-1$ обозначены дискретизированные исходное и градиентное изображения и в качестве дискретного аналога соотношения (17.2) выступает:

$$g(m, n) = [d_1^2(m, n) + d_2^2(m, n)]^{1/2}$$

или его упрощенный вычислительный вариант:

$$g(m, n) = |d_1(m, n)| + |d_2(m, n)|$$

В этих соотношениях $d_1(m, n)$ и $d_2(m, n)$ обозначают оценки частных производных функции яркости в точке (m, n) . Обозначим через F_{mn} и F_{mn} векторы

$$(F_{mn})^T = (f_{mn}, f_{m,n+1}, f_{m+1,n}, f_{m+1,n+1})$$

$$(F_{mn})^T = (f_{m-1,n-1}, f_{m-1,n}, f_{m-1,n+1}, f_{m,n-1}, f_{m,n}, f_{m,n+1}, f_{m+1,n-1}, f_{m+1,n}, f_{m+1,n+1}),$$

представляющие собой упорядоченные по строкам элементы окрестности точки (m, n) , размерностью $2*2$ и $3*3$.

В качестве масок (шаблонов) можно использовать **разностный оператор**:

$$W_1 = \begin{bmatrix} 1 & -1 \\ 1 & -1 \end{bmatrix}; W_2 = \begin{bmatrix} -1 & -1 \\ 1 & 1 \end{bmatrix} \quad d_1(m, n) = [f(m, n) - f(m, n+1)] + [f(m+1, n) - f(m+1, n+1)] \\ d_2(m, n) = [f(m+1, n) - f(m, n)] + [f(m+1, n+1) - f(m, n+1)]$$

Оператор Робертса

$$W_1 = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}; W_2 = \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix} \quad d_1(m, n) = [f(m, n) - f(m+1, n+1)] \\ d_2(m, n) = [f(m+1, n) - f(m, n+1)]$$

$$W_1 = \begin{bmatrix} 1 & 0 & -1 \\ 1 & 0 & -1 \\ 1 & 0 & -1 \end{bmatrix}; W_2 = \begin{bmatrix} -1 & -1 & -1 \\ 0 & 0 & 0 \\ 1 & 1 & 1 \end{bmatrix} \quad \text{Оператор Превитта}$$

$$d_1(m, n) = [f(m-1, n-1) + f(m, n-1) + f(m+1, n-1)] - [f(m-1, n+1) + f(m, n+1) + f(m+1, n+1)] \\ d_2(m, n) = [f(m+1, n-1) + f(m+1, n) + f(m+1, n+1)] - [f(m-1, n-1) + f(m-1, n) + f(m-1, n+1)]$$

$$W_1 = \begin{bmatrix} 1 & 0 & -1 \\ 2 & 0 & -2 \\ 1 & 0 & -1 \end{bmatrix}; W_2 = \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix} \quad \text{Оператор Собеля}$$

$$d_1(m, n) = [f(m-1, n-1) + 2f(m, n-1) + f(m+1, n-1)] - [f(m-1, n+1) + 2f(m, n+1) + f(m+1, n+1)] \\ d_2(m, n) = [f(m+1, n-1) + 2f(m+1, n) + f(m+1, n+1)] - [f(m-1, n-1) + 2f(m-1, n) + f(m-1, n+1)]$$

17 Метод пространственной области применительно к предварительной обработке изображений.

К пространственной области относится совокупность пикселей, составляющих изображение. Функция предварительной обработки в пространственной области записывается в виде:

$$g(x,y) = h[f(x,y)],$$

где $f(x,y)$ - входное изображение, $g(x,y)$ – выходное (обработанное) изображение, h – оператор функции f , определенный в некоторой области (x,y) .

Основным подходом при определении окрестности точки (x,y) является использование квадратной или прямоугольной области части изображения с центром в точке (x,y) . Центр этой части изображения перемещается от пиксела к пикселу, начиная, например, от левого верхнего угла. При этом для получения $g(x,y)$ оператор применяется для каждого положения (x,y) . Хотя используются иногда и другие формы окрестности (например, круг), квадратные формы более предпочтительны из-за простоты реализации. Один из наиболее применяемых методов пространственной области основан на использовании так называемых масок свертки (шаблонов, окон, фильтров). Обычно маска представляет собой небольшую (например, размерность 3×3) двумерную систему, коэффициенты которой выбираются таким образом, чтобы обнаружить заданное свойство изображения.

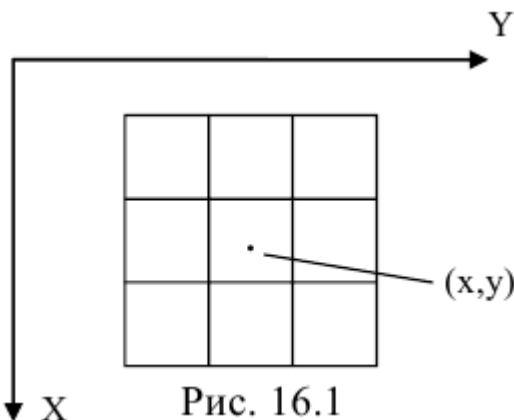


Рис. 16.1

-1	-1	-1
-1	8	-1
-1	-1	-1

Рис. 16.2

На рис 16.1 представлена окрестность размерностью 3×3 точки (x,y) изображения.

Предположим для начала, что дано изображение с постоянной интенсивностью, которое содержит отдельные, удаленные друг от друга пиксели с отличной от фона интенсивностью.

Эти точки могут быть обнаружены маской, показанной на рис. 16.2

Процесс заключается в следующем. Центр маски, помеченный цифрой 8, перемещается по изображению определенным образом. При совпадении центра маски с положением каждого пиксела производится умножение значений всех пикселей, находящихся под маской, на соответствующий коэффициент на маске, т.е. значение пиксела под центром маски умножается на 8, а значения восьми соседних пикселей умножаются на -1. Затем результаты этих умножений суммируются. Если все пиксели под маской имеют одинаковые значения (постоянный фон), то сумма будет равна нулю. Если же центр маски разместится над точкой с другой интенсивностью, сумма будет отлична от нуля. В случае размещения указанной точки вне центра, сумма также будет отлична от нуля, но на меньшую величину. Это различие может быть устранено путем сравнения значения суммы с пороговым значением.

Если величины w_1, w_2, \dots, w_9 представляют собой коэффициенты, маски пиксела (x,y) и его восьми соседей (рис.16.3), то описанный выше алгоритм можно представить как выполнение следующей операции на окрестности 3×3 точки (x,y) .

w_1 $(x-1, y-1)$	w_2 $(x-1, y)$	w_3 $(x-1, y+1)$
w_4 $(x, y-1)$	w_5 (x, y)	w_6 $(x, y+1)$
w_7 $(x+1, y-1)$	w_8 $(x+1, y)$	w_9 $(x+1, y+1)$

$$h[f(x,y)] = w_1 f(x-1,y-1) + w_2 f(x-1,y) + w_3 f(x-1,y+1) + w_4 f(x,y-1) + w_5 f(x,y) + w_6 f(x,y+1) + w_7 f(x+1,y-1) + w_8 f(x+1,y) + w_9 f(x+1,y+1)$$

(16.1)

Рис. 16.3

Методы частотной области используют для своей реализации алгоритмы прямого и обратного двумерного преобразований.

$$F(u, v) = \frac{1}{N^2} \sum_{x=0}^{N-1} \sum_{y=0}^{N-1} f(x, y) e^{-j2\pi(ux+vy)/N}$$

$$f(x, y) = \sum_{U=0}^{N-1} \sum_{V=0}^{N-1} F(u, v) e^{j2\pi(ux+vy)/N}$$

(16.2)

При вычислении прямого двумерного преобразования Фурье, сначала с использованием техники одномерного преобразования Фурье оперируют строками массива изображения, а затем одномерные преобразования осуществляются над столбцами (построчно-столбцовый алгоритм).

Операция сглаживания используется для снижения шума и других помех, которые могут появиться на изображении в результате его дискретизации и квантования.

Усреднение окрестности является прямым методом пространственной области для сглаживания изображения. Для изображения $f(x,y)$ процесс заключается в получении сглаженного изображения $g(x,y)$, интенсивность которого в каждой точке (x,y) равна усредненному значению интенсивности пикселей функции f , содержащихся в заданной окрестности точки (x,y) . Иными словами сглаженное изображение получается с использованием соотношения:

$$g(x, y) = \frac{1}{P} \sum_{n,m \in S} f(n, m)$$

для всех x,y функции $f(x,y)$, где p - общее число точек в окрестности.

Для окрестности размерностью $3*3$ указанное соотношение является частным случаем выражения (16.1) при $w_i=1/9$.

18 Сегментация изображений посредством выделения границ областей.

Сегментация означает выделение областей, однородных по своим яркостным, цветовым или текстурным свойствам, либо посредством выделения их границ, либо путем разметки внутренних точек. В свою очередь, можно выделить два класса, которыми определяются методы выделения границ: пространственное дифференцирование и высокочастотная фильтрация. Причем общим для указанных методов является стремление рассматривать границу (край), как область резкого перепада функции яркости изображения.

Метод пространственного дифференцирования основан на предположении о том, что граничные (краевые) точки имеют большую величину модуля градиента функции $f(x,y)$. Градиент изображения $f(x,y)$ в точке (x,y) определяется как двумерный вектор:

$$\overline{G[f(x,y)]} = \begin{bmatrix} G_x \\ G_y \end{bmatrix} = \begin{bmatrix} \frac{\partial f}{\partial x} \\ \frac{\partial f}{\partial y} \end{bmatrix} \quad (17.1)$$

Из векторного анализа известно, что вектор G указывает направление максимального изменения функции f в точке (x,y) . Величина этого вектора определяется как:

$$G[f(x,y)] = [G_x^2 + G_y^2]^{1/2} = [(\frac{\partial f}{\partial x})^2 + (\frac{\partial f}{\partial y})^2]^{1/2} \quad (17.2)$$

На практике, как правило, градиент аппроксимируется абсолютными значениями

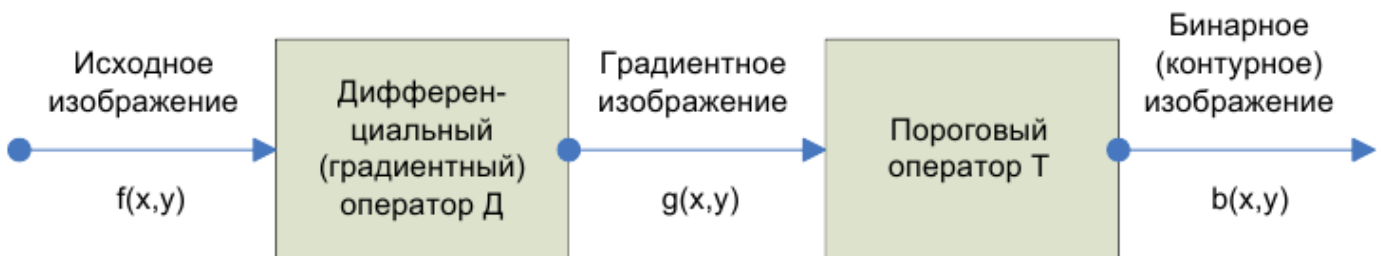
$$G[f(x,y)] \approx |G_x| + |G_y| \quad (17.3)$$

Как видно из указанных соотношений, вычисление градиента основано на нахождении первых производных $\partial f/\partial x$, $\partial f/\partial y$. Для цифрового изображения это можно сделать несколькими путями. Один из подходов состоит в использовании разности между соседними пикселями.

$$G_x = \partial f/\partial x = f(x,y) - f(x-1,y) \quad (17.4)$$

$$G_y = \partial f/\partial y = f(x,y) - f(x,y-1)$$

Структура метода пространственного дифференцирования может быть представлена следующим образом



Получаемое после преобразования D поле $g(x,y)$ называется градиентным изображением или изображением с усиленными границами. Обработка градиентного изображения осуществляется с помощью порогового оператора согласно правилу:

$$b(x, y) = \begin{cases} 1 & \text{при } g(x, y) \geq T \\ 0 & \text{при } g(x, y) < T \end{cases}$$

где T – величина постоянного или меняющегося по определенному правилу порога. Вычислительная реализация данного метода сводится к синтезу численных алгоритмов оценки частных производных в некоторой точке изображения и выполнению далее несложных арифметических операций.

Пусть через $f(m, n)$ и $g(m, n)$, где $m, n=0, N-1$ обозначены дискретизированные исходное и градиентное изображения и в качестве дискретного аналога соотношения (17.2) выступает:

$$g(m, n) = [d_1^2(m, n) + d_2^2(m, n)]^{1/2}$$

или его упрощенный вычислительный вариант:

$$g(m, n) = |d_1(m, n)| + |d_2(m, n)|$$

В этих соотношениях $d_1(m, n)$ и $d_2(m, n)$ обозначают оценки частных производных функции яркости в точке (m, n) . Обозначим через F_{mn} и F_{mn} векторы

$$(F_{mn})^T = (f_{mn}, f_{m,n+1}, f_{m+1,n}, f_{m+1,n+1})$$

$$(F_{mn})^T = (f_{m-1,n-1}, f_{m-1,n}, f_{m-1,n+1}, f_{m,n-1}, f_{m,n}, f_{m,n+1}, f_{m+1,n-1}, f_{m+1,n}, f_{m+1,n+1}),$$

представляющие собой упорядоченные по строкам элементы окрестности точки (m, n) , размерностью $2*2$ и $3*3$.

В качестве масок (шаблонов) можно использовать **разностный оператор**:

$$W_1 = \begin{bmatrix} 1 & -1 \\ 1 & -1 \end{bmatrix}; W_2 = \begin{bmatrix} -1 & -1 \\ 1 & 1 \end{bmatrix} \quad d_1(m, n) = [f(m, n) - f(m, n+1)] + [f(m+1, n) - f(m+1, n+1)] \\ d_2(m, n) = [f(m+1, n) - f(m, n)] + [f(m+1, n+1) - f(m, n+1)]$$

Оператор Робертса

$$W_1 = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}; W_2 = \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix} \quad d_1(m, n) = [f(m, n) - f(m+1, n+1)] \\ d_2(m, n) = [f(m+1, n) - f(m, n+1)]$$

$$W_1 = \begin{bmatrix} 1 & 0 & -1 \\ 1 & 0 & -1 \\ 1 & 0 & -1 \end{bmatrix}; W_2 = \begin{bmatrix} -1 & -1 & -1 \\ 0 & 0 & 0 \\ 1 & 1 & 1 \end{bmatrix} \quad \text{Оператор Превитта}$$

$$d_1(m, n) = [f(m-1, n-1) + f(m, n-1) + f(m+1, n-1)] - [f(m-1, n+1) + f(m, n+1) + f(m+1, n+1)] \\ d_2(m, n) = [f(m+1, n-1) + f(m+1, n) + f(m+1, n+1)] - [f(m-1, n-1) + f(m-1, n) + f(m-1, n+1)]$$

$$W_1 = \begin{bmatrix} 1 & 0 & -1 \\ 2 & 0 & -2 \\ 1 & 0 & -1 \end{bmatrix}; W_2 = \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix} \quad \text{Оператор Собеля}$$

$$d_1(m, n) = [f(m-1, n-1) + 2f(m, n-1) + f(m+1, n-1)] - [f(m-1, n+1) + 2f(m, n+1) + f(m+1, n+1)] \\ d_2(m, n) = [f(m+1, n-1) + 2f(m+1, n) + f(m+1, n+1)] - [f(m-1, n-1) + 2f(m-1, n) + f(m-1, n+1)]$$

19 Алгоритм преобразования двоичного кода в код Грея.

В некоторых практических приложениях, например в аналого-цифровых преобразованиях, желательно использовать коды, у которых все следующие друг за другом кодовые слова различаются только одной цифрой в некотором разряде. Коды, обладающие таким свойством, называются циклическими.

Очень важным циклическим кодом является код Грея. Двоичное представление числа может быть легко преобразовано в код Грея с помощью полусумматоров.

Пусть $g_{n-1}g_{n-2}...g_2g_1g_0$ – кодовое слово в n-разрядном двоичном коде Грея, соответствующее двоичному числу $b_{n-1}b_{n-2}...b_2b_1b_0$. Тогда g_i может быть получена как

$$g_i = b_i \oplus b_{i+1}, 0 \leq i \leq n-2;$$

$$g_{n-1} = b_{n-1},$$

где \oplus означает сложение по модулю два, которое определяется как

$$0 \oplus 0 = 0 \quad 0 \oplus 1 = 1$$

$$1 \oplus 0 = 1 \quad 1 \oplus 1 = 0$$

Например, код Грея, соответствующий двоичному числу 101101, может быть образован как на рис. 6.2.

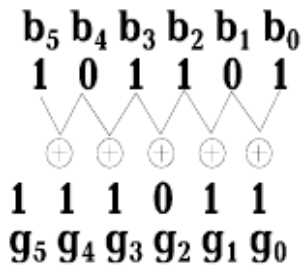


Рис. 6.2. Преобразование двоичного кода в код Грея

Трехразрядный код Грея

Десятичное число	Код Грея			Двоичный код		
	g_2	g_1	g_0	b_2	b_1	b_0
0	0	0	0	0	0	0
1	0	0	1	0	0	1
2	0	1	1	0	1	0
3	0	1	0	0	1	1
4	1	1	0	1	0	0
5	1	1	1	1	0	1
6	1	0	1	1	1	0
7	1	0	0	1	1	1

Преобразование кода Грея в двоичный код начинается с цифры самого левого разряда и движения вправо, принимая $b_i = g_i$, если число единиц, предшествующих g_i четно и $b_i = \bar{g}_i$ (черта обозначает инвертирование), если число единиц, предшествующих g_i нечетно. При этом нулевое число единиц считается четным. Пример двоичного числа, соответствующее коду Грея 1001011, показан на рис. 6.3.

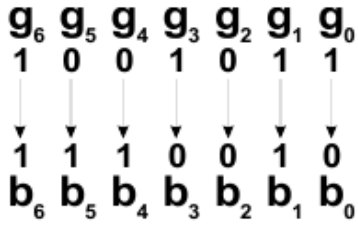


Рис. 6.3. Преобразование кода Грея в двоичный код

20 Теорема корреляции

Корреляция, так же, как свертка, использует два сигнала для получения третьего. Этот третий сигнал называется корреляционным сигналом двух входных сигналов.

Если $\{X(m)\}$ и $\{Y(m)\}$ – последовательности действительных чисел, для которых $X(m) \leftrightarrow C_x(k)$, $Y(m) \leftrightarrow C_y(k)$, а их функция корреляции определяется соотношением

$$\hat{Z}(m) = \frac{1}{N} \sum_{h=0}^{N-1} X(h)Y(m+h), \text{ где } m = \overline{0, N-1}, \quad (3.3)$$

то $C_{\hat{z}}(k) = \overline{C_x}(k)C_y(k)$,

где $\overline{C_x}(k)$ – комплексное сопряжение $C_x(k)$

Доказательство

По определению имеем

$$C_{\hat{z}}(k) = \frac{1}{N} \sum_{m=0}^{N-1} \hat{Z}(m)W^{km}. \quad (3.4)$$

Подставляя (3.3) в (3.4) и меняя порядок суммирования, получаем

$$C_{\hat{z}}(k) = \frac{1}{N} \sum_{h=0}^{N-1} X(h) \left\{ \frac{1}{N} \sum_{m=0}^{N-1} Y(m+h)W^{km} \right\}$$

Применяя теорему сдвига, будем иметь

$$C_{\hat{z}}(k) = C_y(k) \left\{ \frac{1}{N} \sum_{h=0}^{N-1} X(h)W^{-kh} \right\}.$$

Так как $C_x(k) = \frac{1}{N} \sum_{h=0}^{N-1} X(h)W^{kh}$, то $\frac{1}{N} \sum_{h=0}^{N-1} X(h)W^{-kh} = \overline{C_x}(k)$.

Таким образом, $C_{\hat{z}}(k) = C_y(k)\overline{C_x}(k)$.

Если последовательности $\{X(m)\}$ и $\{Y(m)\}$ идентичны друг другу, то

$$C_{\hat{z}}(k) = |C_x(k)|^2, \quad k = \overline{0, N-1}.$$

Обратное ДПФ последовательности $C_{\hat{z}}(k)$ есть $\hat{Z}(m) = \sum_{k=0}^{N-1} C_{\hat{z}}(k)W^{-km}$.

Тогда

$$\frac{1}{N} \sum_{h=0}^{N-1} X^2(h) = \sum_{k=0}^{N-1} |C_x(k)|^2, \text{ т.е. справедлива теорема Парсеваля.}$$

21 Изображение. Получение изображения.

Визуальная информация преобразуется в электрические сигналы с помощью видеодатчиков. После пространственной дискретизации и квантования по амплитуде эти сигналы дают цифровое изображение.

Основными устройствами, используемыми в системах компьютерного видения, являются телевизионные камеры, состоящие из электронной трубки или твердотельного чувствительного элемента с соответствующими электронными блоками. Твердотельные чувствительные элементы представляются, как правило, приборами с зарядовой связью (ПЗС). Они обладают рядом преимуществ перед электронными трубками (например, имеют меньшие вес и размеры, больший ресурс и малую энергоемкость). Однако, разрешающая способность некоторых трубок пока выше возможности твердотельных камер.

Трубка видеокон представляет собой цилиндрическую стеклянную оболочку, содержащую с одного конца электронную пушку, а с другого - экран и мишень. Электронный луч фокусируется и отклоняется с помощью напряжения, прикладываемого к катушкам. Отклоняющий контур обеспечивает сканирование луча по внутренней поверхности мишени для считывания изображения. В процессе сканирования в металлическом слое образуется ток, который подается на выводы трубки. Величина тока пропорциональна числу перемещающихся электронов и следовательно интенсивности светового потока в соответствующем месте. Это изменение тока во время сканирования электронного луча после его обработки в электронном блоке камеры формирует видеосигнал, пропорциональный интенсивности входного изображения. Причем, электронный луч, сканирует по всей поверхности мишени 30 раз в 1 секунду. Полный объем сканирования (называемый кадром) состоит из 525 линий, 480 из которых содержат информацию об изображении.

При рассмотрении устройств на ПЗС удобно подразделить датчики на два типа: датчики линейного сканирования и датчики с плоскостной структурой. Основной частью ПЗС-датчиков является ряд кремниевых чувствительных элементов, называемых фотоячейками. Фотоны от отображаемого объекта проходят через входную прозрачную полукристаллическую кремниевую структуру и образуют пары электрон-дырка. Полученные фотоэлектроны собираются на фотоячейках, при этом величина заряда на каждой фотоячейке пропорциональна соответствующей интенсивности светового потока.

ПЗС-датчики с плоскостной структурой аналогичны датчикам линейного сканирования, с тем отличием, что в них фотоячейки расположены в форме матрицы, а между рядами фотоячеек имеется комбинация переходных регистров. Разрешающая способность датчиков с плоскостной структурой составляет 32×32 элементов на нижней границе и 256×256 элементов для среднего диапазона.

Современные ПЗС-датчики имеют 1024×1024 элементов и больше.

Обозначим через $f(x,y)$ двумерное изображение, получаемое телевизионной камерой или ПЗС-датчиком, дающим изображение:

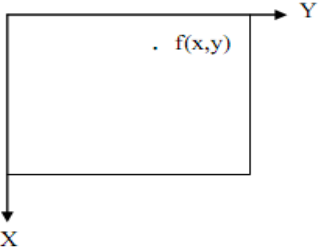


Рис. 13.1

Здесь x и y – пространственные координаты, т. е. координаты плоскости изображения, а величина f в произвольной точке (x,y) пропорциональна яркости (интенсивности) изображения в этой точке.

Для получения удобной с точки зрения вычисления формы, функции изображения $f(x,y)$, ее необходимо дискретизировать как пространственно, так и по амплитуде (интенсивности). Дискретизацию по пространственным координатам (x,y) будем называть дискретизацией изображения, а амплитудную дискретизацию – квантованием по интенсивности или квантованием по уровню серого. Последний термин означает тот факт, что изображение меняется по тону от черного до белого в серых оттенках.

Предположим, что непрерывное изображение дискретизировано равномерно на N строк и M столбцов, причем каждая дискретная величина проквантована по интенсивности. Такая система называется цифровым изображением и может быть представлена в виде:

$$f(x,y) = \begin{bmatrix} f(0,0) & f(0,1) & \dots & f(0,M-1) \\ f(1,0) & f(1,1) & \dots & f(1,M-1) \\ \dots & \dots & \dots & \dots \\ f(N-1,0) & f(N-1,1) & \dots & f(N-1,M-1) \end{bmatrix} \quad \begin{array}{l} \text{где } x \text{ и } y \text{ – дискретные переменные:} \\ x = 0,1,\dots, N-1; y = 0,1,\dots, M-1. \end{array}$$

Каждый элемент системы называется элементом изображения, элементом картинки или пикселем. В соответствии с рис.13.1 $f(0,0)$ является пикселем начала координат изображения, $f(0,1)$ – правый от него пиксел, и так далее. Обычно на практике величины N , M и число уровней дискретной интенсивности каждого квантованного пиксела задают в виде целых степеней числа 2. Степень дискретизации и число уровней интенсивности, необходимые для получения требуемого в системе изображения объекта зависят от самого изображения и от вида его использования.

Необходимо отметить, что для получения качественной черно-белой телевизионной картинки требуется $512*512$ пикселей со 128 уровнями интенсивности. Как правило, универсальная система компьютерного видения должна иметь, как минимум, разрешающую способность порядка $256*256$ пикселей с 64 уровнями интенсивности.

22 Система ортогональных функций Хаара.

Множество функций Хаара $har(n,m,t)$ (рис. 8.1), образующих периодическую, ортонормированную и полную систему функций, было предложено в 1910 году. Рекуррентное соотношение, позволяющее получить $har(n,m,t)$, имеет вид

$$har(0,0,t) = 1, \forall t \in [0,1);$$

$$har(r,m,t) = \begin{cases} 2^{r/2} & , \frac{m-1}{2^r} \leq t < \frac{m-1/2}{2^r}, \\ -2^{r/2} & , \frac{m-1/2}{2^r} \leq t < \frac{m}{2^r}, \\ 0 & , \text{при остальных } t \in [0,1), \end{cases}$$

где $0 \leq r < \log_2 N$, $1 \leq m \leq 2^r$

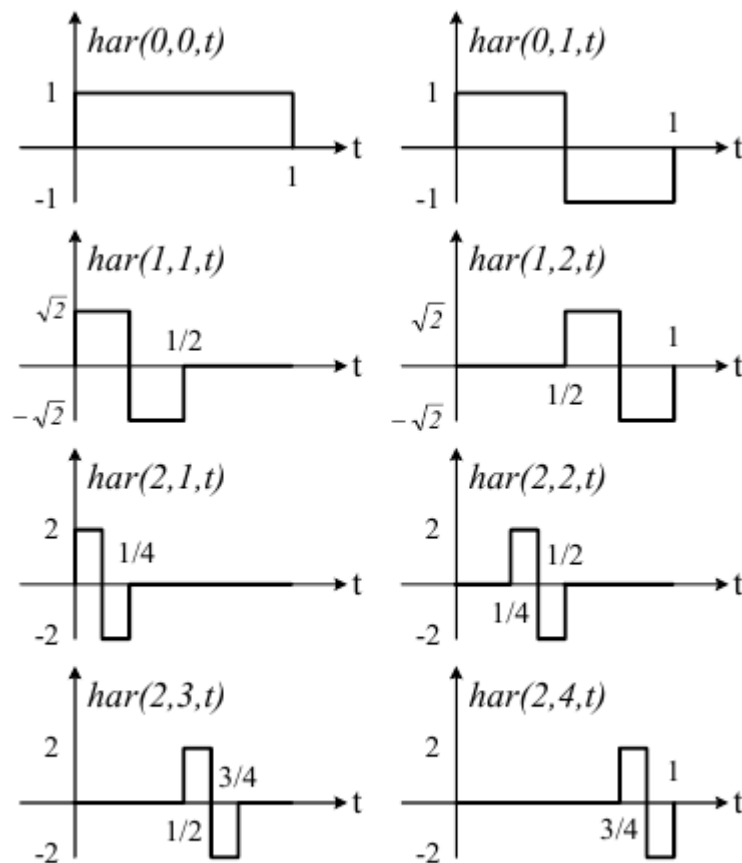


Рис. 8.1. Функции Хаара для $N=8$

Коэффициенты преобразования Хаара $Y(k)$, $k = \overline{0, N-1}$, соответствующие входной последовательности $\{X(m)\}$, получаются в результате преобразования

$$\vec{Y}(n) = \frac{1}{N} H^*(n) \vec{X}(n), \text{ где } H^*(n) \text{ – матрица Хаара размерностью } N \times N.$$

Для $N=8$

$$H^*(3) = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & -1 & -1 & -1 & -1 \\ \sqrt{2} & \sqrt{2} & -\sqrt{2} & -\sqrt{2} & & & & \\ & & & & \sqrt{2} & \sqrt{2} & -\sqrt{2} & -\sqrt{2} \\ 2 & -2 & & & & & & \\ & & 2 & -2 & & & & \\ & & & & 2 & -2 & & \\ & & & & & & 2 & -2 \end{bmatrix}.$$

Рассматривая $H^*(n)$, можно отметить, что $N/2$ коэффициентов преобразования Хаара соответствуют корреляции двух соседних точек в пространстве входных последовательностей, $N/4$ коэффициентов соответствуют связям четырех соседних точек и так далее до N/N коэффициентов, соответствующих всем N координатам пространства входных последовательностей. Это означает, что область преобразования в случае преобразования Хаара обладает свойством как локальной, так и глобальной чувствительности. При дискретном преобразовании Фурье и преобразовании Уолша-Адамара каждый коэффициент преобразования является функцией всех координат пространства входных последовательностей (свойство глобальной чувствительности), а в преобразовании Хаара это относится к первым двум коэффициентам.

Для осуществления преобразования Хаара требуется $2(N-1)$ операций сложения / вычитания и N операций умножения.

23 Алгоритм усеченного преобразования Адамара.

$$\begin{aligned}
 & P_{2^n}^{(k)} HAD_{2^n} P_{2^n}^l = \\
 & \left\{ \begin{aligned}
 & \frac{1}{2^{n/2}} \left(\prod_{i=0}^{n-1-k} (I_{2^i} \times S_2^0 \times P_{2^{n-1-i}}^{(k)}) \right) \times \\
 & \times \left(\prod_{j=n-k}^{l-1} (I_{2^j} \times h_2 \times I_{2^{n-1-j}}) \right) (I_{2^l} \times S_{2^{n-l}}^T) \\
 & k+l \geq n \\
 & \frac{1}{2^{n/2}} \left(\prod_{i=0}^{l-1} (I_{2^i} \times S_2^0 \times P_{2^{n-1-i}}^{(k)}) \right) \times \\
 & (I_{2^l} \times P_{2^{n-l-k}} \times S_{2^k}^T), k+l \leq n
 \end{aligned} \right.
 \end{aligned}$$

24 Принцип обучения в задаче двухклассового распознавания.

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_j , где Z_j обозначает j -ый образ, принадлежащий $C_i, i=1,2$

$$C_1: Z_{11} = \begin{bmatrix} 5 \\ 5 \end{bmatrix}; Z_{12} = \begin{bmatrix} 6 \\ 5 \end{bmatrix}; Z_{13} = \begin{bmatrix} 6 \\ 6 \end{bmatrix}; Z_{14} = \begin{bmatrix} 6 \\ 7 \end{bmatrix}; Z_{15} = \begin{bmatrix} 7 \\ 5 \end{bmatrix}$$

$$C_2: Z_{21} = \begin{bmatrix} 0 \\ 3 \end{bmatrix}; Z_{22} = \begin{bmatrix} -1 \\ 3 \end{bmatrix}; Z_{23} = \begin{bmatrix} -2 \\ 3 \end{bmatrix}; Z_{24} = \begin{bmatrix} -3 \\ 3 \end{bmatrix}; Z_{25} = \begin{bmatrix} -4 \\ 3 \end{bmatrix}$$

Образы $\{Z_j\}$, принадлежащие классам C_1 и C_2 , располагаются в двумерном пространстве признаков, как показано на рис. 23.2

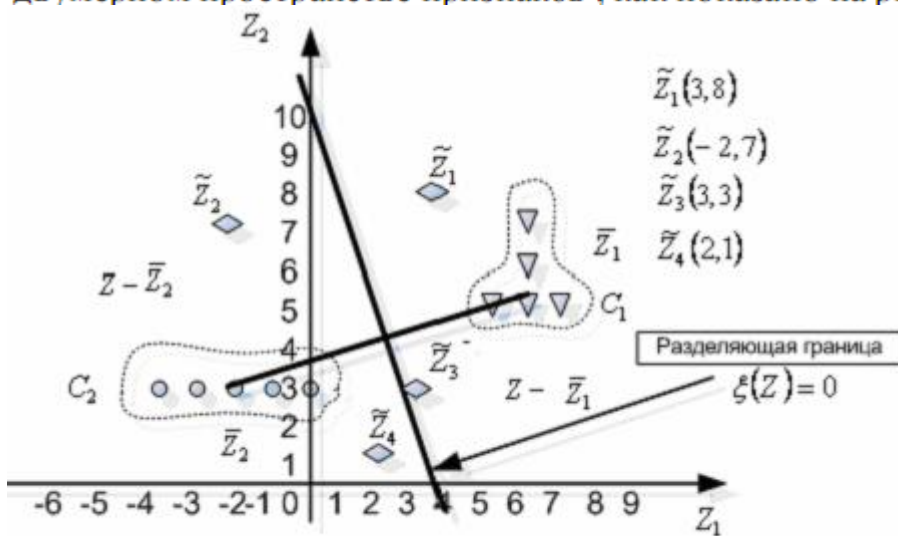


Рис. 23.2

Пусть \bar{Z}_1 и \bar{Z}_2 - средние векторы образов, связанные с C_1 и C_2

соответственно. Тогда
$$\bar{Z}_i = \frac{1}{5} \sum_{j=1}^5 Z_j, i=1,2$$

$$\bar{Z}_1 = \begin{bmatrix} 6 \\ 5,6 \end{bmatrix}, \bar{Z}_2 = \begin{bmatrix} -2 \\ 3 \end{bmatrix} \quad (23.1)$$

Из рис. 23.2 видно, что наиболее целесообразной решающей границей (т.е. линией на плоскости), разделяющей классы C_1 и C_2 , является срединный перпендикуляр к прямой, соединяющей \bar{Z}_1 и \bar{Z}_2 . Следующим шагом является описание решающей границы с помощью уравнения.

Рассмотрим любую точку Z , принадлежащую решающей границе, как показано на рис. Так как решающая граница является срединным перпендикуляром к прямой, соединяющей \bar{Z}_1 и \bar{Z}_2 , то

$$\|Z - \bar{Z}_1\|^2 = \|Z - \bar{Z}_2\|^2 \text{ или в упрощенном виде} \\ (\bar{Z}_1 - \bar{Z}_2)' Z = \frac{1}{2} \{ \|\bar{Z}_1\|^2 - \|\bar{Z}_2\|^2 \} \quad (23.2)$$

Подставляя (23.1) в (23.2), получим уравнение для решающей границы в виде

$$8Z_1 + 2,6Z_2 = 27,18 \quad (23.3)$$

Величина $0,5(\|\bar{Z}_1\|^2 - \|\bar{Z}_2\|^2) = 27,18$ называется порогом классификатора.

Уравнение (2) даёт всю информацию, необходимую для создания классификатора. Основной характеристикой, представляющей классификатор, является дискриминантная функция $\xi(Z)$, которая определяется как $\xi(Z) = 8Z_1 + 2,6Z_2 - 27,18$.

Возникает вопрос, что в действительности делает $\xi(Z)$? Чтобы ответить на этот вопрос, рассмотрим несколько испытательных образов (т.е. образов точная классификация которых неизвестна), которые обозначим как \tilde{Z}_k ($k = 1, 2, 3, 4$).

Вычислим теперь значение $\xi(Z)$ для каждого из испытательных образов

$$\tilde{Z}_1 : \xi(Z) = 8(3) + 2,6(8) - 27,18 > 0$$

$$\tilde{Z}_2 : \xi(Z) = 8(-2) + 2,6(7) - 27,18 < 0$$

$$\tilde{Z}_3 : \xi(Z) = 8(3) + 2,6(3) - 27,18 > 0$$

$$\tilde{Z}_4 : \xi(Z) = 8(2) + 2,6(1) - 27,18 < 0$$

Рассмотрение последнего выражения приводит к выводу, что когда образ \tilde{Z} лежит справа от решающей границы, то $\xi(Z) > 0$ и наоборот, когда образ \tilde{Z} лежит слева от решающей границы, то $\xi(Z) < 0$. Все точки лежащие справа от границы ближе к \tilde{Z}_1 , а точки, лежащие слева от границы ближе к \tilde{Z}_2 .

Таким образом, благодаря дискриминантной функции $\xi(Z)$ получим следующее простое решающее правило:

если $\xi(Z) > 0$, то $Z \in C_1$ если $\xi(Z) < 0$, то $Z \in C_2$

Описанный выше классификатор, называемый элементом линейной пороговой логики, реализуется как показано на рис. 23.3.

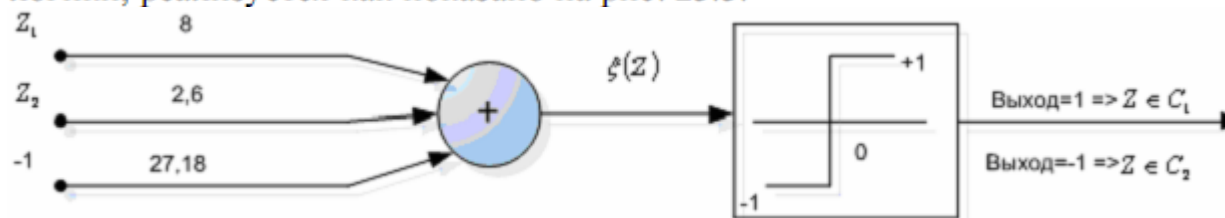


Рис. 23.3

Получив дискриминантную функцию $\xi(Z)$, говорят, что классификатор обучен, т.е. способен классифицировать образы с помощью соответствующего решающего правила. Элемент пороговой логики является

классификатором, работающим по критерию минимума расстояния, так как решающее правило может быть сформулировано также следующим образом:

если Z ближе к \bar{Z}_1 , то $Z \in C_1$ и если Z ближе к \bar{Z}_2 , то $Z \in C_2$

25 Структура системы распознавания образов с применением ортогональных преобразований.

Систему распознавания образов с применением ортогональных преобразований можно представить в виде

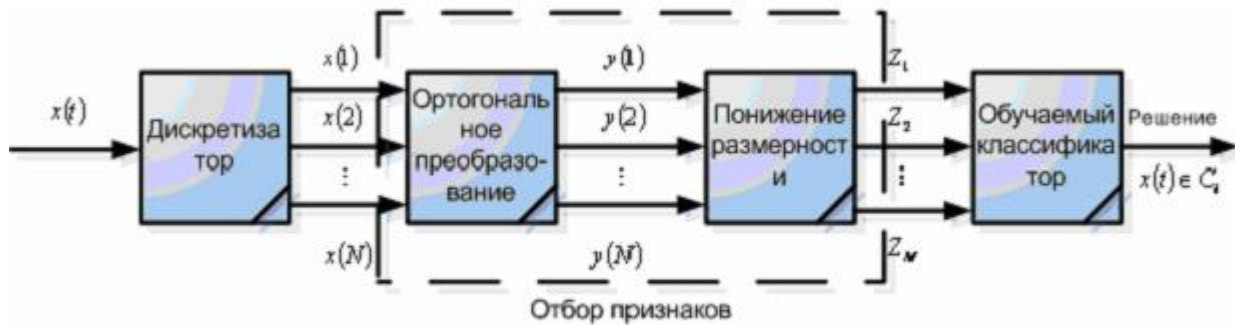


Рис. 23.1

Через $x(t)$ обозначается сигнал, принадлежащий одному из K классов C_1, C_2, \dots, C_K .

На первом этапе выбора осуществляется ортогональное преобразование. Вторым этапом выбора признака является понижение размерности, после чего получаем подмножество M признаков z_1, z_2, \dots, z_M из $\{Y(M)\} = \{Y(1), Y(2), \dots, Y(N)\}$ причём $M \ll N$.

Понижать размерность следует таким образом, чтобы сопутствующее этому увеличению ошибки классификации было относительно невелико.

Классификатор, изображенный на рис. 23.1 является решающим устройством, которое обучается с целью классификации входного сигнала $x(t)$, принадлежащего к одному из K классов.

Рассмотрим элемент линейной пороговой логики, на который воздействуют $(d \times 1)$ образы, где $d > 2$. В этом случае $Z' = [Z_1 Z_2 \dots Z_d]$. Соответствующая дискриминантная функция записывается как

$$\xi(Z) = w_1 Z_1 + w_2 Z_2 + \dots + w_d Z_d - \theta$$

где w_i - веса или параметры классификатора, а θ - порог. Веса w_i и θ получаются из обучающего множества. Границей в этом случае является гиперплоскость, определяемая как

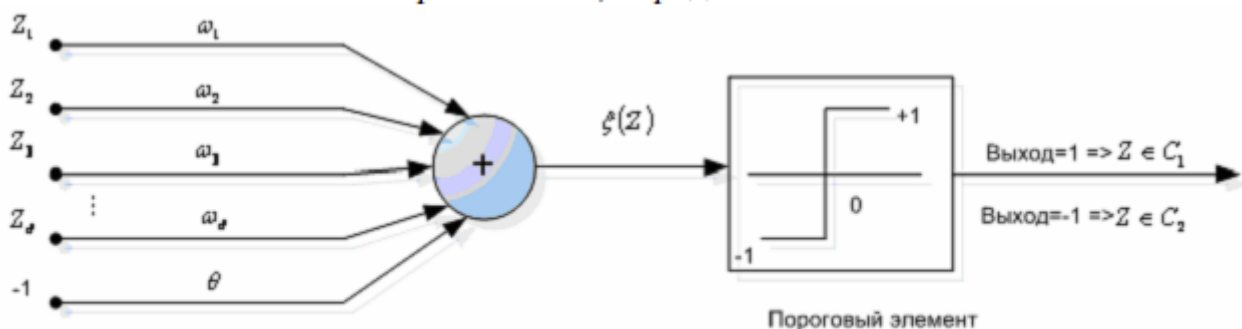


Рис. 23.4

26 Классификатор по минимальному расстоянию

В.Классификатор по минимальному расстоянию.

Важный класс составляют линейные классификаторы, в которых в качестве критерия классификации используется расстояние между входным образом и множеством опорных векторов или эталонных точек в пространстве признаков. Предположим, что задано m опорных векторов R_1, R_2, \dots, R_m , где R_j соответствует классу образов ω_j . При классификации по минимальному расстоянию относительно R_1, R_2, \dots, R_m входной сигнал X предполагается принадлежащим ω_i , т.е.

$X \sim \omega_i$, если $|X - R_i|$ минимально, где $|X - R_i|$ есть расстояние между X и R_i . Расстояние можно определить, например, следующим образом

$$|X - R_i| = \left[(X - R_i)^T (X - R_i) \right]^{1/2}, \quad (22.2)$$

где индекс T определяет операцию транспонирования вектора.

Из последнего соотношения следует, что $|X - R_i|^2 = X^T X - X^T R_i - X R_i^T + R_i^T R_i$

Так как $X^T X$ не зависит от i , то соответствующая разделяющая

функция для классификатора по минимальному расстоянию имеет вид

$$D_i(x) = X^T R_i + X R_i^T - R_i^T R_i, \quad i = \overline{1, m}$$

Как видно из этого соотношения, классификатор по минимальному расстоянию является линейной функцией. В свою очередь, свойства классификатора по минимальному расстоянию конечно зависят от того, как выбраны опорные векторы.

27 Виды пороговых функций в нейросети.

Порог, если он используется, является характеристикой, задающей начальный уровень активности (при нулевом входе) и помогающей настроить нейрон на пороговую функцию. Изменение порога эквивалентно сдвигу пороговой функции по оси абсцисс. Ряд авторов вводят дополнительный вход нейрона x_0 , всегда равный 1, и обозначают порог как его вес w_0 . Это позволяет упростить выражение (29.1) и математическую запись некоторых алгоритмов обучения. Однако на практике при программной реализации это не приводит к экономии времени и способствует ошибкам, кроме этого, порог может настраиваться иначе, чем весовой вектор. Функция активации используется для ограничения выхода нейрона в заданном диапазоне и для нелинейного преобразования взвешенной суммы. Последнее позволяет нейронному классификатору аппроксимировать любую нелинейную границу между классами в пространстве образов. Функция активации выбирается для конкретной задачи и является неизменной характеристикой отдельного нейрона.

Могут использоваться следующие функции активации и их гибриды:

1) линейная функция $y = Ax$;

2) пороговая функция $y = \begin{cases} 1, x > 0 \\ 0, x \leq 0 \end{cases}$;

3) биполярная пороговая функция $y = \begin{cases} 1, x > 0 \\ -1, x \leq 0 \end{cases}$;

4) сигмоидная функция (рис. 29.2)

$$y = \frac{1}{(1 + e^{-x})}; \quad (29.2)$$

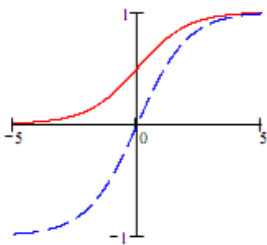


Рис. 7.2. Сигмоидные функции активации

5) биполярная сигмоидная функция (см. рис. 29.2)

$$y = \frac{2}{(1 + e^{-x})} - 1; \quad (29.3)$$

6) гиперболический тангенс (рис. 29.3)

$$y = th(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}}. \quad (29.4)$$

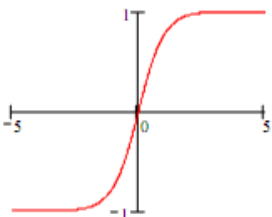


Рис. 7.3. Гиперболический тангенс

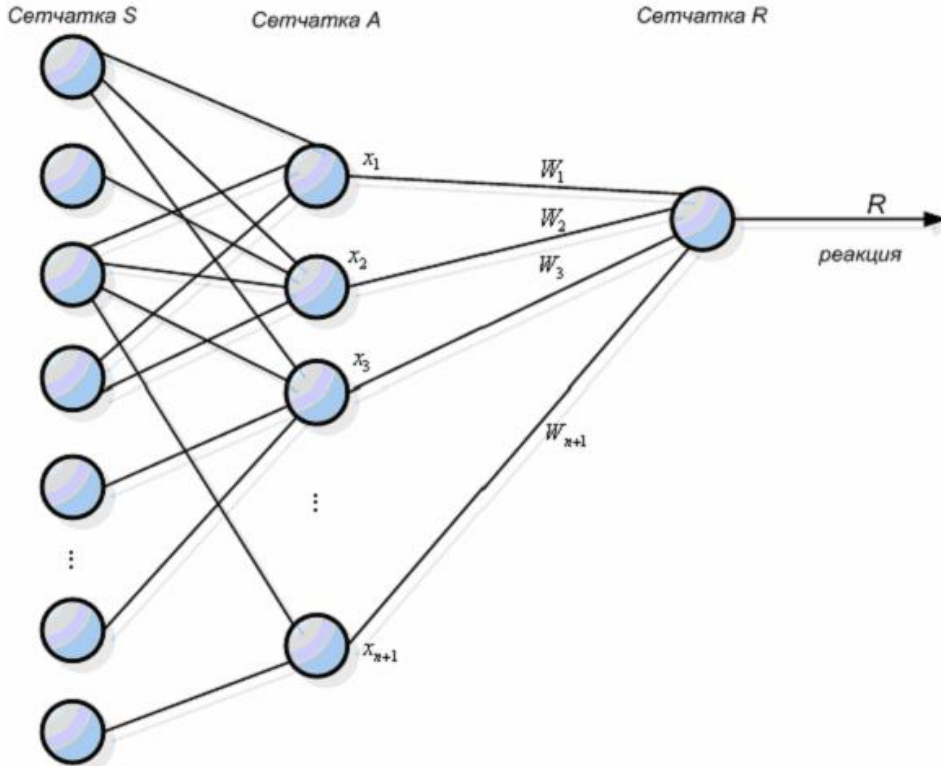
Для классификаторов чаще всего применяют функции (29.2) – (29.4), поскольку они нелинейные и хорошо дифференцируются. Веса и порог конкретного нейрона являются настраиваемыми параметрами. Они содержат знания нейрона, определяющие его по-

ведение. Процесс настройки этих знаний с целью получения нужного поведения называется обучением.

Нейронная сеть – совокупность нейронных элементов и связей между ними. Слоем нейронной сети называется множество нейронных элементов, на которые в каждый такт времени параллельно поступает информация от других нейронных элементов сети. Помимо слоев нейронов часто используется понятие входного распределительного слоя. Распределительный слой передает входные сигналы на первый обрабатывающий слой нейронных элементов (рис. 29.4)..

28 Модель перцептрона

Основная модель перцептрона, обеспечивающая отнесение образа к одному из заданных классов приведена на рис. 28.2.



Устройство состоит из сетчатки S сенсорных элементов, которые случайным образом соединены с ассоциативными элементами второй сетчатки A. Каждый из элементов второй сетчатки воспроизводит выходной сигнал только в том случае, если достаточное число сенсорных элементов, соединённых с его входом, находится в возбуждённом состоянии. Сенсорные элементы можно рассматривать в качестве устройств, с помощью которых вся система воспринимает из внешней среды стимулы, т.е. как некие измерительные устройства, а ассоциативные элементы – как входную часть системы. Реакция всей системы пропорциональна сумме взятых с определёнными весами реакций элементов ассоциативной сетчатки. Таким образом, обозначив через x_i реакцию i -го ассоциативного элемента и через w_i – соответствующий вес

$$R = \sum_{i=1}^{n+1} W_i x_i = W'X$$

реакцию системы можно записать как

Если $R > 0$, значит предъявленный системе образ принадлежит классу ω_1 , если $R < 0$, то образ относится к классу ω_2 .

Отметим, что перцептронная модель представляет собой за исключением сенсорной сетчатки не что иное, как реализацию линейной решающей функции. Схему, приведённую на рис 28.2, легко распространить на случай разделения на несколько классов посредством увеличения числа реагирующих элементов в R-сетчатке.

Классификация проводится обычным способом: рассматриваются значения реакций R_1, R_2, \dots, R_M , где M – число классов и образ причисляется к классу ω_i , если $R_i > R_j$ для всех $i \neq j$.

Основную модель можно также легко распространить на случай нелинейных решающих функций введением соответствующих нелинейных преобразователей между сетчатками A и R.

29 Алгоритм быстрого преобразования Фурье.

Быстрое преобразование Фурье (FFT) является не более, чем алгоритмом для ускоренного вычисления ДПФ путем сокращения требуемого числа операций умножения и сложения. Данное преобразование было предложено в 1960-ых годах. Алгоритм быстрого преобразования Фурье значительно сокращает количество арифметических операций и объем памяти, необходимой для вычисления ДПФ. ДПФ может быть сильно упрощено, если использовать свойства симметрии и периодичности коэффициентов поворота.

При вычислении N -точечного ДПФ требуется N^2 вычислений с комплексными числами, а при реализации N -точечного БПФ $(N/2)\log_2(N)$ вычислений с комплексными числами. Вычислительная эффективность БПФ по сравнению с ДПФ становится весьма существенной, когда количество точек БПФ увеличивается до нескольких тысяч

Если необходимо рассчитать только несколько точек спектра, ДПФ может быть более эффективным. Вычисление одного выходного отсчета спектра с использованием ДПФ требует только N умножений с комплексными числами.

Мы будем предполагать далее, что $N=2^n$. При этом общность не теряется, так как N выбирается достаточно большим для того, чтобы удовлетворять теореме дискретизации Котельникова, т.е. $N \geq 2BT$, где B – полоса частот сигнала $x(t)$; T – его длительность.

Теорема Котельникова-Найквиста-Шеннона: если сигнал таков, что его спектр ограничен частотой F , то после дискретизации сигнала с частотой не менее $2F$ можно восстановить непрерывный сигнал по полученному цифровому сигналу абсолютно точно. Для этого нужно проинтерполировать цифровой сигнал «между отсчетами» специального вида функциями. Рассмотрим случай вещественно-значной последовательности $\{X(m)\}$ при $N=8$. Из свойства комплексной сопряженности ДПФ следует, что

$$C_x(4+l) = C_x^*(4-l); \quad l = \overline{1, N/2-1}.$$

$$\text{Тогда } C_x(k) = \frac{1}{8} \sum_{m=0}^7 X(m) W^{km}; \quad k = \overline{0, N-1}; \quad W = e^{-i2\pi/8} = e^{-i\pi/4};$$

$$8C_x(k) = \sum_{m=0}^7 X(m) \cos\left(\frac{mk\pi}{4}\right) - i \sum_{m=0}^7 X(m) \sin\left(\frac{mk\pi}{4}\right). \quad (4.1)$$

Используя свойство N -периодичности экспонент, для $N=8$ матрица будет иметь вид

$$F = \begin{bmatrix} W_0 & W_0 & W_0 & W_0 & W_0 & W_0 & W_0 & W_0 \\ W_0 & W_1 & W_2 & W_3 & W_4 & W_5 & W_6 & W_7 \\ W_0 & W_2 & W_4 & W_6 & W_0 & W_2 & W_4 & W_6 \\ W_0 & W_3 & W_6 & W_1 & W_4 & W_7 & W_2 & W_5 \\ W_0 & W_4 & W_0 & W_4 & W_0 & W_4 & W_0 & W_4 \\ W_0 & W_5 & W_2 & W_7 & W_4 & W_1 & W_6 & W_3 \\ W_0 & W_6 & W_4 & W_2 & W_0 & W_6 & W_4 & W_2 \\ W_0 & W_7 & W_6 & W_5 & W_4 & W_3 & W_2 & W_1 \end{bmatrix}.$$

Из свойства симметрии экспоненциальных функций следует, что

$$W_{k+N/2} = -W_k, \quad \text{где } k = \overline{0, N/2-1}.$$

$$\text{То есть } W_4 = -W_0; \quad W_5 = -W_1; \quad W_6 = -W_2; \quad W_7 = -W_3.$$

Тогда матрица F будет иметь вид

$$F = \begin{bmatrix} W_0 & W_0 & W_0 & W_0 & W_0 & W_0 & W_0 & W_0 \\ W_0 & W_1 & W_2 & W_3 & -W_0 & -W_1 & -W_2 & -W_3 \\ W_0 & W_2 & -W_0 & -W_2 & W_0 & W_2 & -W_0 & -W_2 \\ W_0 & W_3 & -W_2 & W_1 & -W_0 & -W_3 & W_2 & -W_1 \\ W_0 & -W_0 & W_0 & -W_0 & W_0 & -W_0 & W_0 & -W_0 \\ W_0 & -W_1 & W_2 & -W_3 & -W_0 & W_1 & -W_2 & W_3 \\ W_0 & -W_2 & -W_0 & W_2 & W_0 & -W_2 & -W_0 & W_2 \\ W_0 & -W_3 & -W_2 & -W_1 & -W_0 & W_3 & W_2 & W_1 \end{bmatrix} \begin{matrix} 0 \\ 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \\ 7 \end{matrix}$$

Используя двоичную инверсию (перестановку) строк,

$(0,1,2,3,4,5,6,7) \rightarrow (0,4,2,6,1,5,3,7)$ будем иметь

$$F = \begin{bmatrix} W_0 & W_0 & W_0 & W_0 & W_0 & W_0 & W_0 & W_0 \\ W_0 & -W_0 & W_0 & -W_0 & W_0 & -W_0 & W_0 & -W_0 \\ W_0 & W_2 & -W_0 & -W_2 & W_0 & W_2 & -W_0 & -W_2 \\ W_0 & -W_2 & -W_0 & W_2 & W_0 & -W_2 & -W_0 & W_2 \\ W_0 & W_1 & W_2 & W_3 & -W_0 & -W_1 & -W_2 & -W_3 \\ W_0 & -W_1 & W_2 & -W_3 & -W_0 & W_1 & -W_2 & W_3 \\ W_0 & W_3 & -W_2 & W_1 & -W_0 & -W_3 & W_2 & -W_1 \\ W_0 & -W_3 & -W_2 & -W_1 & -W_0 & W_3 & W_2 & W_1 \end{bmatrix} \begin{bmatrix} X_0 \\ X_1 \\ X_2 \\ X_3 \\ X_4 \\ X_5 \\ X_6 \\ X_7 \end{bmatrix} = \begin{bmatrix} A_1^1 & A_1^1 \\ B_1^1 & -B_1^1 \end{bmatrix} \bar{X} = \begin{matrix} A_1^1 \begin{Bmatrix} X_0 + X_4 \\ X_1 + X_5 \\ X_2 + X_6 \\ X_3 + X_7 \end{Bmatrix} \\ B_1^1 \begin{Bmatrix} X_0 - X_4 \\ X_1 - X_5 \\ X_2 - X_6 \\ X_3 - X_7 \end{Bmatrix} \end{matrix} \quad (4.2)$$

В свою очередь, матрицы A_1^1 и B_1^1 можно представить в виде, где верхний индекс представляет собой номер шага процедуры БПФ

$$A_1^1 = \begin{bmatrix} A_1^2 & A_1^2 \\ A_2^2 & -A_2^2 \end{bmatrix}; \quad (4.3)$$

$$B_1^1 = \begin{bmatrix} B_1^2 & W_2 B_1^2 \\ B_2^2 & -W_2 B_2^2 \end{bmatrix}.$$

Подставляя выражения для A_1^1 и B_1^1 в (4.2) получим

$$A_1^2 \begin{Bmatrix} (X_0 + X_4) + (X_2 + X_6) \\ (X_1 + X_5) + (X_3 + X_7) \end{Bmatrix}; \quad A_2^2 \begin{Bmatrix} (X_0 + X_4) - (X_2 + X_6) \\ (X_1 + X_5) - (X_3 + X_7) \end{Bmatrix}; \quad B_1^2 \begin{Bmatrix} (X_0 - X_4) + W_2(X_2 - X_6) \\ (X_1 - X_5) + W_2(X_3 - X_7) \end{Bmatrix};$$

$$B_2^2 \begin{Bmatrix} (X_0 - X_4) - W_2(X_2 - X_6) \\ (X_1 - X_5) - W_2(X_3 - X_7) \end{Bmatrix}.$$

Наконец, на последнем шаге получим

$$\begin{aligned} & [(X_0 + X_4) + (X_2 + X_6)] + [(X_1 + X_5) + (X_3 + X_7)] \\ & [(X_0 + X_4) + (X_2 + X_6)] - [(X_1 + X_5) + (X_3 + X_7)] \\ & [(X_0 + X_4) - (X_2 + X_6)] + W_2[(X_1 + X_5) - (X_3 + X_7)] \\ & [(X_0 + X_4) - (X_2 + X_6)] - W_2[(X_1 + X_5) - (X_3 + X_7)] \\ & [(X_0 - X_4) + W_2(X_2 - X_6)] + W_1[(X_1 - X_5) + W_2(X_3 - X_7)] \\ & [(X_0 - X_4) + W_2(X_2 - X_6)] - W_1[(X_1 - X_5) + W_2(X_3 - X_7)] \\ & [(X_0 - X_4) - W_2(X_2 - X_6)] + W_3[(X_1 - X_5) - W_2(X_3 - X_7)] \\ & [(X_0 - X_4) - W_2(X_2 - X_6)] - W_3[(X_1 - X_5) - W_2(X_3 - X_7)] \end{aligned} \quad (4.5)$$

Описанный алгоритм удобно представить графически (рис. 4.1).

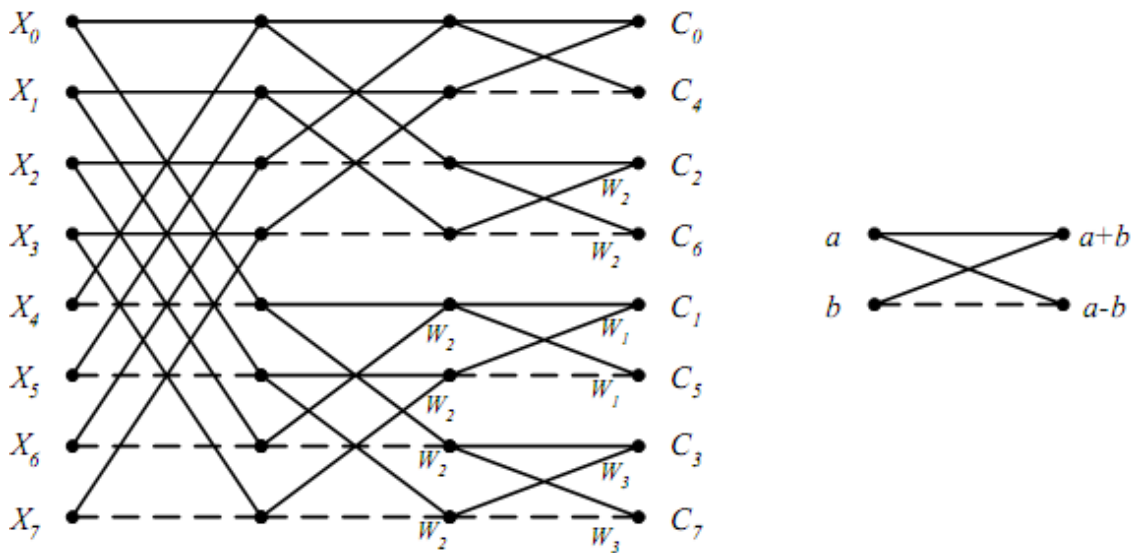


Рис. 4.1. Граф-схема быстрой процедуры вычисления коэффициентов преобразования Фурье

Для определения степеней W на каждом шаге необходимо выразить последовательность $l=0, 1, 2, \dots, N/2-1$ в виде $(n-1)$ – разрядных двоичных последовательностей. В результате для $N=16$, к примеру, получим множество $S_1=(000,001,010,011,100,101,110,111)$.

Для получения S_2 необходимо выполнить двоичную инверсию каждой $(n-1)$ -разрядной последовательности множества S_1 , т.е.

$$S_2=(000,100,010,110,001,101,011,111),$$

и записать двоичную последовательность в виде десятичных чисел

$$S_3=(0,4,2,6,1,5,3,7), \text{ и таким образом имеем } W_0, W_4, W_2, W_6, W_1, W_5, W_3, W_7. \text{ (табл. 4.2).}$$

Итерация r для БПФ состоит из 2^{r-1} групп, где $r = \overline{1, n}$ ($N=2^n$). Для $N=16$, $r = \overline{1, 4}$.

Таблица 4.2 Значения степени W

Номер итерации	Степени W ($N=16$)
1	$W_0 W_0 W_0 W_0 \quad W_0 W_0 W_0 W_0 \quad W_0 W_0 W_0 W_0 \quad W_0 W_0 W_0 W_0$
2	$W_0 W_0 W_0 W_0 \quad W_0 W_0 W_0 W_0 \quad W_4 W_4 W_4 W_4 \quad W_4 W_4 W_4 W_4$
3	$W_0 W_0 W_0 W_0 \quad W_4 W_4 W_4 W_4 \quad W_2 W_2 W_2 W_2 \quad W_6 W_6 W_6 W_6$
4	$W_0 W_0 W_4 W_4 \quad W_2 W_2 W_6 W_6 \quad W_1 W_1 W_5 W_5 \quad W_3 W_3 W_7 W_7$

Процедура, обеспечивающая инверсию, может быть описана в матричном виде следующим образом.

Первый элемент первой строки таблицы равен нулю. Последующие первые элементы каждой из строк определяются как $n_s = N/2^s$, где $s = \overline{1, n}$,

$N=2^n$. Каждая k строка таблицы получается прибавлением элемента n_{k-1} к каждому элементу предыдущих строк. Тогда таблица будет иметь вид

$$\begin{array}{cccc}
 \bullet & & & \\
 n_1 & & & \\
 n_2 & (n_1+n_2) & & \\
 n_3 & (n_1+n_3) & (n_2+n_3) & (n_1+n_2+n_3) \\
 n_4 & (n_1+n_4) & (n_2+n_4) & (n_1+n_2+n_4) \dots \\
 \bullet & & &
 \end{array}$$

$n_k \dots$

Требуемая последовательность L_n , соответствующая двоичной инверсии, определяется как $L_n=(0, n_1, n_2, (n_1+n_2), n_3, (n_1+n_3), \dots, n_k, \dots)$. В качестве примера рассмотрим случай для $N=16$. Тогда $n_1=8, n_2=4, n_3=2, n_4=1$, т.е. таблица будет иметь вид

0
8
4 12
2 10 6 14
1 9 5 13 3 11 7 15

$L_n=(0, 8, 4, 12, 2, 10, 6, 14, 1, 9, 5, 13, 3, 11, 7, 15)$.

Для обработки исходных данных (которые предполагаются комплексными) с помощью алгоритма БПФ требуется $2N$ ячеек оперативной памяти. Поэтому выходной массив может храниться в тех же ячейках памяти, что и исходный массив. Процедура перестановки данных может потребовать дополнительно $2N$ ячеек памяти. Таким образом, для алгоритма БПФ необходимо примерно $4N$ ячеек. В противоположность этому прямой метод требует приблизительно $2N^2$ ячеек памяти, т.к. необходимо запомнить N^2 значений степеней W .

В общем виде матрицу преобразования Фурье в факторизованной форме можно представить как

$$F_N = \prod_{i=1}^n F_i D_i. \quad (4.6)$$

Для $N=8$ $F_8 = F_1 D_1 \cdot F_2 D_2 \cdot F_3 D_3$, где $F_1 = \begin{bmatrix} I_4 & I_4 \\ I_4 & -I_4 \end{bmatrix}$; I_4 – единичная матрица размерностью 4×4 ; D_1 – диагональная матрица с элементами W_0 ;

$$F_2 = \begin{bmatrix} I_2 & I_2 \\ I_2 & -I_2 \\ & I_2 & I_2 \\ & I_2 & -I_2 \end{bmatrix}.$$

D_2 – диагональная матрица с элементами W_0, W_2 ;

$$D_2 = \begin{bmatrix} W_0 & & & & & & & \\ & W_0 & & & & & & \\ & & W_0 & & & & & \\ & & & W_0 & & & & \\ & & & & W_0 & & & \\ & & & & & W_2 & & \\ & & & & & & W_2 & \\ & & & & & & & W_2 \end{bmatrix}; F_3 = \begin{bmatrix} I_1 & I_1 & & & & & & \\ I_1 & -I_1 & & & & & & \\ & & I_1 & I_1 & & & & \\ & & I_1 & -I_1 & & & & \\ & & & & I_1 & I_1 & & \\ & & & & I_1 & -I_1 & & \\ & & & & & & I_1 & I_1 \\ & & & & & & I_1 & -I_1 \end{bmatrix};$$

$$D_3 = \begin{bmatrix} W_0 & & & & & & & \\ & W_0 & & & & & & \\ & & W_0 & & & & & \\ & & & W_2 & & & & \\ & & & & W_0 & & & \\ & & & & & W_1 & & \\ & & & & & & W_0 & \\ & & & & & & & W_3 \end{bmatrix}.$$

Факторизованная форма – это такая форма, когда в каждой строке матрицы, являющейся множителем, присутствует не более 2 значащих элементов, а остальные равны нулю. Алгоритм быстрого преобразования Фурье можно выразить следующим образом. Общая схема алгоритма состоит в повторяющемся сведении ДПФ вектора длины N к векторам длины $N/2$ и объединении результатов. Базисом рекурсии служат векторы длины 1, для которых ДПФ – сам вектор.

30 Основные преобразования изображения

Основными преобразованиями при обработке изображений являются вращение, изменение масштаба и смещение (сдвиг) изображения. Все преобразования записываются в трехмерной декартовой системе координат, в которой координаты точки записываются как (X, Y, Z) . В случае двумерных изображений для обозначения координат пиксела используется сокращенная запись (x, y) .

Смещение (сдвиг) изображения

Предположим, что требуется сместить точку с координатами (X, Y, Z) в новое место, используя перемещения (X_0, Y_0, Z_0) . Смещение выполняется в соответствии с соотношениями:

$$X^* = X + X_0, Y^* = Y + Y_0, Z^* = Z + Z_0,$$

где X^*, Y^*, Z^* - координаты новой точки.

В матричном виде это можно записать как $V^* = TV$, то есть:

$$\begin{bmatrix} X^* \\ Y^* \\ Z^* \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & X_0 \\ 0 & 1 & 0 & Y_0 \\ 0 & 0 & 1 & Z_0 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & X_0 \\ 0 & 1 & 0 & Y_0 \\ 0 & 0 & 1 & Z_0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}$$

Изменение масштаба изображения

Масштабирование на коэффициенты S_x, S_y, S_z по осям X, Y, Z производится с помощью матрицы преобразования:

$$S = \begin{bmatrix} S_x & 0 & 0 & 0 \\ 0 & S_y & 0 & 0 \\ 0 & 0 & S_z & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Вращение изображения

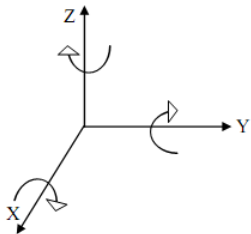


Рис. 13.2

$$R_\theta = \begin{bmatrix} \cos \theta & \sin \theta & 0 & 0 \\ -\sin \theta & \cos \theta & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Вращение точки относительно координатной оси Z на угол θ реализуется с помощью преобразования

$$R_\alpha = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos \alpha & \sin \alpha & 0 \\ 0 & -\sin \alpha & \cos \alpha & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Вращение точки вокруг оси X на угол α выполняется с помощью преобразования

$$R_\beta = \begin{bmatrix} \cos \beta & 0 & -\sin \beta & 0 \\ 0 & 1 & 0 & 0 \\ \sin \beta & 0 & \cos \beta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Вращение точки вокруг оси Y на угол β реализуется с помощью преобразования

31 Основные этапы распознавания

Задача распознавания образов заключается в классификации некоторой группы объектов на основе определённых требований. Требования эти могут быть различными, так как в различных ситуациях возникает необходимость в определённом типе классификатора. К примеру при распознавании английских букв образуется 26 классов образов. Однако, чтобы отличить при распознавании буквы от китайских иероглифов нужны лишь два класса образов. Проблема распознавания получила широкое распространение при решении различных задач в различных областях народного хозяйства: распознавание рукописных букв или слов, медицинская диагностика, диагностика неисправностей в технических системах, обнаружение цели, прогноз погоды, классификация сейсмических волн. Простейший подход к распознаванию образов заключается в сопоставлении с эталоном. В этом случае некоторое множество образов по одному из каждого класса образов хранится в памяти машины. Входной, подлежащий распознаванию образ (неизвестного класса) сравнивается с эталоном каждого класса. Классификация основывается на заранее выбранном критерии соответствия или критерии подобия. Иными словами, если входной образ лучше соответствует эталону i -го класса образов, чем любому другому эталону, то входной образ классифицируется как принадлежащий i -му классу образов.

Более совершенный подход заключается в том, что вместо сравнения входного образа с эталонами, классификация основывается на некотором множестве отобранных замеров, называемых информативными признаками, которые предполагаются инвариантными или малочувствительными по отношению к изменениям и искажениям. В этом случае распознавание образов можно рассматривать состоящим из двух задач.

Первая задача заключается в выборе подмножества признаков и их упорядочиванию в заданном множестве измерений. Вторая задача заключается в классификации (принятии решения о принадлежности входного образа тому или иному классу), которая основывается на измерениях отобранных признаков. Упрощенная блок-схема системы распознавания приведена на рис. 1.

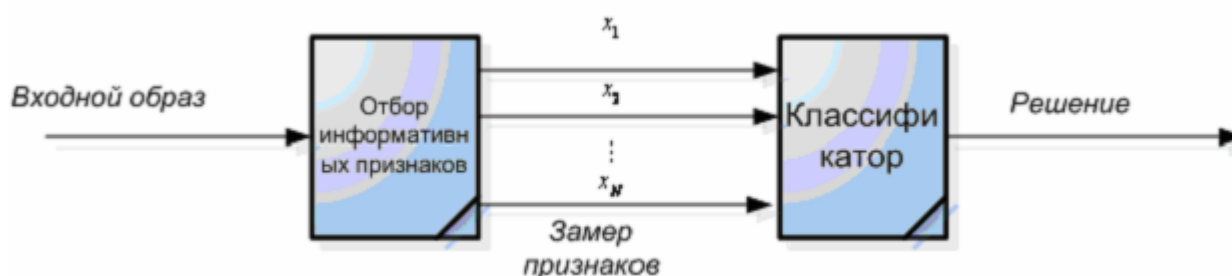


Рис. 21.1

Таким образом, подход, заключающийся в сопоставлении с эталоном, можно рассматривать как частный случай второго подхода - подхода измерения признаков, при этом эталоны хранятся в виде измеренных признаков, а в классификаторе используется специальный критерий сопоставления. Математически задача классификации может быть сформулирована с помощью разделяющей функции.

Пусть $\omega_1, \omega_2, \dots, \omega_m$ обозначают m возможных классов образов, подлежащих распознаванию и пусть $X' = [x_1, x_2, \dots, x_n]$ вектор замеров

признаков, где x_k предоставляет k -ый замер.

Тогда разделяющая функция $D_j(x)$, относящаяся к классу образов $\omega_j, j = \overline{1, m}$, такова, что если входной образ представленный вектором признаков X , принадлежит классу ω_i , то величина $D_i(x)$ должна быть наибольшей.

Пусть $X \sim \omega_i$ обозначает, что вектор признаков X входного образа принадлежит классу ω_i . Тогда можно записать, что для всех $X \sim \omega_i$

$$D_i(x) > D_j(x); i, j = \overline{1, m} \quad i \neq j \quad (21.1)$$

Таким образом, в пространстве признаков Ω_x граница разбиений, называемая решающей границей между областями, относящимися

соответственно к классам ω_i, ω_j , выражается уравнением $D_i(x) - D_j(x) = 0$

Общая схема классификатора, использующего критерий (21.1) и типичный двумерный пример приведены на рис. 21.2 и рис. 21.3.

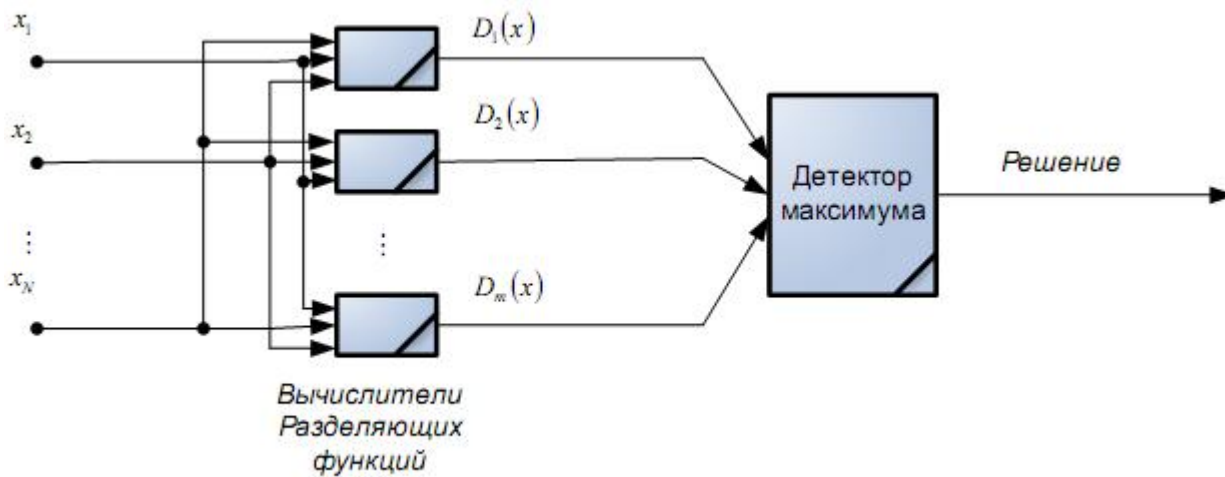


Рис. 21.2

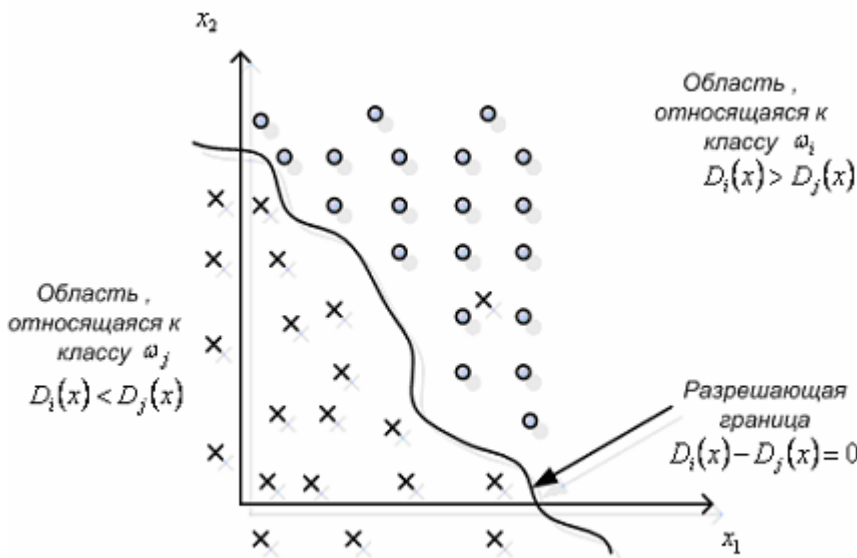


Рис. 21.3

32 Основные взаимосвязи между пикселями изображений.

Пиксел p с координатами (x, y) имеет четыре горизонтальных и вертикальных соседних пиксела с координатами: $(x+1, y)$, $(x-1, y)$, $(x, y+1)$, $(x, y-1)$

Эта группа пикселов, называемая четыре соседа p , обозначается через $N_4(p)$. Четыре диагональных соседних пиксела по отношению к пикселу p имеют координаты

$(x+1, y+1)$, $(x+1, y-1)$, $(x-1, y+1)$, $(x-1, y-1)$ и обозначаются через $N_d(p)$. Эти четыре точки вместе с четырьмя указанными выше называются восемь соседей p и обозначаются через $N_8(p)$. Некоторые из точек $N_4(p)$, $N_d(p)$, $N_8(p)$ могут выходить за пределы изображения, если (x, y) находится на границе изображения.

Рассмотрим три типа связей пикселов:

1. **Четырехсвязный.** Два пиксела p и q с определенными значениями интенсивности являются четырехсвязными, если q относится к группе $N_4(p)$.
2. **Восьмисвязный.** Два пиксела p и q со значениями интенсивности из V , где V -ряд значений интенсивности, являются восьмисвязными, если q относится к группе $N_8(p)$.
3. **m -связный**(смешанная связь). Два пиксела p и q со значениями интенсивностей из V являются m -связными, если:

а) q относится к группе $N_4(p)$

б) q относится к группе $N_d(p)$ и множество $N_4(p) \cap N_4(q)$ – пустое.

Иными словами, это множество пикселов, являющихся четырьмя соседними, как по отношению к p , так и по отношению к q со значениями интенсивности из V .

33 Сжатие данных.

Важной областью применения ортогональных преобразований является сжатие данных. Если дискретный сигнал содержит N отсчётов, то его можно рассматривать как точку N -мерного пространства. Тогда каждый отсчёт является координатой N -мерного вектора данных. Для более эффективного представления можно осуществить ортогональное преобразование X , что приводит к: $Y = T * X$,

где Y и T – вектор коэффициентов преобразования и матрица преобразования соответственно. Целью сжатия данных является выбор подмножества M координат вектора Y , где M существенно меньше N . Остальные $(N - M)$ координат можно отбросить, не вызывая существенной ошибки при восстановлении сигнала по M координатам вектора Y . Следовательно, сравнивать ортогональные преобразования следует в соответствии с некоторым критерием ошибки. Одним из часто используемых является критерий среднеквадратической ошибки.

Поиск оптимального преобразования

Найдём ортогональное преобразование, которое с одной стороны будет обеспечивать представление сигнала, а с другой стороны будет оптимальным в смысле среднеквадратичного критерия.

Пусть T - ортогональное преобразование, заданное в виде: $T = [\phi_1 \ \phi_2 \ \dots \ \phi_N]$, где ϕ – векторы размерностью $N \times 1$. Для удобства базисные векторы $\{\phi_m\}_{m=1, \dots, N}$, будем считать вещественнозначными и ортонормированными.

Для каждого вектора X , принадлежащего данному классу векторов исходных данных получим: $Y = T * X$, где $X = [x_1 \ x_2 \ \dots \ x_N]$, $Y = [y_1 \ y_2 \ \dots \ y_N]$

Из соотношения (20.1) : $X = T^T Y = [y_1 \ y_2 \ \dots \ y_N] Y$, что можно записать как

$$X = y_1 \phi_1 + y_2 \phi_2 + \dots + y_N \phi_N = \sum_{i=1}^N y_i \phi_i$$

Желательно сохранить подмножество $\{y_1, y_2, \dots, y_M\}$ координат Y и при этом получить оценку X . Это может быть осуществлено заменой остальных $(N - M)$ координат Y заранее выбранными константами, что приводит к

$$\tilde{X}(M) = \sum_{i=1}^M y_i \phi_i + \sum_{i=M+1}^N b_i \phi_i,$$

где $\tilde{X}(M)$ обозначает оценку X . Ошибку, возникающую при отбрасывании $N - M$ координат можно представить в виде $\Delta X = X - \tilde{X}(M)$ т.е.

$$\Delta X = X - \sum_{i=1}^M y_i \phi_i - \sum_{i=M+1}^N b_i \phi_i = \sum_{i=M+1}^N y_i \phi_i - \sum_{i=M+1}^N b_i \phi_i = \sum_{i=M+1}^N (y_i - b_i) \phi_i.$$

Таким образом, среднеквадратичная ошибка $\varepsilon(M)$ определяется в виде

$$\varepsilon(M) = \varepsilon\{\|\Delta x\|^2\} = \varepsilon\{(\Delta x)^T \Delta x\} \quad \varepsilon(M) = \varepsilon\{\|\Delta x\|^2\} = \varepsilon\{(\Delta x)^T \Delta x\}$$

С учётом подстановки получим $\varepsilon(M) = \varepsilon\left\{ \sum_{i=M+1}^N \sum_{j=M+1}^N (y_i - b_i)^T (y_j - b_j) \phi_i^T \phi_j \right\},$

что в упрощённом виде можно записать как $\varepsilon(M) = \sum_{i=M+1}^N E\{(y_i - b_i)^2\}$

Из последнего соотношения следует, что для каждого выбора ϕ_i и b_i получаем определённое значение $\varepsilon(M)$. Требуется определить такую комбинацию этих величин, чтобы минимизировать $\varepsilon(M)$. Процедура выбора оптимальных ϕ_i и b_i разбивается соответственно на 2 этапа. **Первый этап** Оптимальное значение b_i определяется из выражения

$$\frac{\partial}{\partial b_i} E\{(y_i - b_i)^2\} = -2[E\{y_i\} - b_i] = 0 \quad \text{Тогда } E\{y_i\} = b_i. \quad \text{В свою очередь } y_i = \phi_i \cdot x.$$

Таким образом $b_i = \phi_i E\{x\} = \phi_i \bar{x}$ где $E\{x\} = \bar{x}$

Так как разность $(y_i - b_i)$ является скалярной, то $\varepsilon(M)$ можно записать как

$$\varepsilon(M) = \sum_{i=M+1}^N E\{(y_i - b_i)(y_i - b_i)^T\} \quad \text{Подстановка } y_i = \phi_i x \quad b_i = \phi_i \bar{x} \quad \text{в (20.1) приводит к}$$

$$\varepsilon(M) = \sum_{i=M+1}^N \phi_i E\{(x - \bar{x})(x - \bar{x})^T\} \phi_i^T$$

Так как $\sum_x = E\{(x - \bar{x})(x - \bar{x})^T\}$ является ковариационной матрицей x , то получим

$$\varepsilon(M) = \sum_{i=M+1}^N \phi_i \sum_x \phi_i^T \quad \text{Второй этап}$$

Для нахождения оптимального ϕ_i следует не только минимизировать

$$\varepsilon(M) \text{ по отношению к } \phi_i, \text{ но и удовлетворить ограничению } \phi_i^T \phi_i = 1$$

$$\text{При этом, используя метод множителей Лагранжа, получим } \sum_x \phi_i^T = \phi_i^T \lambda_i$$

Подставляя (20.4) в (20.3), получим минимальное значение

$$\text{среднеквадратической ошибки в виде } E_{\min}(M) = \sum_{i=M+1}^N \lambda_i \quad \text{Таким образом, разложение}$$

$$x = \sum_{i=1}^N y_i \phi_i \text{ представляет собой разложение по собственным векторам}$$

ковариационной матрицы. Это разложение называется разложением Карунена-Лозва. В литературе по статистике задачи по минимизации $\varepsilon(M)$ называются факторным анализом или анализом главных компонент.

Таким образом можно сделать 2 вывода:

1. Преобразование Карунена-Лозва является оптимальным для представления сигналов по отношению к критерию среднеквадратичной ошибки.

2. Поскольку $Y = TX$, то ковариационная функция в области изображения \sum_Y определяется как

$$\sum_Y = T \sum_X T^{-1} = T \sum_X T^T \quad \text{Так как } T \text{ состоит из собственных векторов } \sum_X, \text{ то}$$

$$\sum_Y = \text{diag}(\lambda_1, \lambda_2, \dots, \lambda_N)$$

где λ_i - собственные значения \sum_X . Так как \sum_Y - диагональная матрица, то координаты вектора преобразованных данных y_i в выражении $y_i = \phi_i^T x$ некоррелированы.

Из выражения $E_{\min}(M) = \sum_{i=M+1}^N \lambda_i$ следует, что эффективность коэффициента преобразования y_i для представления вектора данных X определяется соответствующим ему собственным значением. Если какой-либо коэффициент y_k не учитывается, то среднеквадратичная ошибка увеличивается на соответствующее собственное значение λ_k . Таким образом, необходимо выбрать множество y_i соответствующее M наибольшим собственным значениям, а остальные y_i отбросить, так как их можно заменить константами $b_i \quad i = \overline{M+1, N}$.

Так как $b_i = \phi_i^T \bar{X}$, то остальные y_i можно приравнять к нулю, если исходные данные центрированы, то есть $\bar{X} = 0$.

Так как собственные значения являются элементами \sum_Y , стоящими на главной диагонали, то они соответствуют дисперсиям коэффициентов преобразования y_i . Для всех остальных преобразований \sum_Y содержит ненулевые внедиагональные элементы. Поэтому естественным критерием при выборе множества оставляемых коэффициентов преобразования является сохранение M коэффициентов с наибольшими дисперсиями, а остальные $(N - M)$ коэффициентов можно отбросить. Приведённый выше критерий выбора коэффициентов преобразования носит название дисперсионного критерия.

Выше предполагалось, что X принадлежит одному классу сигналов, но все рассуждения легко распространяются на несколько классов. Отличие заключается в том, что базисные векторы для соответствующего преобразования Карунена-Лоэва являются собственными векторами общей матрицы \sum_x^* , а не \sum_x . Матрица \sum_x^* определяется как:

$$\sum_x^* = P_1 \sum_{x_1} + P_2 \sum_{x_2} + \dots + P_K \sum_{x_K}, \quad \text{Где } K - \text{число классов сигналов,}$$

\sum_{x_i} - ковариационная матрица i -ого класса с априорной вероятностью P_i .

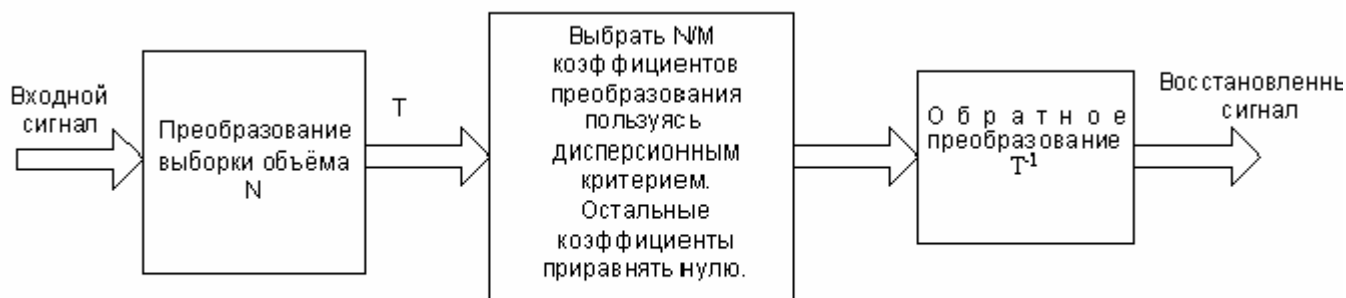


Рис. 20.1 Модель сжатия данных в соотношении M:1.

При цифровой обработке изображений каждый элемент обычно кодируется в виде слова, содержащего 6 бит. Таким образом, элемент изображения можно представить десятичным числом от 1 до 64 или от 0 до 63 (то есть 64 уровнями). Закодированные, таким образом, данные обычно обрабатываются блоками размером $N \times N$, как показано на рис.

Элемент изображения в i -ой строке и j -ом столбце можно представить в виде случайной величины $f(i, j)$, а матрицу ($N \times N$) случайных величин как $[f(i, j)]$. Тогда двумерное преобразование $[f(i, j)]$ и обратное преобразование можно записать как:

$$[F(u, v)] = T[f(i, j)]T' \quad [f(i, j)] = T'[F(u, v)]T,$$

где $[F(u, v)]$ - матрица коэффициентов преобразования, T - матрица преобразования.

Обозначим через $\sigma^2(i, j)$ и $\sigma^2(u, v)$ дисперсии $f(i, j)$ и $F(u, v)$

соответственно. Если функция распределения дисперсии $\sigma^2(u, v)$ неизвестна для изображения или класса изображений, подвергаемых преобразованию, её получают обычно моделированием. Один из подходов заключается в том что изображение описывается статистически как Марковский процесс первого порядка, а строки и столбцы обрабатываются независимо. Если дисперсия каждого столбца и строки равна σ^2 , то ковариационные матрицы для строк и столбцов можно записать как

$$\sum_k = \sigma^2 R_k \quad \text{где } R_k = \begin{pmatrix} 1 & \rho_k & \rho_k^2 & \dots & \rho_k^{N-1} \\ \rho_k & 1 & \rho_k & \rho_k^2 & \dots \\ \rho_k^2 & \rho_k & 1 & \rho_k & \dots \\ \dots & \dots & \dots & \dots & \dots \\ \rho_k^{N-1} & \dots & \dots & \dots & 1 \end{pmatrix}$$

а ρ_1 и ρ_2 - коэффициенты корреляции для строк и столбцов случайных величин соответственно

Тогда ковариационные матрицы в области преобразований, соответствующие \sum_k запишутся как:

$$\tilde{\sum}_k = \sigma^2 [TR_k T'] \quad k=1,2$$

В свою очередь, функция распределения дисперсии $\sigma^2(u, v)$

определяется как $\tilde{\sigma}^2(u, v) = \tilde{\sum}_1(u, u) \tilde{\sum}_2(v, v)$

Рассмотрим простейший способ сжатия данных, называемый зональным кодированием, который реализуется в 3 этапа:

1. Получают двумерное преобразование данного изображения, путём обработки блоками (NxN).
2. Из N^2 коэффициентов преобразования сохраняют N^2/m коэффициентов, обладающих наибольшими двумерными дисперсиями $\sigma^2(u, v)$. Все остальные коэффициенты приравниваются нулю.
3. Каждый из N^2/m сохранённых коэффициентов кодируется с помощью k бит. Затем восстанавливается соответствующий (NxN) блок с помощью обратного преобразования.

Так как кодируется только N^2/m коэффициентов преобразования, а не N^2 исходных элементов изображения для каждого (NxN) блока, то среднее число бит на элемент восстановленного изображения равняется R/m .

Следует отметить, что при заданном m выбор типа преобразования определяется объёмом вычислений, объёмом требуемой аппаратуры и требованиями, предъявляемыми к качеству изображения.

34 Вычисление оценки взаимной корреляционной функции.

Оценки взаимных корреляционных функций так же, как и оценки автоковариационных функций могут быть получены двумя основными способами: непосредственным вычислением и с использованием БПФ.

• Непосредственное вычисление. В этом случае несмещённые оценки взаимных ковариационных функций при сдвигах $r = \overline{0, m}$ ($m < N$) находятся в виде

$$\hat{R}_{xy}(r\Delta t) = \frac{1}{N-r} \sum_{n=0}^{N-r-1} x_n y_{n+r} \quad \hat{R}_{yx}(r\Delta t) = \frac{1}{N-r} \sum_{n=0}^{N-r-1} y_n x_{n+r}$$

Выборочную взаимную ковариационную функцию $\hat{R}_{xy}(r\Delta t)$ можно нормировать так? что значения её будут находиться между -1 и +1, для чего необходимо разделить её на величину $\sqrt{R_{xx}(0)}\sqrt{R_{yy}(0)}$. Таким образом, получается нормированная взаимная корреляционная функция:

$$\hat{\rho}_{xy}(r\Delta t) = \frac{\hat{R}_{xy}(r\Delta t)}{\sqrt{R_{xx}(0)}\sqrt{R_{yy}(0)}} \quad r = \overline{0, m}?$$

которая теоретически должна удовлетворять равенству $-1 \leq \hat{\rho}_{xy}(r\Delta t) \leq 1$.

Аналогично определяется функция $\hat{\rho}_{yx}(r\Delta t)$.

• Для получения взаимной ковариационной функции при помощи БПФ можно рекомендовать следующую последовательность вычислений.

Пусть объём каждой выборки из реализаций $x(t)$ и $y(t)$ составляет $N = 2^p$ чисел. Взаимную ковариационную функцию в этом варианте находят по спектральной мощности процесса, для чего дважды применяют БПФ – к функции $x(t)$ и к функции $y(t)$. Оба преобразования можно получить одновременно, приравняв одну реализацию действительной, а другую мнимой части некоторой комплексной $z(n) = x(n) + jy(n)$? $n = 0, 1, \dots, N-1$.

Тогда преобразование Фурье функции $z(n)$ будет иметь вид:

$$Z(k) = \sum_{n=0}^{N-1} [x(n) + jy(n)] \exp\left[-j \frac{2\pi kn}{N}\right]$$

и может быть вычислено по алгоритму БПФ. Следует отметить, что

$$\exp\left[j \frac{2\pi m(N-k)}{N}\right] = \exp\left[-j \frac{2\pi mk}{N}\right] \quad \text{Поскольку } \exp(j2\pi m) \equiv 1 \text{ при любых } n.$$

Следовательно, комплексно сопряжённая функция равна:

$$Z^*(N-k) = \sum_{n=0}^{N-1} [x(n) - jy(n)] \exp\left[-j \frac{2\pi nk}{N}\right] \quad \text{Тогда } \rightarrow$$

$$Z(k) + Z^*(N-k) = 2 \sum_{n=0}^{N-1} x(n) \exp\left[-j \frac{2\pi nk}{N}\right] = 2X(k)$$

$$Z(k) - Z^*(N - k) = 2j \sum_{n=0}^{N-1} y(n) \exp\left[-j \frac{2\pi kn}{N}\right] = 2jY(k)$$

Таким образом, преобразования Фурье $X(k)$ и $Y(k)$ двух действительных реализаций $x(n)$ и $y(n)$ будут иметь вид:

Этим же приёмом можно воспользоваться для расчёта преобразования на интервале вдвое большей длины. Для этого вещественная реализация $v(n)$ ($n = 0, 1, \dots, 2N - 1$) разделяется на две части: одна состоит из значений $v(n)$ при чётном n , другая – при нечётном. Иными словами:

$$x(n) = v(2n) \quad y(n) = v(2n + 1) \quad n = \overline{0, N-1}$$

Рекомендуемая в итоге последовательность операций для получения оценки взаимной ковариационной функции основана на предположении, что длина исходных реализаций $x(n\Delta t)$, $y(n\Delta t)$ есть $n_d N$, $N = 2^p$.

1. Выбирается максимальный сдвиг m и исходные реализации разбиваются на n_d отрезков, каждый из которых состоит из $N > m$ отсчётов.
2. Каждый отрезок $\{x_n\}$, $\{y_n\}$, $n = \overline{1, N}$ дополняется N нулями, в результате чего новый отрезок содержит $2N$ отсчётов.
3. По отрезкам, содержащим по $2N$ точек в соответствии с

$$Z(k) = \sum_{n=0}^{2N-1} [x(n) + jy(n)] \exp\left[-j \frac{2\pi kn}{2N}\right] \quad k = \overline{0, 2N-1}$$

Методом БПФ вычисляются коэффициенты $Z(k)$. 4. На основе соотношений

$$X(k) = \frac{Z(k) + Z^*(N - k)}{2} \quad Y(k) = \frac{Z(k) - Z^*(N - k)}{2j} \quad k = \overline{0, 2N-1}$$

Вычисляются $X(k)$ и $Y(k)$.

5. По методу оценивания взаимной спектральной плотности мощности вычисляется оценка $\hat{S}_{xy}(f_k)$ двухсторонней спектральной плотности.

6. На основе соотношения $x_n = \frac{1}{N} \sum_{k=0}^{N-1} x_k \exp\left[j \frac{2\pi kn}{N}\right]$ вычисляется обратное БПФ последовательности $\hat{S}_{xy}(f_k)$, дающее оценку $\hat{R}_{xy}^s(r\Delta t)$, $r = \overline{0, 2N-1}$.

7. Несмещённая оценка $\hat{R}_{xy}(r\Delta t)$ взаимной ковариационной функции при положительном сдвиге $r = \overline{0, N-1}$ находится умножением

$$\hat{R}_{xy}^s(r\Delta t) \text{ на масштабные коэффициенты } \frac{N}{(N-r)}.$$

8. Несмещённая оценка $\hat{R}_{xy}(r\Delta t)$ взаимной ковариационной функции при отрицательном сдвиге $r = \overline{0, N-1}$ находится умножением

при отрицательном сдвиге $r = \overline{0, N-1}$ находится умножением $\hat{R}_{xy}^s(r\Delta t)$ на масштабные коэффициенты $\frac{N}{(r-N)}$ ($r = \overline{(N+1), 2N}$).

35. 37. Оценка взаимной спектральной плотности стационарного случайного процесса. Оценка спектра мощности стационарного случайного процесса.

Для определения оценок спектральной плотности мощности используют корреляционный метод, основанный на использовании соотношения

$$\hat{G}(f) = 4 \int_0^T \tilde{R}(\tau) \cos 2\pi f \tau d\tau$$

То есть вначале с использованием (12.1) рассчитывалась оценка ковариационной функции, а затем оценка спектральной плотности мощности вычислялась как преобразование Фурье при соответствующем значении максимального сдвига.

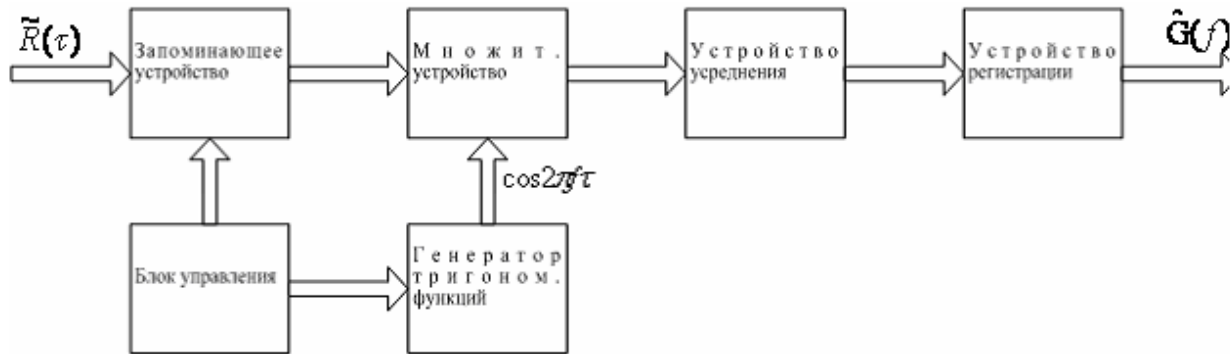


Рис. 12.1.

Исходная оценка функции ковариации вводится в ЗУ, с помощью которой она периодически воспроизводится и анализируется затем на заданных частотах с помощью соотношения (12.1). Синхронизация воспроизведения $\tilde{R}(\tau)$ с фазой выходных колебаний генератора

производится с помощью блока управления и получаемые значения СПМ фиксируются в устройстве регистрации.

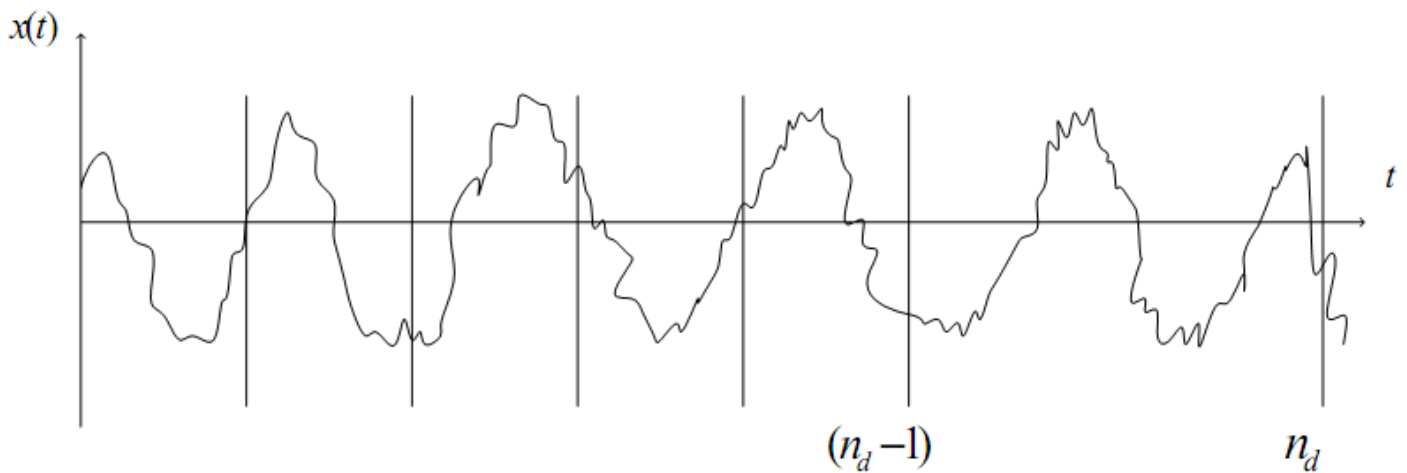
Разобьём реализацию на n_d смежных отрезков длиной T каждый и обозначим эти отрезки через x_i :

$$(i-1)T \leq t \leq iT \quad i = \overline{1, n_d}$$

Тогда оценка спектральной плотности мощности на произвольной частоте f может быть записана как:

$$\tilde{S}_{xx}(f) = \frac{1}{n_d T} \sum_{i=1}^{n_d} |x_i(f, T)|^2 \quad (12.2)$$

$$\text{где } X_i(f, T) = \int_0^T x_i(t) e^{-i2\pi ft} dt$$



При дискретном временном параметре каждая реализация $x_i(t)$ будет представлена N значениями временного ряда $\{x_m\}$, $m = \overline{0, N-1}$. Конечное преобразование Фурье даст соответствующее соотношению (12.2) значения спектральной плотности мощности на дискретных частотах

$$f_k = \frac{k}{T} = \frac{k}{N\Delta t} \quad k = \overline{0, N-1}$$

При этом коэффициенты Фурье для каждого отрезка определяются как

$$x_i(f_k) = \Delta t \sum_{n=0}^{N-1} x_m \exp\left(\frac{-j2\pi kn}{N}\right) \quad (12.3)$$

При этом оценка двусторонней спектральной плотности принимает вид

$$\hat{S}_{xx}(f_k) = \frac{1}{n_d N \Delta t} \sum_{i=1}^{n_d} |x_i(f_k)|^2 \quad k = \overline{0, N-1} \quad (12.4)$$

В свою очередь, односторонняя функция СПМ оценивается как:

$$\hat{G}_{xx}(f_k) = \frac{2}{n_d N \Delta t} \sum_{i=1}^{n_d} |x_i(f_k)|^2 \quad k = \overline{0, N/2} \quad (12.5)$$

Число отсчётов N в соотношении для СПМ при каждом БПФ называют размером блока и оно служит ключевым параметром, определяющим разрешающую способность по частоте

$$\Delta f = \frac{1}{T} = \frac{1}{N\Delta t}$$

Кроме того, при осуществлении БПФ по алгоритму Кули – Тьюки размер выборки удобно выбирать как целую степень числа 2, т.е. $N = 2^p$.

Определённое формулой (12.3) конечное преобразование Фурье функции $x(t)$ можно рассматривать как преобразование Фурье заданной на бесконечном интервале функции $v(t)$, умноженной на прямоугольное «временное окно» $u(t)$, имеющее вид:

$$u(t) = \begin{cases} 1, & 0 \leq t \leq T \\ 0, & \text{в остальных случаях} \end{cases}$$

Иными словами реализацию $x(t)$ можно рассматривать как произведение $x(t) = u(t)v(t)$.



Рис. 12.3

Следовательно, преобразование Фурье функции $x(t)$ есть свёртка преобразования функций $u(t)$ и $v(t)$:

В свою очередь, преобразование Фурье прямоугольной функции $u(t)$ будет иметь вид:

$$x(f) = \int_{-\infty}^{\infty} u(\alpha)v(f - \alpha)d\alpha$$

В свою очередь, преобразование Фурье прямоугольной функции $u(t)$ будет иметь вид:

$$u(t) = T \left(\frac{\sin n\pi f T}{\pi f T} \right) e^{-i\pi f T}$$

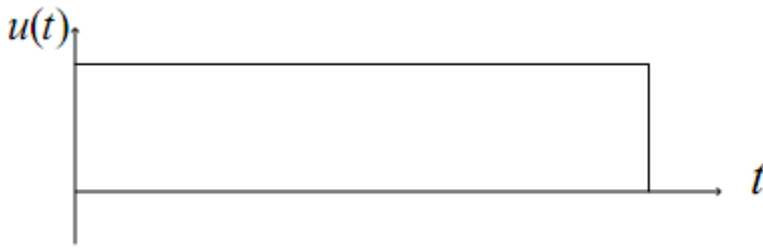


Рис. 12.4

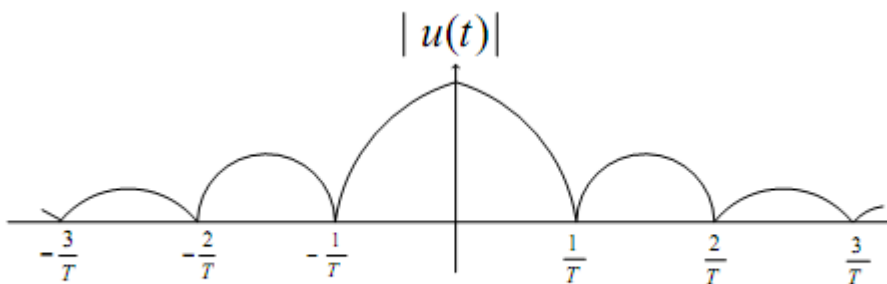


Рис. 12.5

Отметим, что описанная функция представляет собой основное спектральное окно, используемое при анализе. Большие боковые максимумы функции $|u(t)|$ обуславливают большое просачивание энергии с частот, далёких от главного максимума спектрального окна и могут привести к сильным искажениям спектральной оценки, особенно заметным для процессов с узкополосным спектром. Проблема просачивания не возникает при анализе периодических процессов (с периодом T_p).

Для подавления просачивания обычно используют временные окна, сглаживающие исходную реализацию таким образом, чтобы подавить резкие вариации на её начальном и конечном участках.

Существует множество таких окон, но одно из наиболее ранних и широко используемых – косинусоидальное сглаживающее окно, называемое окном Ханна.

$$U_h(t) = \begin{cases} \frac{1}{2} \left(1 - \cos \frac{2\pi t}{T} \right) = 1 - \cos^2 \left(\frac{\pi t}{T} \right), & 0 \leq t \leq T \\ 0, & \text{в остальных случаях} \end{cases}$$

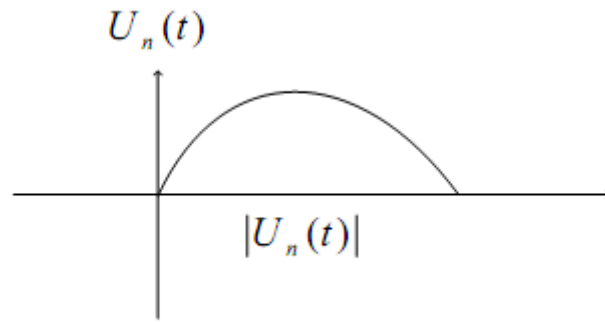


Рис. 12.6

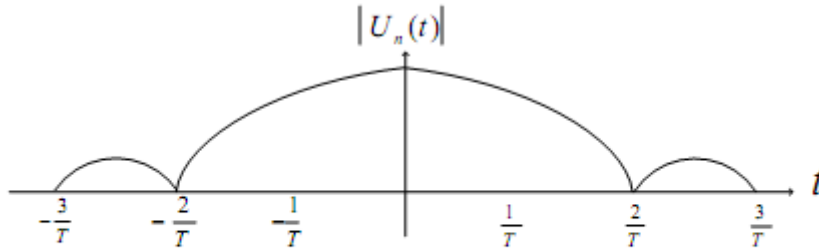


Рис. 12.7

В этом случае $x(t) = u_n(t)v(t)$.

Её преобразование Фурье есть

$$x(f) = \int_0^T x(t)e^{-2\pi ft} dt = \int_{-\infty}^{\infty} u_n(\alpha)v(\alpha)d\alpha$$

Для дискретных значений частоты $f_k = k/T$ при $k = \overline{0, N/2}$ получим:

$$x(f_k) = \frac{1}{2}v(f_k) - \frac{1}{4}v(f_{k-1}) - \frac{1}{4}v(f_{k+1}) \quad \text{где } v(f_k) = \int_0^T v(t)e^{-i2\pi k t/T} dt$$

Предположим, что в пределах каждой полосы частот шириной $\Delta f = 1/T$ функция $v(t)$ ведёт себя как ограниченный по частоте белый шум. Тогда математическое ожидание произведения $V^*(f)$ и $V(\zeta)$, вычисленное для любых частот f и ζ из набора $k\Delta f = k/T$

$$E[V^*(f)V(\zeta)] = \begin{cases} 0, & f \neq \zeta \\ 1, & f = \zeta \end{cases}$$

С учётом этого свойства

$$E[|x(f_k)|^2] = \left(\frac{1}{2}\right)^2 + \left(\frac{1}{4}\right)^2 + \left(\frac{1}{4}\right)^2 = \frac{3}{8} \quad \text{при любых } f_k = k/T.$$

Эта величина характеризует потери, вызванные использованием спектрального окна Ханна для оценки спектральной плотности при помощи конечного преобразования Фурье. Поэтому при оценивании спектров необходимо умножение на масштабный множитель $\sqrt{8/3}$. В частности при вычислении спектральной плотности по формуле (12.3) с использованием окна Ханна следует использовать соотношение

$$x_i(f_k) = \Delta t \sqrt{8/3} \sum_{n=0}^{N-1} x_{in} (1 - \cos^2 \frac{\pi n}{N}) \exp \left[-j \frac{2\pi k n}{N} \right]$$

Для вычисления оценки СПМ необходимо использовать следующие этапы:

1. Имеющаяся реализация x_n разбивается на n_d отрезков, каждый из которых содержит N отсчётов.
2. При необходимости подавить просачивание энергии через боковые максимумы каждый отрезок $\{x_n\} \quad n = \overline{0, N-1}$ сглаживается фильтром, или весовым окном Ханна, или другим подходящим образом выбранным фильтром (к примеру, Баттерворта, Сорта, Хэмминга, Дольфа-Чебышева и т.д).
3. Для каждого отрезка реализации выбирается N значений БПФ $x(f_k)$ $k = \overline{0, N-1}$.
4. Значения $x(f_k)$ умножаются на масштабный коэффициент для компенсации изменений, внесённых сглаживанием (при использовании окна Ханна этот коэффициент равен $\sqrt{8/3}$).
5. По формулам (12.4) и (12.5) вычисляется соответственно оценка двусторонней и односторонней спектральной плотности для n_d отрезков.

Косвенный способ оценивания ковариационной функции основан на использовании соотношения Винера – Хинчина. Оценка ковариационной функции находится как обратное преобразование Фурье спектральной плотности.

В случае дискретной последовательности $\{x_n\}$ $n = \overline{0, N-1}$ циклическая ковариационная функция принимает вид:

$$\hat{R}_{xx}^c(r\Delta t) = \frac{N-r}{N} \hat{R}_{xx}(r\Delta t) + \frac{r}{N} \hat{R}_{xx}[(N-r)\Delta t]$$

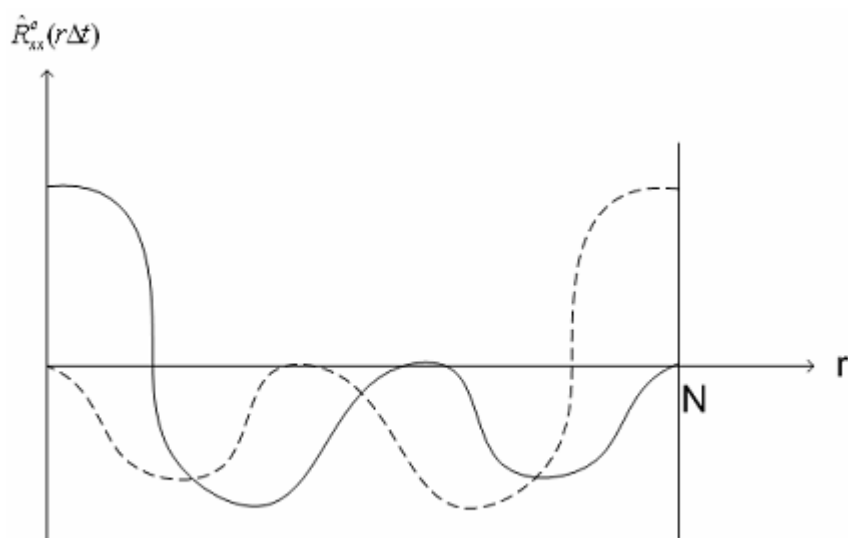
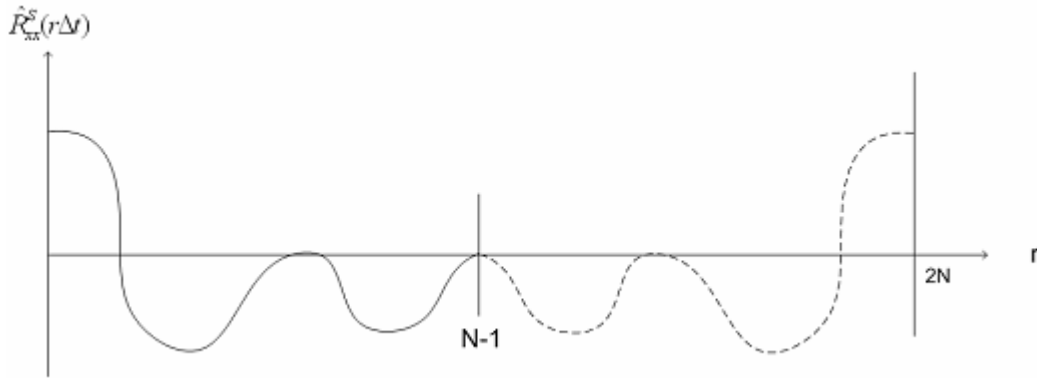


Рис. 12.8

Как видно из рисунка, имеет место эффект наложения. Этого осложнения можно избежать, дополняя исходный временной ряд нулями. В результате этого происходит разделение обеих частей циклической оценки ковариационной функции.

В частности, при добавлении N нулей к исходному временному ряду той же длины будет иметь место полное разделение, т.е.

$$\hat{R}_{xx}^s = \begin{cases} \frac{N-r}{N} \hat{R}_{xx}(r\Delta t), & r = 0, 1, \dots, (N-1) \\ \frac{r-N}{N} \hat{R}_{xx}[(2N-r)\Delta t], & r = N, (N+1), \dots, (2N-1) \end{cases}$$



Заметим, что первая половина этой оценки при $0 \leq r \leq N-1$ представляет собой значения ковариационной функции при положительных значениях сдвига ($0 \leq r \leq m$), а вторая ($N \leq r \leq 2N-1$) при отрицательных значениях сдвига. Но поскольку ковариационная функция есть функция чётная, то вторую половину можно отбросить и окончательно несмещённая оценка ковариационной функции находится в виде

$$\hat{R}_{xx}(r\Delta t) = \frac{N}{N-r} \hat{R}_{xx}^s(r\Delta t)$$

Полученная оценка статистически эквивалентна прямой оценке (непосредственно по реализации случайного процесса). Однако, в зависимости от величины её такой косвенный метод с использованием БПФ может потребовать заметно меньшего объёма вычислений. Действительно, косвенный метод требует нахождения вначале оценки спектральной плотности мощности, для чего осуществляется БПФ по n_d независимым реализациям, каждая из которых содержит N отсчётов и дополнена N нулями, так что БПФ производится для реализации длиной $2N$. Затем $2N$ значений сглаженной спектральной оценки подвергаются обратному преобразованию Фурье, что в общей сложности даёт (n_d+1) быстрых преобразований Фурье, каждое из которых требует $4Np$ операций над действительными числами, где $N = \log_2 N$. Для реализации прямого метода потребуется $m n_d N$ операций. Следовательно, при одном и том же максимальном сдвиге $m=N$ коэффициент ускорения вычислений равен:

$$к.у.в. = \frac{n_d N^2}{(n_d + 1)4Np} \approx \frac{N}{4p} \quad \text{При } N=1024 \text{ к.у.в.} = 1024/40 \approx 26.$$

Оценки взаимных корреляционных функций так же, как и оценки автоковариационных функций могут быть получены двумя основными способами: непосредственным вычислением и с использованием БПФ.

- Непосредственное вычисление.

В этом случае несмещённые оценки взаимных ковариационных функций при сдвигах $r = \overline{0, m}$ ($m < N$) находятся в виде

$$\hat{R}_{xy}(r\Delta t) = \frac{1}{N-r} \sum_{n=0}^{N-r-1} x_n y_{n+r} \quad \hat{R}_{yx}(r\Delta t) = \frac{1}{N-r} \sum_{n=0}^{N-r-1} y_n x_{n+r}$$

Выборочную взаимную ковариационную функцию $\hat{R}_{xy}(r\Delta t)$ можно нормировать так? что значения её будут находиться между -1 и +1, для чего необходимо разделить её на величину $\sqrt{R_{xx}(0)}\sqrt{R_{yy}(0)}$. Таким образом, получается нормированная взаимная корреляционная функция:

$$\hat{\rho}_{xy}(r\Delta t) = \frac{\hat{R}_{xy}(r\Delta t)}{\sqrt{R_{xx}(0)}\sqrt{R_{yy}(0)}} \quad r = \overline{0, m}?$$

которая теоретически должна удовлетворять равенству $-1 \leq \hat{\rho}_{xy}(r\Delta t) \leq 1$.

Аналогично определяется функция $\hat{\rho}_{yx}(r\Delta t)$.

Для получения взаимной ковариационной функции при помощи БПФ можно рекомендовать следующую последовательность вычислений. Пусть объём каждой выборки из реализаций $x(t)$ и $y(t)$ составляет $N = 2^p$ чисел. Взаимную ковариационную функцию в этом варианте находят по спектральной мощности процесса, для чего дважды применяют БПФ – к функции $x(t)$ и к функции $y(t)$. Оба преобразования можно получить одновременно, приравняв одну реализацию действительной, а другую мнимой части некоторой комплексной реализации:

$$z(n) = x(n) + jy(n) \quad n = 0, 1, \dots, N-1.$$

Тогда преобразование Фурье функции $z(n)$ будет иметь вид:

$$Z(k) = \sum_{n=0}^{N-1} [x(n) + jy(n)] \exp\left[-j \frac{2\pi kn}{N}\right]$$

и может быть вычислено по алгоритму БПФ. Следует отметить, что

$$\exp\left[j \frac{2\pi n(N-k)}{N}\right] = \exp\left[-j \frac{2\pi nk}{N}\right]$$

Поскольку $\exp(j2\pi n) \equiv 1$ при любых n .

Следовательно, комплексно сопряжённая функция равна:

$$Z^*(N-k) = \sum_{n=0}^{N-1} [x(n) - jy(n)] \exp\left[-j \frac{2\pi nk}{N}\right] \quad \text{Тогда}$$

$$Z(k) + Z^*(N-k) = 2 \sum_{n=0}^{N-1} x(n) \exp\left[-j \frac{2\pi nk}{N}\right] = 2X(k)$$

$$Z(k) - Z^*(N-k) = 2j \sum_{n=0}^{N-1} y(n) \exp\left[-j \frac{2\pi nk}{N}\right] = 2jY(k)$$

Таким образом, преобразования Фурье $X(k)$ и $Y(k)$ двух действительных реализаций $x(n)$ и $y(n)$ будут иметь вид:

$$X(k) = \frac{Z(k) + Z^*(N-k)}{2} \quad Y(k) = \frac{Z(k) - Z^*(N-k)}{2j}$$

Этим же приёмом можно воспользоваться для расчёта преобразования на интервале вдвое большей длины. Для этого вещественная реализация $v(n)$ ($n = 0, 1, \dots, 2N-1$) разделяется на две части: одна состоит из значений $v(n)$ при чётном n , другая – при нечётном. Иными словами:

$$x(n) = v(2n) \quad y(n) = v(2n+1) \quad n = \overline{0, N-1}$$

Рекомендуемая в итоге последовательность операций для получения оценки взаимной ковариационной функции основана на предположении, что длина исходных реализаций $x(n\Delta t)$, $y(n\Delta t)$ есть $n_d N$, $N = 2^p$.

1. Выбирается максимальный сдвиг m и исходные реализации разбиваются на n_d отрезков, каждый из которых состоит из $N > m$ отсчётов.
2. Каждый отрезок $\{x_n\}$, $\{y_n\}$, $n = \overline{1, N}$ дополняется N нулями, в результате чего новый отрезок содержит $2N$ отсчётов.
3. По отрезкам, содержащим по $2N$ точек в соответствии с

$$Z(k) = \sum_{n=0}^{2N-1} [x(n) + jy(n)] \exp\left[-j \frac{2\pi kn}{2N}\right] \quad k = \overline{0, 2N-1}$$

Методом БПФ вычисляются коэффициенты $Z(k)$.

4. На основе соотношений

$$X(k) = \frac{Z(k) + Z^*(N-k)}{2} \quad Y(k) = \frac{Z(k) - Z^*(N-k)}{2j}$$

$$k = \overline{0, 2N-1}$$

Вычисляются $X(k)$ и $Y(k)$.

5. По методу оценивания взаимной спектральной плотности мощности вычисляется оценка $\hat{S}_{xy}(f_k)$ двухсторонней спектральной плотности.
6. На основе соотношения

$$x_n = \frac{1}{N} \sum_{k=0}^{N-1} x_k \exp\left[j \frac{2\pi kn}{N}\right]$$

вычисляется обратное БПФ последовательности $\hat{S}_{xy}(f_k)$, дающее оценку $\hat{R}_{xy}^S(r\Delta t)$, $r = \overline{0, 2N-1}$.

7. Несмещённая оценка $\hat{R}_{xy}(r\Delta t)$ взаимной ковариационной функции при положительном сдвиге $r = \overline{0, N-1}$ находится умножением $\hat{R}_{xy}^S(r\Delta t)$ на масштабные коэффициенты $N/(N-r)$.
8. Несмещённая оценка $\hat{R}_{xy}(r\Delta t)$ взаимной ковариационной функции при отрицательном сдвиге $r = \overline{0, N-1}$ находится умножением $\hat{R}_{xy}^S(r\Delta t)$ на масштабные коэффициенты $N/(r-N)$ ($r = \overline{(N+1), 2N}$).

По аналогии с оцениванием автоспектров можно рассматривать следующие этапы оценивания функций взаимной спектральной плотности:

1. Имеющиеся реализации разделяются на n_d отрезков x_n, y_n , каждый из которых содержит N отсчётов.
2. При необходимости подавить просачивание энергии через боковые максимумы каждый отрезок x_n, y_n ($n = \overline{0, N-1}$) сглаживается фильтром Ханна или любым другим подходящим образом выбранным фильтром.
3. Сглаженные значения x_n запоминаются как действительная часть, а сглаженные значения y_n - как мнимая часть последовательности $z(n) = x(n) + jy(n)$ ($n = \overline{0, N-1}$).

4. По формуле
$$Z(k) = \sum_{n=0}^{2N-1} [x(n) + jy(n)] \exp\left[-j \frac{2\pi kn}{2N}\right] \quad k = \overline{0, N-1}$$

для каждого отрезка вычисляются N значений БПФ $Z(k)$.

5. По формулам
$$\begin{cases} X(k) = \frac{Z(k) + Z^*(N-k)}{2} \\ Y(k) = \frac{Z(k) - Z^*(N-k)}{2j} \end{cases} \quad k = \overline{0, N-1}$$

для каждого отрезка вычисляются коэффициенты $X(k)$ и $Y(k)$, $k = \overline{0, N-1}$.

6. Значения $X(k)$ и $Y(k)$ умножаются на масштабный коэффициент для компенсации внесённых сглаживанием изменений (при использовании окна Ханна этот коэффициент равен $\sqrt{8/3}$).

7. Для каждой пары отрезков по значениям $X^*(f_k) = \Delta t X^*(k)$, $Y(f_k) = \Delta t Y(k)$ вычисляется оценка двухсторонней взаимной спектральной плотности

$$\tilde{S}_{xy}(f_k) = \frac{1}{N\Delta t} [X^*(f_k)Y(f_k)] \quad k = \overline{0, N-1}$$

или односторонней взаимной спектральной плотности

$$\hat{G}_{xy}(f_k) = \frac{2}{N\Delta t} [X^*(f_k)Y(f_k)] \quad k = \overline{0, N/2}$$

8. Осреднение оценок по n_d парам отрезков даёт окончательную сглаженную оценку $\hat{S}_{xy}(f_k)$ ($k = \overline{0, N-1}$) или $\hat{G}_{xy}(f_k)$ ($k = \overline{0, N/2}$).

36. Задачи распознавания образов

Задача распознавания образов заключается в классификации некоторой группы объектов на основе определённых требований.

Требования эти могут быть различными, так как в различных ситуациях возникает необходимость в определённом типе классификатора. К примеру при распознавании английских букв образуется 26 классов образов. Однако, чтобы отличить при распознавании буквы от китайских иероглифов нужны лишь два класса образов.

Проблема распознавания получила широкое распространение при решении различных задач в различных областях народного хозяйства: распознавание рукописных букв или слов, медицинская диагностика, диагностика неисправностей в технических системах, обнаружение цели, прогноз погоды, классификация сейсмических волн.

Простейший подход к распознаванию образов заключается в сопоставлении с эталоном. В этом случае некоторое множество образов по одному из каждого класса образов хранится в памяти машины. Входной, подлежащий распознаванию образ (неизвестного класса) сравнивается с эталоном каждого класса. Классификация основывается на заранее выбранном критерии соответствия или критерии подобия. Иными словами, если входной образ лучше соответствует эталону i -го класса образов, чем любому другому эталону, то входной образ классифицируется как принадлежащий i -му классу образов.

Более совершенный подход заключается в том, что вместо сравнения входного образа с эталонами, классификация основывается на некотором множестве отобранных замеров, называемых информативными признаками, которые предполагаются инвариантными или малочувствительными по отношению к изменениям и искажениям.

В этом случае распознавание образов можно рассматривать состоящим из двух задач.

Первая задача заключается в выборе подмножества признаков и их упорядочиванию в заданном множестве измерений. Вторая задача заключается в классификации (принятии решения о принадлежности входного образа тому или иному классу), которая основывается на измерениях отобранных признаков.

Упрощенная блок-схема системы распознавания приведена на рис. 1.

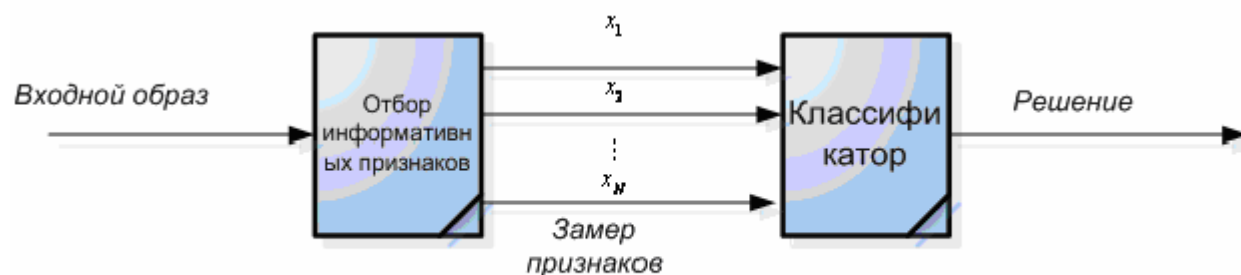


Рис. 21.1

Таким образом, подход, заключающийся в сопоставлении с эталоном, можно рассматривать как частный случай второго подхода - подхода измерения признаков, при этом эталоны хранятся в виде измеренных признаков, а в классификаторе используется специальный критерий сопоставления.

Математически задача классификации может быть сформулирована с помощью разделяющей функции.

Пусть $\omega_1, \omega_2, \dots, \omega_m$ обозначают m возможных классов образов, подлежащих распознаванию и пусть $X' = [x_1, x_2, \dots, x_n]$ вектор замеров признаков, где x_k предоставляет k -ый замер.

Тогда разделяющая функция $D_j(x)$, относящаяся к классу образов $\omega_j, j = \overline{1, m}$, такова, что если входной образ представленный вектором признаков X , принадлежит классу ω_i , то величина $D_i(x)$ должна быть наибольшей. Пусть $X \sim \omega_i$ обозначает, что вектор признаков X входного образа принадлежит классу ω_i . Тогда можно записать, что для всех $X \sim \omega_i$

$$D_i(x) > D_j(x); i, j = \overline{1, m} \quad i \neq j \quad (21.1)$$

Таким образом, в пространстве признаков Ω_x граница разбиений, называемая решающей границей между областями, относящимися соответственно к классам ω_i, ω_j , выражается уравнением

$$D_i(x) - D_j(x) = 0$$

Общая схема классификатора, использующего критерий (21.1) и типичный двумерный пример приведены на рис. 21.2 и рис. 21.3.

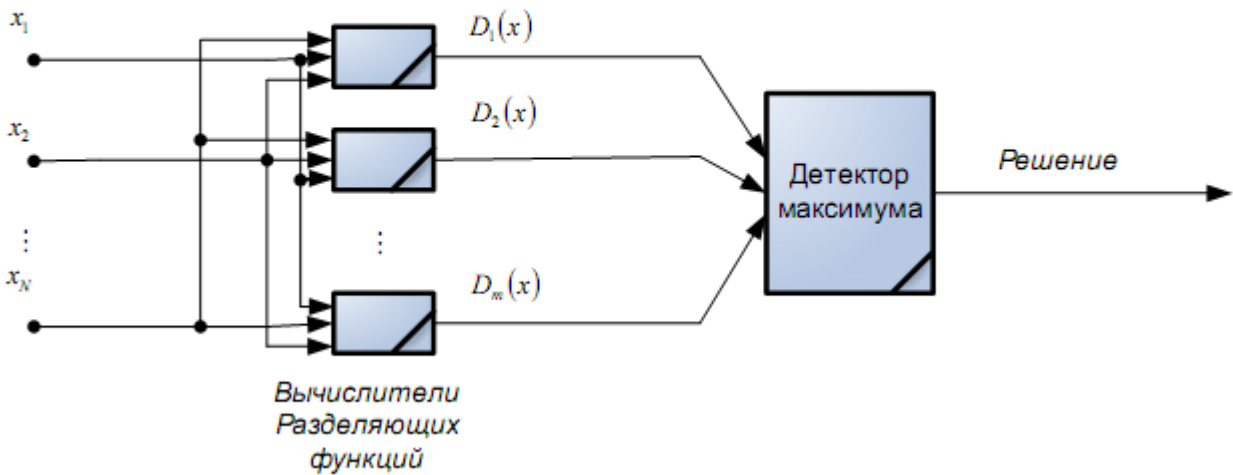


Рис. 21.2

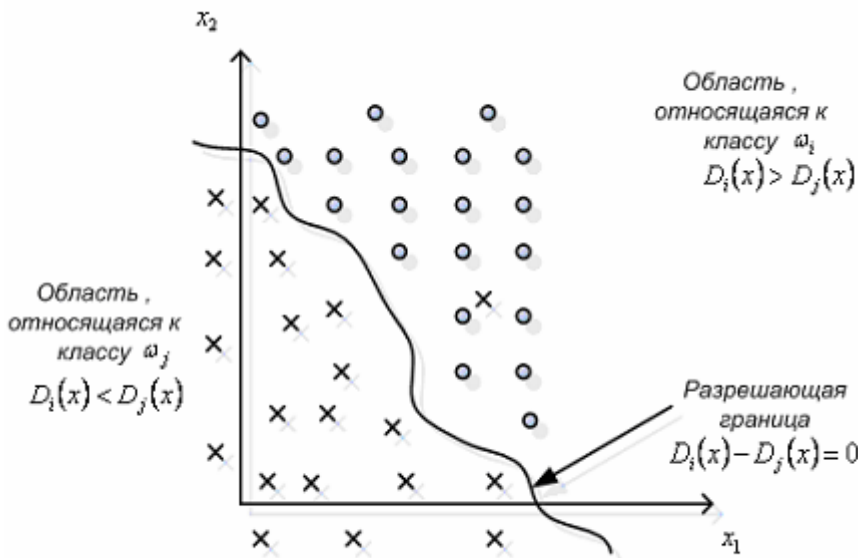


Рис. 21.3

38. Алгоритм бинаризации изображений.

При бинаризации изображения яркость каждого пикселя $B(x, y)$ сравнивается с пороговым значением яркости $B_T(x, y)$; если значение яркости пикселя выше значения яркости порога, то на бинарном изображении соответствующий пиксель будет «белым», или «черным» в противном случае. Необходимость устранения большого числа ошибок процесса бинаризации повлекла за собой появление большого числа методов бинаризации, которые делятся на две группы по принципу построения пороговой поверхности: методы глобальной и локальной бинаризации. Пороговой поверхностью является матрица размерностью $M \times N$, соответствующей размерности исходного изображения, каждая ячейка матрицы задает порог яркости бинаризации для соответствующего пикселя на исходном изображении. В методах глобальной бинаризации пороговая поверхность является плоскостью с постоянным значением пороговой яркости, а в методах локальной бинаризации значение пороговой яркости меняется от точки к точке изображения, и рассчитывается на основе некоторых локальных признаков в окрестности пикселя.

Метод Ниблэка

Простота метода Ниблэка локальной адаптивной бинаризации позволяет достичь высокую скорость обработки изображений (см. [табл. 1](#)). Метод используется на практике для быстрой фильтрации контрастных изображений, на которых отсутствуют сильно зашумленные области с плавными переходами яркости.

Идея данного метода состоит в варьировании порога яркости B бинаризации от точки к точке на основании локального значения стандартного отклонения. Порог яркости в точке (x, y) рассчитывается так:

$$B(x, y) = m(x, y) + k * s(x, y),$$

где $m(x, y)$ $s(x, y)$ – среднее и стандартное отклонение выборки для некоторой окрестности точки. Размер окрестности должен быть минимальным, но таким, чтобы сохранить локальные детали изображения. В то же время размер должен быть достаточно большим, чтобы понизить влияние шума на результат (для изображений, приведенных здесь, радиус окрестности составляет 16 пикселей). Значение k определяет, какую часть границы объекта взять в качестве самого объекта. Значение $k = -0.2$ задает достаточно хорошее разделение объектов, если они представлены черным цветом, а значение $k = +0.2$, – если объекты представлены белым цветом.

В местах плавного перехода яркости метод дает ложные объекты с небольшим шумом (см. [рис. 56](#)). Метод получил свое распространение на практике благодаря его интеграции с этапом постпроцессинга. При этом скорость обработки падает в 3 раза, и количество ошибок сокращается на 20% (см. [табл. 1](#)).

Удаление ложных объектов

Необходимость коррекции ошибок процесса бинаризации обусловила появление алгоритма удаления ложных объектов. Применяется он после бинаризации для устранения ложных областей, которые не соответствуют реальным объектам на изображении, и для устранения шума, порождаемого бинаризацией.



Рисунок 2. Возникновение ложных объектов при бинаризации.

Заключение

Существующие алгоритмы бинаризации позволяют проводить обработку изображений со значительной зональной неравномерностью яркости, с монотонными областями яркости, с сильно зашумленными изображениями. Однако остаются нерешенным ряд проблем. К таким проблемам относится обнаружение зон наложения объектов изображения и необходимость автоматической интерпретации данных зон. Другими словами, процесс подготовки документа требует определения конфигурации объекта в местах его наложения с другими объектами. Не является удовлетворительным часто используемый вариант «обрывания» объекта в месте наложения, т. к. на реальных документах, соответствующих обрабатываемым изображениям, объекты образуют большое количество наложений, и «обрывание» приводит к появлению серьезной ошибки – потери целостности объектов. Неудовлетворительным является и вариант, в котором зона наложения считается частью каждого из прилегающих к ней объектов.

Реализовать алгоритм бинаризации, способный решать проблему интерпретации зон наложения объектов без интеграции его с процедурой распознавания образов, невозможно. Подобного рода интеграция определяет дальнейшее развитие методов бинаризации и расслоения.

39. Метрические свойства для изображений

Расстояния между двумя векторами X и Y размерностью $n \times 1$

удовлетворяют трем аксиомам:

- 1) $d(X, Y) = 0$, если $X = Y$
- 2) $d(X, Y) = d(Y, X)$
- 3) $d(X, Y) \leq d(X, Z) + d(Z, Y)$, для всех Z

Наиболее важной метрикой является l^p -метрика

$$l^p(x, y) = \left(\sum_{i=1}^n |x_i - y_i|^p \right)^{1/p} \text{ для } p \geq 1$$

Если $p=2$, мы получим классическую Евклидову метрику.

Определим для пикселей p, q, z соответственно с координатами (x, y) ,

(s, t) и (u, v) функцию расстояния или метрику D следующим образом:

- 1) $D(p, q) \geq 0$ [$D(p, q) = 0$, если $p = q$]
- 2) $D(p, q) = D(q, p)$ (симметрия)
- 3) $D(p, z) \leq D(p, q) + D(q, z)$ (неравенство треугольника)

Евклидово расстояние между p и q определяется из соотношения:

$$D_E(p, q) = [(x-s)^2 + (y-t)^2]^{1/2}$$

При измерении этого расстояния пиксели, имеющие расстояние? меньшее или равное некоторой величине r от (x, y) располагаются в окружности радиусом r с центром в (x, y) .

Расстояние D_4 , называемое модульным между p и q , определяется соотношением:

$$D_4(p, q) = |x-s| + |y-t|$$

Метрику D_4 иногда называют метрикой городских кварталов, из-за того, что расстояния измеряются параллельно осям координат.

Покажем, что функция D_4 удовлетворяет к примеру, условию 3.

Запишем соотношение:

$$D_4(p, q) + D_4(q, z) = |x-s| + |y-t| + |s-u| + |t-v|$$

Но, согласно неравенству треугольника для вещественных чисел будем иметь:

$$|x-s| + |s-u| \geq |x-u|,$$

$$|y-t| + |t-v| \geq |y-v|,$$

$$D_4(p,q) + D_4(q,z) \geq |x-u| + |y-v| \geq D_4(p,z), \text{ т.е.}$$

$D_4(p,z) \leq D_4(p,q) + D_4(q,z)$, что и требовалось доказать.

В этом случае пиксели, имеющие расстояние D_4 , меньшее или равное некоторой величине r от (x,y) образуют ромбовидную структуру с центром в (x,y) (центральная точка). Например, пиксели с расстоянием $D_4 < 2$ от (x,y) образуют следующие контуры с равными от центра расстояниями

```

    2
  2 1 2
2 1 0 1 2
  2 1 2
    2
  
```

Расстояние D_8 , называемое также шахматным между p и q определяется по формуле:

$$D_8(p,q) = \max \{ |x-s|, |y-t| \}$$

При этом пиксели с расстоянием, меньшим или равным некоторой величине r образуют квадрат с центром в (x,y) . Например, пиксели с расстоянием $D_8 \leq 2$ от центральной точки образуют следующие равноудаленные от центра контуры:

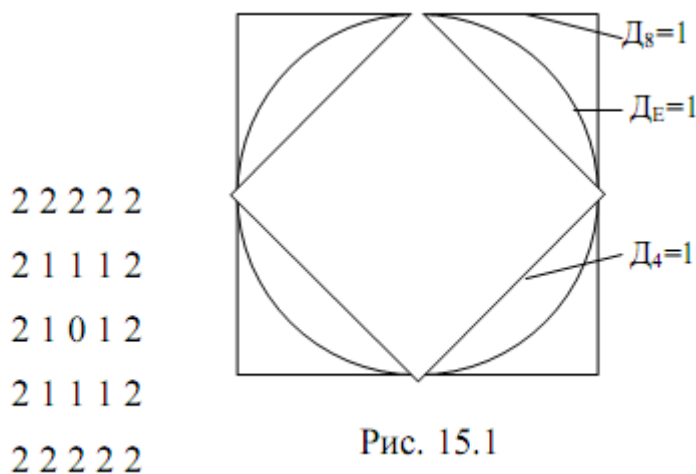


Рис. 15.1

Геометрические места точек, удаленных на единичное расстояние от начала координат, построенные для различных метрик будут иметь вид:

15.1

40. Классификация случайных процессов.

Случайный процесс $\{x_k(t)\}$ ($-\infty < t < \infty$), иначе временной ряд или стохастический процесс, обозначается символом $\{\cdot\}$ - это совокупность действительных (или комплексных) функций, которую можно охарактеризовать её вероятностной структурой. Каждая отдельная функция $x_k(t)$, где t - переменная, а k фиксировано, называется выборочной функцией. С практической точки зрения, выборочную функцию (или некоторый отрезок выборочной функции конечной длины) можно считать наблюдаемым результатом отдельного эксперимента. Возможное число экспериментов определяется выборочным пространством индексов k , причём последнее может быть счётным или несчётным (рис. 9.1).

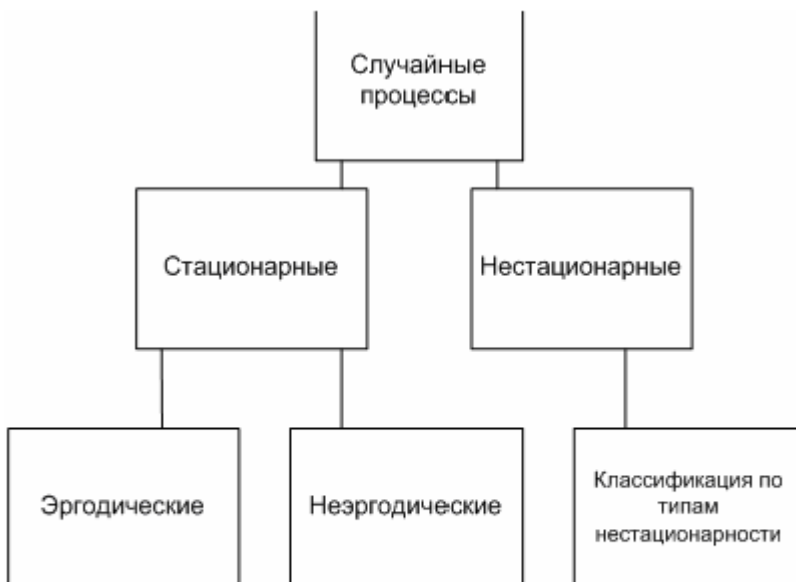


Рис. 9.2. Классификация случайных процессов.

Однако в большинстве случаев, характеристики стационарного случайного процесса (ССП) можно вычислить, усредняя по времени в пределах отдельных выборочных функций, входящих в ансамбль.

В этом случае среднее значение $\mu_x(k)$ и ковариационная функция

$R_{xx}(\tau, k)$, вычисленные по k -ой реализации будут равны:

$$\mu_x(k) = \lim_{T \rightarrow \infty} \frac{1}{T} \int_0^T x_k(t) dt \quad R_{xx}(\tau, k) = \lim_{T \rightarrow \infty} \frac{1}{T} \int_0^T x_k(t) x_k(t + \tau) dt$$

Случайный процесс называется эргодическим, если он стационарен, а математическое ожидание и ковариационная функция, вычисленные при усреднении по ансамблю реализаций и по времени совпадают. Отметим, что свойством эргодичности могут обладать только стационарные процессы. К счастью, на практике ССП обычно оказываются эргодическими и именно по этой причине свойства ССП можно определить по одной наблюдаемой реализации.

К нестационарным процессам относятся все случайные процессы, не удовлетворяющие условию стационарности, то есть такие процессы, основные характеристики которых зависят от времени.

Идеальный белый шум – это стационарный случайный сигнал, обладающий фундаментальным свойством статистической независимости любых двух его отсчётов. Кроме того, он имеет нулевое среднее значение, а автокорреляционная функция такого сигнала равна

$$R_{xx}(\tau) = a\delta(\tau)$$

где $\delta(\tau)$ - дельта-функция Дирака, равная нулю всюду, кроме точки $\tau = 0$, и интеграл от неё равен единице по любому интервалу, содержащему начало координат; коэффициент a - скалярная величина, называемая уровнем мощности или интенсивностью белого шума.

41. Классификация нейросетевых систем

Для лучшего понимания данного вопроса приведем одну из возможных классификаций НС в зависимости от различных характеристик:

1. По типу входной информации:
 - аналоговые НС (используют информацию в форме Действительных чисел);
 - двоичные НС (оперируют с информацией, представленной в Двоичном виде);
2. По характеру обучения:
 - с учителем (известно выходное пространство решений НС);
 - без учителя (НС формирует выходное пространство решений Только на основе входных воздействий – самоорганизующиеся сети);
3. По характеру настройки синапсов:
 - сети с фиксированными связями (весовые коэффициенты НС Выбираются сразу, исходя из условия задачи);
 - сети с динамическими связями (в процессе обучения Происходит настройка синаптических связей);
4. По методу обучения
 - НС с алгоритмом обратного распространения ошибки;
 - НС с конкурентным обучением;
 - НС, использующие правило Хебба;
 - НС с гибридным обучением, в которых используются Различные алгоритмы обучения;
5. По характеру связей:
 - НС с прямыми связями;
 - НС с обратным распространением информации;
6. По архитектуре и обучению:
 - персептронные сети с прямыми связями;
 - самоорганизующиеся НС (НС Кохонена, НС адаптивного резонанса, рециркуляционные НС);
 - НС с обратными связями (НС Хопфилда, НС Хэмминга, Двухнаправленная ассоциативная память, рекуррентные НС);
 - гибридные НС (НС встречного распространения, НС с радиально-базисной функцией активации).

42. Отображение по методу наименьших квадратов

При обосновании классификаторов, работающих по критерию наименьшего расстояния, предполагалось, что классы образов в пространстве признаков группируются вокруг соответствующих им средних \bar{Z}_i ($i = 1, 2, \dots, K$). Однако, возможен и другой подход. При этом классификатор должен в первую очередь отображать образы в пространство решений, в котором образы, принадлежащие C_i обязательно группируются вокруг заранее выбранной точки V_i . 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_i , если он отображён ближе к точке V_i . Введём отображение по методу наименьших квадратов, на котором основываются классификаторы с минимальным среднеквадратичным расстоянием. Рассмотрим множество M -мерных образов Z_{ij} , $j = 1, 2, \dots, N_i$, которые должны отображаться в определённую точку в K -мерном пространстве, обозначаемую $V_i = [v_1, v_2, \dots, v_k]$.

Найдём преобразование A , которое отображает $\{Z_{ij}\}$ в точку V_i , таким образом, чтобы общая среднеквадратичная ошибка, вызываемая отображением, была минимальной.

Обозначим результат отображения образа Z_{ij} через L_{ij} . Тогда соответствующий вектор ошибки будет равен

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

Из выражения следует, что общая среднеквадратичная ошибка при отображении Z_{ij} в V_i определяется как

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

Подставим одно в другое:

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

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

$$\frac{1}{N_i} \sum_{j=1}^{N_i} \nabla_A \left\{ Z_{ij}' A' A Z_{ij} \right\} - \frac{2}{N_i} \sum_{j=1}^{N_i} \nabla_A \left\{ Z_{ij}' A' V_i \right\} + \frac{1}{N_i} \sum_{j=1}^{N_i} \nabla_A \left\{ \|V_i\|^2 \right\} = 0$$

Поскольку

$$\nabla_A \left\{ Z_{ij}' A' A Z_{ij} \right\} = 2A \left(Z_{ij} Z_{ij}' \right),$$

$$\nabla_A \left\{ Z_{ij}' A' V_i \right\} = V_i Z_{ij}',$$

$$\nabla_A \left\{ \|V_i\|^2 \right\} = 0$$

Подставим это в предыдущую формулу:

$$A \left[\frac{2}{N_i} \sum_{j=1}^{N_i} \left(Z_{ij} Z_{ij}' \right) \right] = \frac{2}{N_i} \sum_{j=1}^{N_i} V_i Z_{ij}',$$

что позволяет определить A как

$$A = \left[\sum_{j=1}^{N_i} V_i Z_{ij}' \right] \left[\sum_{j=1}^{N_i} \left(Z_{ij} Z_{ij}' \right) \right]^{-1}$$

Рассмотрим пример. Пусть множество $\{Z_{ij}\}$ имеет вид

$$Z_{i1} = \begin{bmatrix} 5 \\ 5 \end{bmatrix}; Z_{i2} = \begin{bmatrix} 5 \\ 6 \end{bmatrix}; Z_{i3} = \begin{bmatrix} 5 \\ 7 \end{bmatrix}; Z_{i4} = \begin{bmatrix} 6 \\ 6 \end{bmatrix}; Z_{i5} = \begin{bmatrix} 6 \\ 7 \end{bmatrix}$$

что соответствует $N_i = 5$. Пусть $V_i = \begin{bmatrix} 1 \\ 1 \end{bmatrix}$ тогда

$$\begin{aligned} \sum_{j=1}^5 V_i Z_{ij}' &= \begin{bmatrix} 1 \\ 1 \end{bmatrix} [5 \ 5] + \begin{bmatrix} 1 \\ 1 \end{bmatrix} [5 \ 6] + \begin{bmatrix} 1 \\ 1 \end{bmatrix} [5 \ 7] + \begin{bmatrix} 1 \\ 1 \end{bmatrix} [6 \ 6] + \begin{bmatrix} 1 \\ 1 \end{bmatrix} [6 \ 7] = \\ &= \begin{bmatrix} 5 & 5 \\ 5 & 5 \end{bmatrix} + \begin{bmatrix} 5 & 6 \\ 5 & 6 \end{bmatrix} + \begin{bmatrix} 5 & 7 \\ 5 & 7 \end{bmatrix} + \begin{bmatrix} 6 & 6 \\ 6 & 6 \end{bmatrix} + \begin{bmatrix} 6 & 7 \\ 6 & 7 \end{bmatrix} = \begin{bmatrix} 27 & 31 \\ 27 & 31 \end{bmatrix} \end{aligned}$$

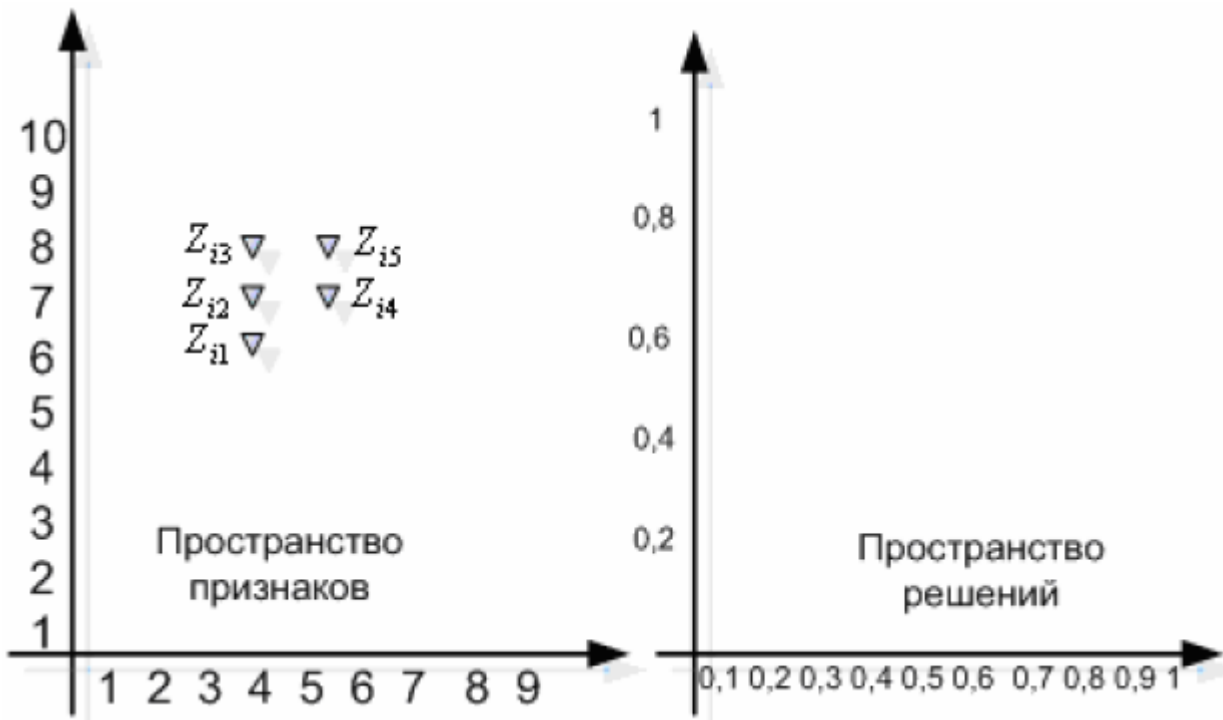
$$\sum_{j=1}^5 Z_{ij} Z_{ij}' = \begin{bmatrix} 25 & 25 \\ 25 & 25 \end{bmatrix} + \begin{bmatrix} 25 & 30 \\ 30 & 36 \end{bmatrix} + \begin{bmatrix} 25 & 35 \\ 35 & 49 \end{bmatrix} + \begin{bmatrix} 36 & 36 \\ 36 & 36 \end{bmatrix} + \begin{bmatrix} 36 & 42 \\ 42 & 49 \end{bmatrix} = \begin{bmatrix} 147 & 168 \\ 168 & 195 \end{bmatrix}$$

Тогда:
$$A = \begin{Bmatrix} 0,129 & 0,0475 \\ 0,129 & 0,0475 \end{Bmatrix}$$

Далее вычислим:

$$L_{ij} = AZ_{ij}$$

$$L_{i1} = \begin{bmatrix} 0,883 \\ 0,883 \end{bmatrix}; L_{i2} = \begin{bmatrix} 0,930 \\ 0,930 \end{bmatrix}; L_{i3} = \begin{bmatrix} 0,978 \\ 0,978 \end{bmatrix}; L_{i4} = \begin{bmatrix} 1,06 \\ 1,06 \end{bmatrix}; L_{i5} = \begin{bmatrix} 1,11 \\ 1,11 \end{bmatrix}$$



Понятие расширенного пространства признаков необходимо при разработке классификатора с минимальным среднеквадратичным расстоянием. Такое понятие вытекает из определения дискриминантной функции

$$\xi(Z) = w_1 Z_1 + w_2 Z_2 + \dots + w_d Z_d - \theta$$

В матричном виде это соотношение можно записать как $\xi(Z) = W' \hat{Z}$,

где $W' = [w_1 w_2, \dots, w_d \theta]$; $\hat{Z}' = [Z_1 Z_2, \dots, Z_d - 1] = [Z' - 1]$

Из последнего соотношения следует, что Z (с крышкой) можно получить изданного образа Z присписыванием к нему дополнительной координаты, равной -1. Пространство, состоящее из $(d + 1)$ -мерных образов Z , называется расширенным пространством признаков.

44. Классификатор для распознавания 3-х и к классов образов по критерию наименьшего среднеквадратического расстояния

Классификатор для распознавания трёх классов, работающий по критерию наименьшего среднеквадратического расстояния можно представить иллюстративно следующим образом:

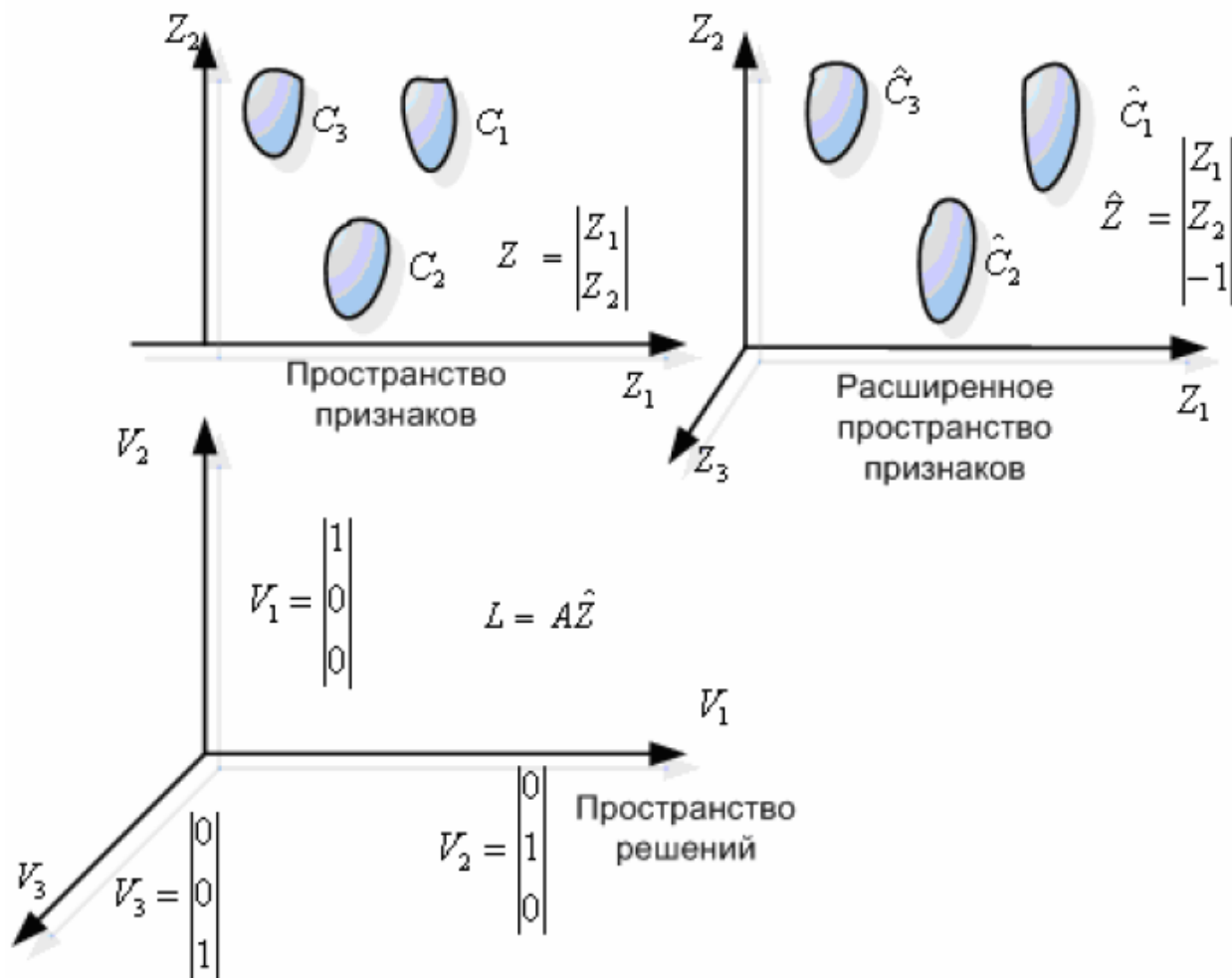


Рис. 26.1

Определяем преобразование A , отображающее Z (с крышкой) $\in C_k$; $k=1,2,3$ в V_k в смысле наименьших квадратов. Выберем в качестве вершины трёхединичных векторов, как показано на рис. Обозначим априорные вероятности классов C_k через P_k . Преобразование A можно непосредственно найти из соотношения

$$A \left[\frac{P_1}{N_1} \sum_{j=1}^{N_1} \hat{Z}_{1j} \hat{Z}'_{1j} + \frac{P_2}{N_2} \sum_{j=1}^{N_2} \hat{Z}_{2j} \hat{Z}'_{2j} + \frac{P_3}{N_3} \sum_{j=1}^{N_3} \hat{Z}_{3j} \hat{Z}'_{3j} \right] =$$

$$= \frac{P_1}{N_1} \sum_{j=1}^{N_1} V_1 \hat{Z}'_{1j} + \frac{P_2}{N_2} \sum_{j=1}^{N_2} V_2 \hat{Z}'_{2j} + \frac{P_3}{N_3} \sum_{j=1}^{N_3} V_3 \hat{Z}'_{3j}$$

Решая это уравнение относительно A , получим

$$A = S_{v\hat{z}} S^{-1}_{\hat{z}\hat{z}};$$

$$\text{где } S_{v\hat{z}} = \sum_{i=1}^3 \sum_{j=1}^{N_i} \frac{P_i}{N_i} (V_i \hat{Z}'_{ij}) = E\{V\hat{Z}\}$$

$$S_{\hat{z}\hat{z}} = \sum_{i=1}^3 \sum_{j=1}^{N_i} \frac{P_i}{N_i} (\hat{Z}_{ij} \hat{Z}'_{ij}) = E\{\hat{Z}\hat{Z}'\}$$

Очевидно, что S_{vz} (скрышкой) и S_{zz} (обе с крышкой) являются матрицами взаимной корреляции и автокорреляции соответственно. Следует отметить, что матрица A имеет размерность $[3 \times (d + 1)]$. Для классификации образа Z' (скрышкой) $[Z_1, Z_2, Z_{d-1}]$ классификатор вычисляет в первую очередь $L = AZ$ (скрышкой), а затем отображает Z (с кр.) в пространство решений. При этом применяется следующее решающее правило по критерию минимума расстояния: если L наиболее близко к V_{i_0} , то Z классифицируется как принадлежащий C_{i_0} . Расстояния, которые вычисляет классификатор определяются как

$$D_i^2 = \|L - V_i\|^2, i = \overline{1,3}; \text{ т.е.}$$

$$D_i^2 = (L - V_i)' (L - V_i) = \|L\|^2 - 2V_i' L + \|V_i\|^2$$

Так как $|V_i|^2 = 1$, то D_i^2 будет минимально, когда $V_i' L$ максимально. Поэтому вместо D_i^2 в последнем соотношении достаточно, чтобы классификатор вычислял

$$d_i = V_i' L \quad (26.3)$$

Подставляя $V_1' = [1 \ 0 \ 0]$, $V_2' = [0 \ 1 \ 0]$, $V_3' = [0 \ 0 \ 1]$ и $L = A\hat{Z}$ в (26.3)

$$\begin{bmatrix} d_1 \\ d_2 \\ d_3 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} A\hat{Z} = A\hat{Z}$$

Матрица преобразования обозначается как

$$A = \begin{bmatrix} a_{11} & a_{12} & \dots & a_{1d} & \theta_1 \\ a_{21} & a_{22} & \dots & a_{2d} & \theta_2 \\ a_{31} & a_{32} & \dots & a_{3d} & \theta_3 \end{bmatrix},$$

где d_i представляют собой следующие дискриминантные функции, которые определяют классификатор

$$\xi(Z) = d_i = a_{i1}Z_1 + a_{i2}Z_2 + \dots + a_{id}Z_d - \theta_i$$

Из последнего соотношения следует, что классификатор полностью определяется матрицей преобразования A , которая получается из множества обучающих образов. Для классификации данного образа классификатор вычисляет три числа

$$\xi_1(Z), \xi_2(Z) \text{ и } \xi_3(Z)$$

Если $\max\{\xi_i(Z)\} = \xi_{i_0}(Z)$, то этот образ приписывается C_{i_0}

Задача трёх классов может быть обобщена для случая K классов. Расширенные образы, принадлежащие C_i отображаются в вершине единичного K -вектора V_i

$$V_i = [0 \dots 0 \ 1 \ 0 \dots 0]$$

где нулевой элемент «1» находится в i -ой позиции V_i , $i = \overline{1, K}$. Вместо матрицы преобразования A размерностью $3 \times (d+1)$ получаем матрицу размерностью $K \times (d+1)$

$$A = S_{v\hat{x}} S_{\hat{x}\hat{x}}, \text{ где}$$

$$S_{v\hat{x}} = \sum_{i=1}^K \sum_{j=1}^{N_i} P_i / N_i (V_i \hat{Z}'_{ij}) = E(V\hat{Z}')$$

$$S_{\hat{x}\hat{x}} = \sum_{i=1}^K \sum_{j=1}^{N_i} P_i / N_i (\hat{Z}_{ij} \hat{Z}'_{ij}) = E(\hat{Z}\hat{Z}')$$

Таким образом, дискриминантные функции, определяющие классификатор, имеют вид

$$\xi_i(z) = a_{i1}z_1 + a_{i2}z_2 + \dots + a_{id}z_d - \theta_i, i = \overline{1, K}$$

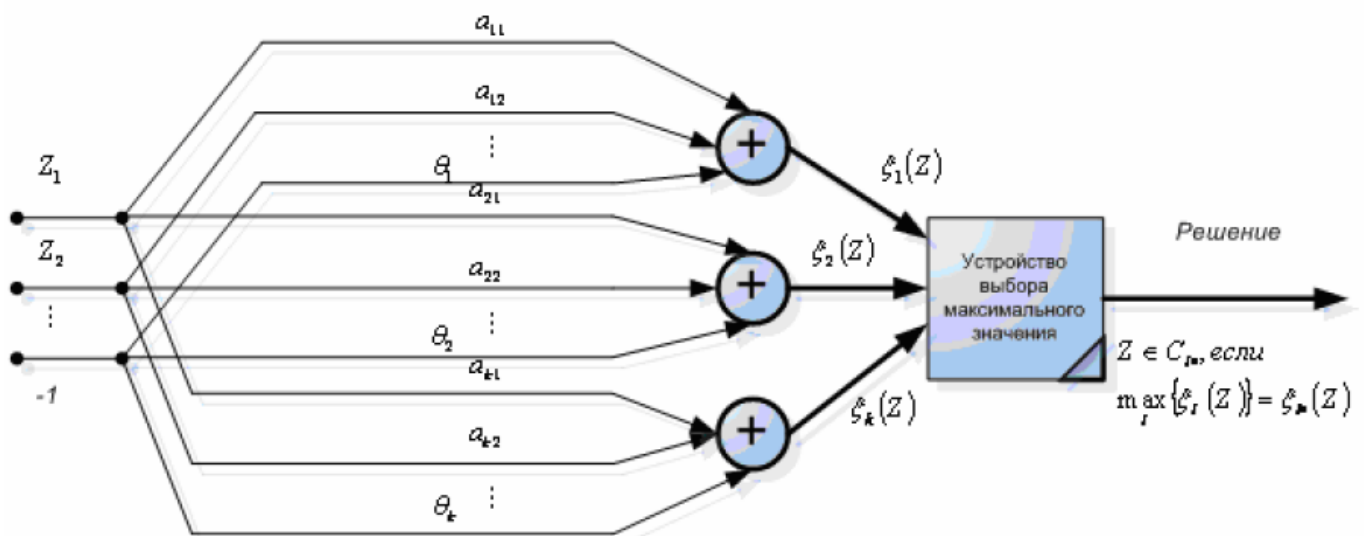


Рис. 27.1 Структура классификатора k классов

При обосновании классификаторов по минимуму расстояния мы ограничивались рассмотрением только линейных классификаторов, т.е. классификаторов, которые обладают линейными разделяющими границами в пространстве признаков. Соответствующие

щую процедуру обучения нетруднополучить для случая квадратичных разделяющих границ, но при этомреализация классификатора усложняется. Квадратичный классификаторопределяется дискриминантными функциями вида

$$\xi_i(z) = \sum_{j=1}^d w_{ij} z_j^2 + \sum_{j=1}^{d-1} \sum_{k=j+1}^d w_{jk} z_j z_k + \sum_{j=1}^d w_j z_j - \theta_i, i = \overline{1, K} \quad (27.1)$$

Изпоследнего соотношения следует, что квадратичнаядискриминантная функция $\xi_i(z)$ имеет $[(d+1)/(d+2)]/2$ весовыхкоэффициентов или параметров, а именно:

d параметров , соответствующих коэффициентам при $z_j^2 - w_{jj}$

d параметров , соответствующих коэффициентам при $z_j - w_j$

$d(d-1)/2$ параметров, соответствующих коэффициентам при

$z_j z_k - w_{jk}, j \neq k$

порог- θ_i

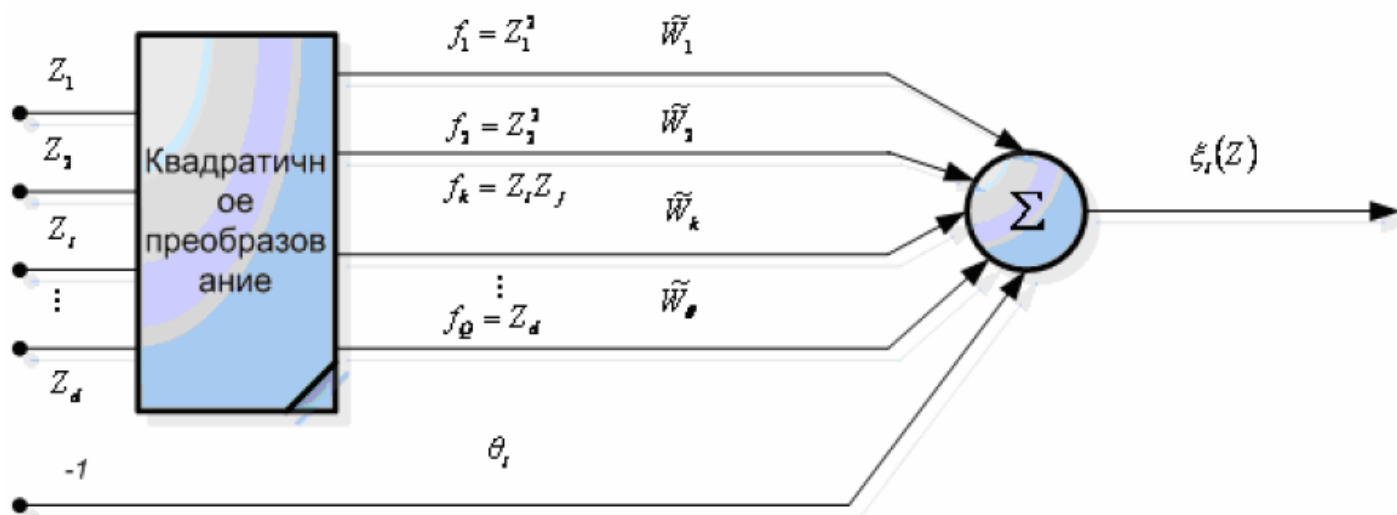
Приведённыевышепараметры определяются в процессе обучения.Для пояснения реализации классификатора, связанного с вычислением(27.1) определим Q -мерный вектор G , координаты которого f_1, f_2, \dots, f_Q являются функциями $z_i, i = 1, d$ (сверху надчер-

кивание). Первые d координат вектора G имеют вид $z_1^2, z_2^2, \dots, z_d^2$, следующие $[d(d$

$-1)]/2$ координаты всем парам $z_1 z_2, z_1 z_3, \dots, z_{d-1} z_d$

последние a координатпредставляютсобой z_1, z_2, \dots, z_d

Тогда реализация классификаторабудетиметьвид



$$\xi_i(z) = \tilde{w}_1 f_1 + \tilde{w}_2 f_2 + \dots + \tilde{w}_Q f_Q - \Theta_i$$

46 Алгоритм быстрого преобразования Уолша-Адамара.

Быстрое преобразование Уолша можно получить с помощью технологии разбиения матриц. Рассмотрим вывод алгоритма для $N=8$. При $N=8$ в матричном виде преобразование Уолша можно записать как

$$C_x(3) = \frac{1}{8} H(3) X(3).$$

Используя соотношение $H(k) = \begin{bmatrix} H(k-1) & H(k-1) \\ H(k-1) & -H(k-1) \end{bmatrix}$, $H(3)$ можно выразить

через $H(2)$, что приводит к

$$\begin{bmatrix} C_x(0) \\ C_x(1) \\ C_x(2) \\ C_x(3) \\ C_x(4) \\ C_x(5) \\ C_x(6) \\ C_x(7) \end{bmatrix} = \frac{1}{8} \begin{bmatrix} H(2) & H(2) \\ H(2) & -H(2) \end{bmatrix} \begin{bmatrix} x(0) \\ x(1) \\ x(2) \\ x(3) \\ x(4) \\ x(5) \\ x(6) \\ x(7) \end{bmatrix}.$$

Из разбиения матрицы следует, что

$$\begin{bmatrix} C_x(0) \\ C_x(1) \\ C_x(2) \\ C_x(3) \end{bmatrix} = \frac{1}{8} H(2) \begin{bmatrix} x_1(0) \\ x_1(1) \\ x_1(2) \\ x_1(3) \end{bmatrix} \quad \text{и} \quad \begin{bmatrix} C_x(4) \\ C_x(5) \\ C_x(6) \\ C_x(7) \end{bmatrix} = \frac{1}{8} H(2) \begin{bmatrix} x_1(4) \\ x_1(5) \\ x_1(6) \\ x_1(7) \end{bmatrix},$$

где $x_1(l) = x(l) + x(4+l); l = \overline{0,3}$;

$x_1(l) = x(l-4) - x(l); l = \overline{4,7}$.

Подставляя вместо $H_2 = \begin{bmatrix} H(1) & H(1) \\ H(1) & -H(1) \end{bmatrix}$, будем иметь

$$\begin{bmatrix} C_x(0) \\ C_x(1) \\ C_x(2) \\ C_x(3) \end{bmatrix} = \frac{1}{8} \begin{bmatrix} H(1) & H(1) \\ H(1) & -H(1) \end{bmatrix} \begin{bmatrix} x_1(0) \\ x_1(1) \\ x_1(2) \\ x_1(3) \end{bmatrix} \quad \text{и}$$

$$\begin{bmatrix} C_x(4) \\ C_x(5) \\ C_x(6) \\ C_x(7) \end{bmatrix} = \frac{1}{8} \begin{bmatrix} H(1) & H(1) \\ H(1) & -H(1) \end{bmatrix} \begin{bmatrix} x_1(4) \\ x_1(5) \\ x_1(6) \\ x_1(7) \end{bmatrix}.$$

Тогда получим

$$\begin{bmatrix} C_x(0) \\ C_x(1) \end{bmatrix} = \frac{1}{8} H(1) \begin{bmatrix} x_1(0) + x_1(2) \\ x_1(1) + x_1(3) \end{bmatrix} = \frac{1}{8} H(1) \begin{bmatrix} x_2(0) \\ x_2(1) \end{bmatrix};$$

$$\begin{bmatrix} C_x(2) \\ C_x(3) \end{bmatrix} = \frac{1}{8} H(1) \begin{bmatrix} x_1(0) - x_1(2) \\ x_1(1) - x_1(3) \end{bmatrix} = \frac{1}{8} H(1) \begin{bmatrix} x_2(2) \\ x_2(3) \end{bmatrix};$$

$$\begin{bmatrix} C_x(4) \\ C_x(5) \end{bmatrix} = \frac{1}{8} H(1) \begin{bmatrix} x_1(4) + x_1(6) \\ x_1(5) + x_1(7) \end{bmatrix} = \frac{1}{8} H(1) \begin{bmatrix} x_2(4) \\ x_2(5) \end{bmatrix};$$

$$\begin{bmatrix} C_x(6) \\ C_x(7) \end{bmatrix} = \frac{1}{8} H(1) \begin{bmatrix} x_1(4) - x_1(6) \\ x_1(5) - x_1(7) \end{bmatrix} = \frac{1}{8} H(1) \begin{bmatrix} x_2(6) \\ x_2(7) \end{bmatrix}.$$

Так как $H(1) = \begin{bmatrix} H(0) & H(0) \\ H(0) & -H(0) \end{bmatrix} = \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}$, то окончательно получим

$$8C_x(0) = x_2(0) + x_2(1) = x_3(0);$$

$$8C_x(1) = x_2(0) - x_2(1) = x_3(1);$$

$$8C_x(2) = x_2(2) + x_2(3) = x_3(2);$$

$$8C_x(3) = x_2(2) - x_2(3) = x_3(3);$$

$$8C_x(4) = x_2(4) + x_2(5) = x_3(4);$$

$$8C_x(5) = x_2(4) - x_2(5) = x_3(5);$$

$$8C_x(6) = x_2(6) + x_2(7) = x_3(6);$$

$$8C_x(7) = x_2(6) - x_2(7) = x_3(7).$$

Для $N=2^n$:

1. Общее число итераций равно $n = \log_2 N$. Индекс r принимает значения $r=1, 2, \dots, n$.

2. В r итерации участвует 2^{r-1} групп по $N/2^{r-1}$ элементов. Половина элементов в каждой группе связана с операцией сложения, а другая половина – с операцией вычитания.

3. Общее число арифметических операций, необходимое для вычисления всех коэффициентов преобразования, равняется приблизительно $N \log_2 N$.

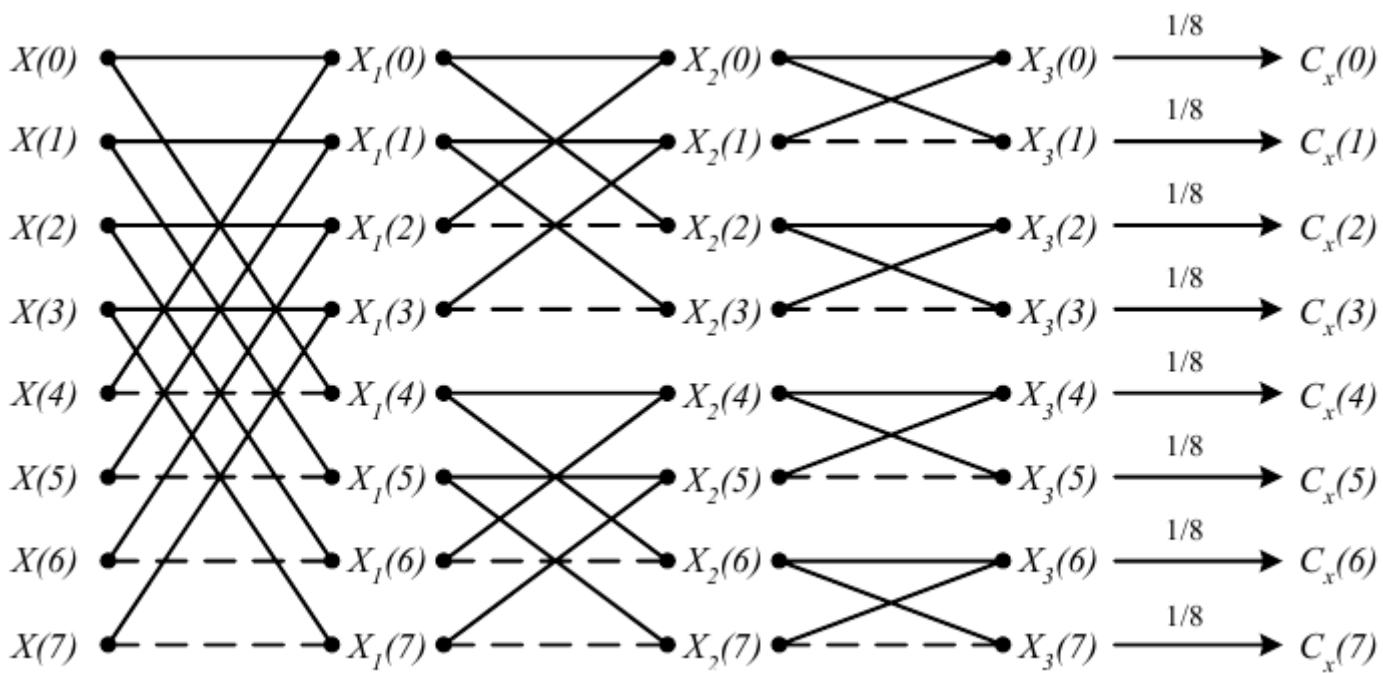


Рис. 6.4. Процедура быстрого алгоритма Уолша

47. Основные характеристики случайных процессов.

Случайный процесс $\{x_k(t)\}$ ($-\infty < t < \infty$), иначе временной ряд или стохастический процесс, обозначается символом $\{\cdot\}$ - это совокупность действительных (или комплекснозначных) функций, которую можно охарактеризовать её вероятностной структурой. Каждая отдельная функция $x_k(t)$, где t - переменная, а k фиксировано, называется выборочной функцией. С практической точки зрения, выборочную функцию (или некоторый отрезок выборочной функции конечной длины) можно считать наблюдаемым результатом отдельного эксперимента. Возможное число экспериментов определяется выборочным пространством индексов k , причём последнее может быть счётным или несчётным (рис. 9.1).

Рассмотрим два произвольных случайных процесса $\{x_k(t)\}$ и $\{y_k(t)\}$. В первую очередь, представляют интерес средние значения по ансамблю в произвольный фиксированный момент времени t , где $x_k(t)$ и $y_k(t)$ - случайные величины от аргумента k . Такие средние значения определяются:

$\mu_x(t) = E[x_k(t)]$ $\mu_y(t) = E[y_k(t)]$, (9.1) где E - символ математического ожидания.

Следующие важные статистические характеристики - это корреляционные функции, определённые для произвольных фиксированных моментов времени $t_1 = t$ и $t_2 = t + \tau$.

$$C_{xx}(t_1, t_2) = C_{xx}(t, t + \tau) = E\{[x_k(t) - \mu_x(t)][x_k(t + \tau) - \mu_x(t + \tau)]\}$$

$$C_{yy}(t_1, t_2) = C_{yy}(t, t + \tau) = E\{[y_k(t) - \mu_y(t)][y_k(t + \tau) - \mu_y(t + \tau)]\}$$

$$C_{xy}(t_1, t_2) = C_{xy}(t, t + \tau) = E\{[x_k(t) - \mu_x(t)][y_k(t + \tau) - \mu_y(t + \tau)]\}$$

В общем случае эти величины различны для различных комбинаций t_1 и t_2 .

При $\tau = 0$:

$$C_{xx}(t, t) = E[(x_k(t) - \mu_x(t))^2] = \sigma_x^2$$

$$C_{yy}(t, t) = E[(y_k(t) - \mu_y(t))^2] = \sigma_y^2$$

$$C_{xy}(t, t) = E\{[x_k(t) - \mu_x(t)][y_k(t) - \mu_y(t)]\}$$

Следовательно, корреляционные функции $C_{xx}(t, t)$ и $C_{yy}(t, t)$ совпадают с обычными дисперсиями процессов $\{x_k(t)\}$ и $\{y_k(t)\}$ в фиксированный момент времени, в то время как $C_{xy}(t, t)$ представляет собой корреляцию случайных величин $\{x_k(t)\}$ и $\{y_k(t)\}$.

По ансамблю реализаций можно определить и другие статистические характеристики, зависящие от трёх и более моментов времени.

Если средние значения $\mu_x(t)$ и $\mu_y(t)$ вместе с корреляционными функциями $C_{xx}(t, t + \tau)$, $C_{yy}(t, t + \tau)$, и $C_{xy}(t, t + \tau)$ принимают одинаковые значения при всех фиксированных значениях t , то случайные процессы $\{x_k(t)\}$ и $\{y_k(t)\}$ называются слабо стационарными. Если всевозможные распределения, связанные с $\{x_k(t)\}$ и $\{y_k(t)\}$, не зависят от выбора начала отсчёта времени, то такие процессы называются строго стационарными.

Для стационарных случайных процессов $\{x_k(t)\}$ и $\{y_k(t)\}$ средние значения постоянны и не зависят от t . То есть:

$$\mu_x = E[x_k(t)] = \int_{-\infty}^{\infty} xp(x)dx \qquad \mu_y = E[y_k(t)] = \int_{-\infty}^{\infty} yp(y)dy$$

Где $p(x)$, $p(y)$ - плотности вероятности случайных величин $x_k(t)$ и $y_k(t)$ соответственно.

В свою очередь, для произвольных фиксированных величин t и τ :

$$R_{xx}(\tau) = E[x_k(t)x_k(t+\tau)]$$

$$R_{yy}(\tau) = E[y_k(t)y_k(t+\tau)]$$

$$R_{xy}(\tau) = E[x_k(t)y_k(t+\tau)]$$

Величины $R_{xx}(\tau)$ и $R_{yy}(\tau)$ называются автокорреляционными или ковариационными функциями процессов $\{x_k(t)\}$ и $\{y_k(t)\}$ соответственно, а $R_{xy}(\tau)$ называется взаимной ковариационной функцией процессов $\{x_k(t)\}$ и $\{y_k(t)\}$.

Для того, чтобы $R_{xx}(\tau)$ была ковариационной функцией слабо стационарного случайного процесса $\{x_k(t)\}$, необходимо и достаточно, чтобы $R_{xx}(\tau) = R_{xx}(-\tau)$ и $R_{xx}(\tau)$ была положительно определённой функцией.

В терминах плотностей вероятности:

$$R_{xx}(\tau) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} x_1 x_2 p(x_1, x_2) dx_1 dx_2$$

$$R_{yy}(\tau) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} y_1 y_2 p(y_1, y_2) dy_1 dy_2$$

$$R_{xy}(\tau) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} x_1 y_2 p(x_1, y_2) dx_1 dy_2$$

При произвольных μ_x и μ_y ковариационные функции связаны с корреляционными с помощью соотношений:

$$R_{xx}(\tau) = C_{xx}(\tau) - \mu_x^2$$

$$R_{yy}(\tau) = C_{yy}(\tau) - \mu_y^2$$

$$R_{xy}(\tau) = C_{xy}(\tau) - \mu_x \mu_y$$

Следовательно, корреляционные функции совпадают с ковариационными, если средние значения равны нулю.

В свою очередь, два стационарных процесса не коррелированы, если $C_{xy}(\tau) = 0$ для всех τ . Следовательно, если $R_{xy}(\tau) = 0$, то два процесса не коррелированы только тогда, когда хотя бы одно из средних $\mu_x(t)$ или $\mu_y(t)$ равно нулю.

Из предположения стационарности следует, что ковариационные функции $R_{xx}(\tau)$ и $R_{yy}(\tau)$ - чётные функции от τ , т.е.

$$R_{xx}(-\tau) = R_{xx}(\tau) \qquad R_{yy}(-\tau) = R_{yy}(\tau)$$

В то же время взаимная ковариационная функция удовлетворяет соотношению:

$$R_{xy}(-\tau) = R_{yx}(\tau)$$

Доказательство:

По определению: $R_{xy}(-\tau) = E[x(t)y(t-\tau)]$, где зависимость от t опущена для простоты обозначений.

Поскольку результат не зависит от сдвига начала отсчёта, то t можно заменить на $t+\tau$ до взятия математического ожидания.

Поэтому:

$$R_{xy}(-\tau) = E[x(t+\tau)y(t+\tau-\tau)] = E[x(t+\tau)y(t)] = R_{yx}(\tau)$$

Что и требовалось доказать.

При $x = y$ справедливо:

$$\left. \begin{aligned} R_{xx}(-\tau) &= R_{xx}(\tau) \\ R_{yy}(-\tau) &= R_{yy}(\tau) \end{aligned} \right\}$$

Т.е. эти соотношения – частный случай формулы (9.9). Корреляционные свойства стационарных случайных процессов $\{x_k(t)\}$ и $\{y_k(t)\}$ описываются четырьмя функциями $R_{xx}(\tau)$, $R_{yy}(\tau)$, $R_{xy}(\tau)$ и $R_{yx}(\tau)$, которые целесообразно вычислять только для значений $\tau \geq 0$, поскольку из соотношений (9.9) и (9.10) легко получить их значения для $\tau < 0$.

В случае ансамбля выборочных функций среднее значение (математическое ожидание) и ковариационная функция определяются как:

$$\mu_x(t_1) = \lim_{N \rightarrow \infty} \frac{1}{N} \sum_{k=1}^N x_k(t_1)$$

$$R_{xx}(t_1, t_1 + \tau) = \lim_{N \rightarrow \infty} \frac{1}{N} \sum_{k=1}^N x_k(t_1)x_k(t_1 + \tau)$$

То есть среднее значение (первый момент) случайного процесса в момент t_1 можно вычислить, выбрав мгновенные значения всех выборочных функций ансамбля в момент времени t_1 , сложив эти значения и разделив на число выборочных функций. Аналогично ковариационная функция вычисляется путём усреднения по ансамблю произведений мгновенных значений в моменты времени t_1 и $t_1 + \tau$.

Однако в большинстве случаев, характеристики стационарного случайного процесса (ССП) можно вычислить, усредняя по времени в пределах отдельных выборочных функций, входящих в ансамбль. В этом случае среднее значение $\mu_x(k)$ и ковариационная функция $R_{xx}(\tau, k)$, вычисленные по k -ой реализации будут равны:

$$\mu_x(k) = \lim_{T \rightarrow \infty} \frac{1}{T} \int_0^T x_k(t) dt$$

$$R_{xx}(\tau, k) = \lim_{T \rightarrow \infty} \frac{1}{T} \int_0^T x_k(t)x_k(t + \tau) dt$$

49. Классификатор для распознавания 3-х и к классов образов по критерию минимума расстояния.

Ранее рассматривались аспекты так называемой задачи двух классов . Распространим эти понятия на общий случай трёх классов: C_1, C_2, C_3 . Пусть обучающее множество записывается как

$$C_1 : Z_{11} = \begin{bmatrix} 0 \\ 3 \end{bmatrix}; Z_{12} = \begin{bmatrix} -1 \\ 3 \end{bmatrix}; Z_{13} = \begin{bmatrix} -2 \\ 3 \end{bmatrix}; Z_{14} = \begin{bmatrix} -3 \\ 3 \end{bmatrix}; Z_{15} = \begin{bmatrix} -4 \\ 3 \end{bmatrix}$$

$$C_2 : Z_{21} = \begin{bmatrix} 5 \\ 5 \end{bmatrix}; Z_{22} = \begin{bmatrix} 6 \\ 5 \end{bmatrix}; Z_{23} = \begin{bmatrix} 6 \\ 6 \end{bmatrix}; Z_{24} = \begin{bmatrix} 6 \\ 7 \end{bmatrix}; Z_{25} = \begin{bmatrix} 7 \\ 5 \end{bmatrix}$$

$$C_3 : Z_{31} = \begin{bmatrix} 6 \\ -1 \end{bmatrix}; Z_{32} = \begin{bmatrix} 7 \\ 0 \end{bmatrix}; Z_{33} = \begin{bmatrix} 8 \\ 1 \end{bmatrix}; Z_{34} = \begin{bmatrix} 9 \\ 1 \end{bmatrix}; Z_{35} = \begin{bmatrix} 10 \\ 1 \end{bmatrix}$$

В этом случае средние векторы образов равны

$$\bar{Z}_1 = \begin{bmatrix} -2 \\ 3 \end{bmatrix}; \bar{Z}_2 = \begin{bmatrix} 6 \\ 5,6 \end{bmatrix}; \bar{Z}_3 = \begin{bmatrix} 8 \\ 0,4 \end{bmatrix}$$

Классификатор, работающий по минимуму расстояния имеет следующее решающее правило : данный образ Z принадлежит C_i , если Z ближе всего к \bar{Z}_i , $i = 1, 2, 3$.

Пусть D_i обозначает расстояние образа Z от \bar{Z}_i . Тогда получим

$$D_i^2 = \|Z - \bar{Z}_i\|^2 = (Z - \bar{Z}_i)^T (Z - \bar{Z}_i)$$

Упрощение D_i^2 приводит к

$$D_i^2 = Z^T Z - Z^T \bar{Z}_i - \bar{Z}_i^T Z + \bar{Z}_i^T \bar{Z}_i = \|Z\|^2 - 2 \left(\bar{Z}_i^T Z - \frac{1}{2} \|\bar{Z}_i\|^2 \right)$$

Очевидно, что D_i^2 минимально, когда величина $\left\{ \bar{Z}_i^T Z - \frac{1}{2} \|\bar{Z}_i\|^2 \right\}$ максимальна. Поэтому вместо вычисления D_i^2 по формуле (24.1) проще потребовать, чтобы в классификаторе вычислялось значение $\left\{ \bar{Z}_i^T Z - \frac{1}{2} \|\bar{Z}_i\|^2 \right\}$.

Классификатор в этом случае описывается дискриминантными функциями.

$$\xi_i(Z) = \left\{ \bar{Z}_i^T Z - \frac{1}{2} \|\bar{Z}_i\|^2 \right\} \quad i = 1, 2, 3.$$

Подстановка численных значений \bar{Z}_i и $\|\bar{Z}_i\|^2$ приводит к

$$\xi_1(Z) = -2Z_1 + 3Z_2 - 6,5$$

$$\xi_2(Z) = 6Z_1 + 5,6Z_2 - 33,68$$

$$\xi_3(Z) = 8Z_1 + 0,4Z_2 - 32,08$$

Таким образом, классификатор вычисляет три числа: $\xi_1(Z)$, $\xi_2(Z)$ и $\xi_3(Z)$, а затем сравнивает их. Классификатор относит Z к классу C_1 , если $\xi_1(Z)$ максимально, к классу C_2 , если $\xi_2(Z)$ максимально и к классу C_3 , если $\xi_3(Z)$ максимально.

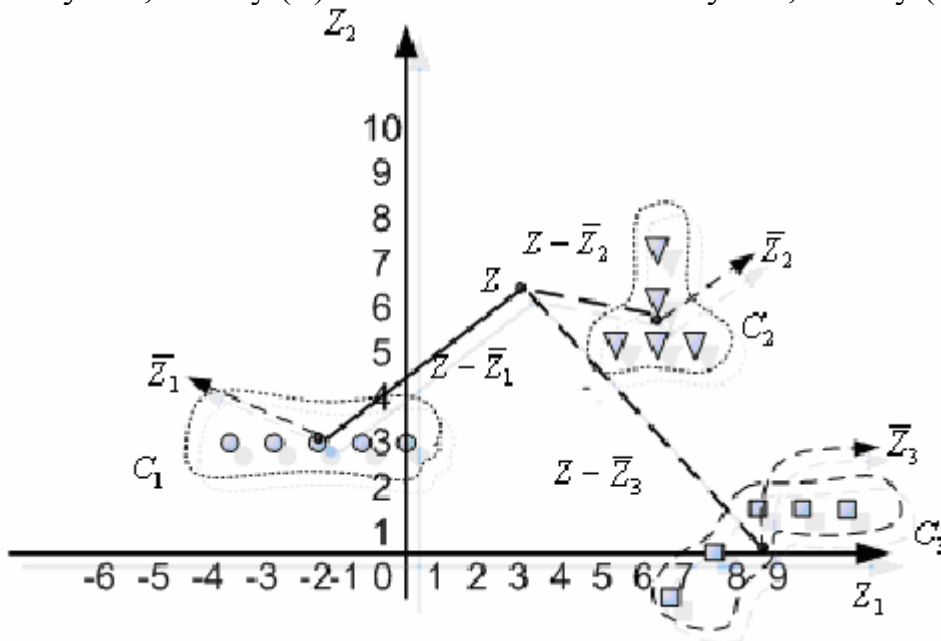


Рис. 24.1 Двумерное пространство признаков связанное с C_1, C_2, C_3 .

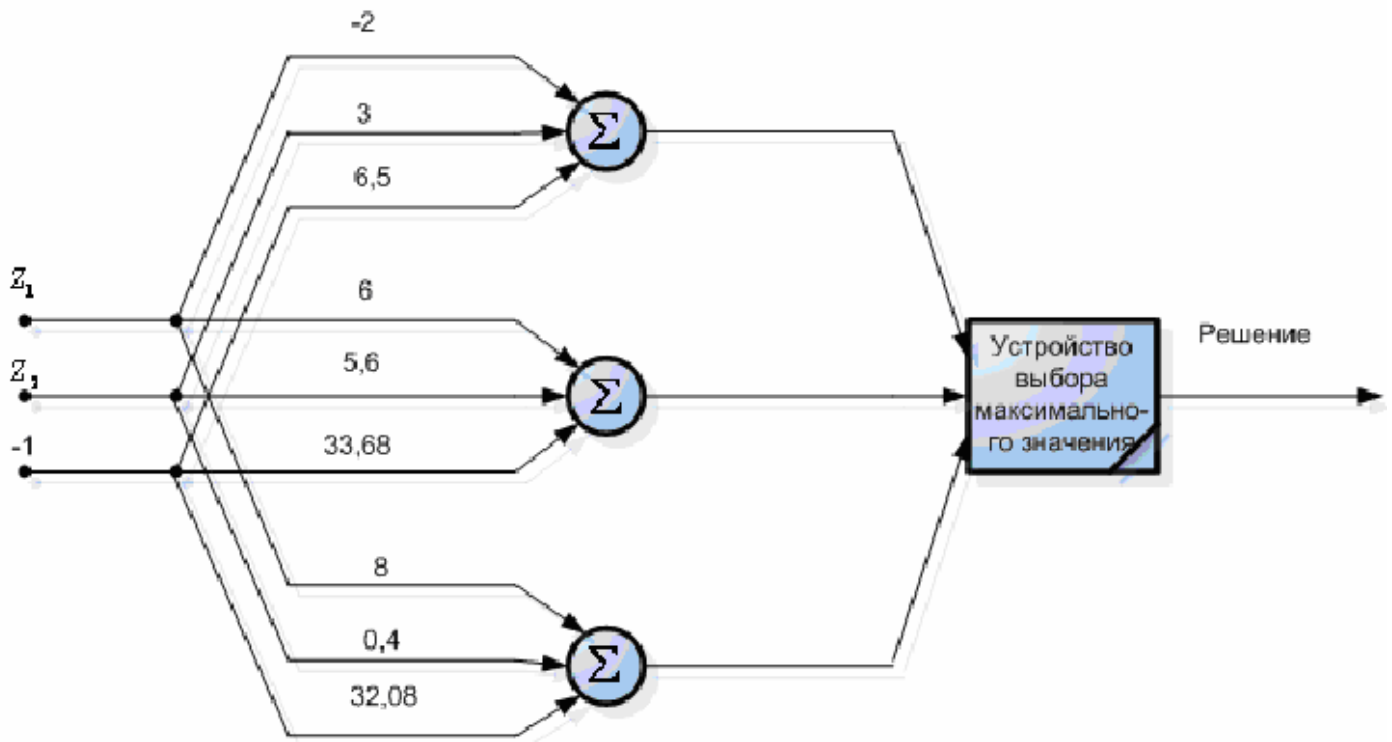


Рис. 24.2 Классификатор для трёх классов образов, работающий по критерию минимального расстояния.

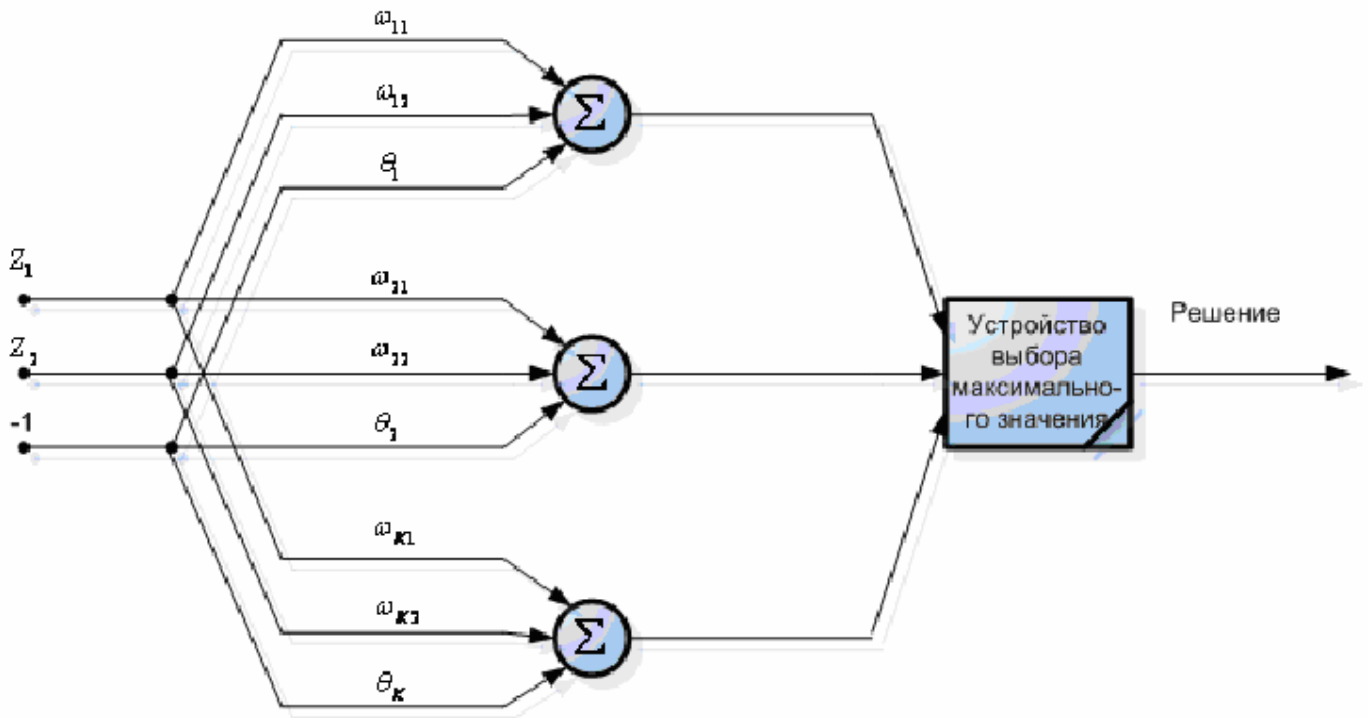


Рис. 24.3 Классификатор для K классов образов, работающий по критерию минимального расстояния.

50. Медианная фильтрация.

Фильтры, основанные на порядковых статистиках, относятся к классу нелинейных пространственных фильтров. Отклик фильтра определяется предварительным упорядочиванием (ранжированием), значений пикселей, покрываемых маской фильтра и последующим выбором значения, находящегося на определенной позиции упорядоченной последовательности (т. е. имеющего определенный ранг). При этом фильтрация сводится к замещению исходного значения пикселя (в центре маски) на полученное значение отклика фильтра. Из указанного класса наиболее известен медианный фильтр, который заменяет значение пикселя на значение медианы распределения яркости всех пикселей в окрестности, включая исходный. Медианные фильтры особенно эффективны при фильтрации импульсных шумов, иногда называемых шумами «соль и перец», которые выглядят как наложение на изображение случайных черных и белых точек.

Медиана набора чисел есть такое число ξ , что половина чисел из набора меньше или равны ξ , а другая половина – больше и равны ξ . Чтобы выполнить медианную фильтрацию для элемента изображения, необходимо сначала упорядочить по возрастанию значения пикселей внутри окрестности, затем найти значение медианы и наконец присвоить полученное значение обрабатываемому элементу. Так для окрестности размером 3×3 медианой будет пятое значение по величине, для окрестности 5×5 – тринадцатое значение и так далее.

Аналитически действие медианного фильтра определяется как

$$\hat{f}(x, y) = \underset{s, t \in S_{xy}}{med} \{g(s, t)\}$$

Причем при вычислении медианы значение в самой точке (т. е. в центре окрестности) также учитывается.

Если несколько элементов в окрестности имеют одинаковые значения, то значения будут сгруппированы. Например, пусть в окрестности 3×3 имеют место следующие значения (10, 20, 20, 20, 15, 20, 20, 25, 100). После упорядочения они будут расположены следующим образом (10, 15, 20, 20, 20, 20, 20, 25, 100), а, следовательно, медианой будет значение 20.

Хотя медианный фильтр значительно более распространен в обработке изображений, чем остальные виды фильтров, тем не менее, он не является единственным. Медиана представляет собой 50-ый процентиль упорядоченного набора чисел.

Широкая популярность медианных фильтров обусловлена тем, что они прекрасно приспособлены для подавления импульсных шумов и при этом приводят к меньшему размыванию по сравнению с линейными сглаживающими фильтрами.

51. Основные свойства двумерного преобразования Фурье.

Сдвиг

Пара преобразований Фурье обладает следующими трансляционными (сдвиговыми) свойствами

$$f(x,y)e^{j2\pi(u_0 x/M+v_0 y/N)} \Leftrightarrow F(u-u_0, v-v_0)$$

$$f(x-x_0, y-y_0) \Leftrightarrow F(u, v)e^{-j2\pi(x_0 u/M+y_0 v/N)},$$

где двойная стрелка использована для обозначения того, что соответствующие функции образуют пару Фурье-преобразований.

При $u_0=M/2$, $v_0=N/2$ будем иметь

$$e^{j2\pi(u_0 x/M+v_0 y/N)} = e^{j\pi(x+y)} = (-1)^{x+y}$$

В этом случае будем иметь

$$f(x,y)(-1)^{x+y} \Leftrightarrow F(U-M/2, V-N/2)$$

и аналогично при $x_0=M/2$; $y_0=N/2$

$$f(x-M/2, y-N/2) \Leftrightarrow F(u, v)(-1)^{u+v}$$

Из определения преобразования Фурье следует, что

$$F[f_1(x,y)+f_2(x,y)] = F[f_1(x,y)] + F[f_2(x,y)] \text{ и}$$

$$F[f_1(x,y)*f_2(x,y)] \neq F[f_1(x,y)]*F[f_2(x,y)]$$

Иными словами, преобразование Фурье обладает свойством дистрибутивности по отношению к сложению, но не обладает этим свойством по отношению к умножению. Указанное относится и к обратному Фурье-преобразованию.

Аналогично, если a , b – две постоянные, то

$$f(ax, by) \Leftrightarrow aF(u, v)$$

$$f(ax, by) \Leftrightarrow \frac{1}{|ab|} F(u/a, v/b), \quad ab \neq 0$$

(изменение масштаба)

Поворот

Введем понятие координаты

$$x=r*\cos\theta, \quad y=r*\sin\theta$$

$$u=\omega*\cos\varphi$$

$$v=\omega*\sin\varphi$$

И обозначим через $f(r, \theta)$ и $F(\omega, \varphi)$ функции, рассматриваемые как функции полярных координат соответственно.

Тогда $f(r, \theta + \theta_0) \Leftrightarrow F(\omega, \varphi + \theta_0)$

Это выражение указывает на то, что поворот функции $f(x, y)$ на угол θ_0 приводит к повороту функции $F(u, v)$ на тот же угол.

Дискретное преобразование Фурье обладает следующими свойствами периодичности

$$F(\mathbf{u}, \mathbf{v}) = F(\mathbf{u} + \mathbf{M}, \mathbf{v}) = F(\mathbf{u}, \mathbf{v} + \mathbf{N}) = F(\mathbf{u} + \mathbf{M}, \mathbf{v} + \mathbf{N})$$

Обратное Фурье-преобразование также периодично:

$$f(\mathbf{x}, \mathbf{y}) = f(\mathbf{x} + \mathbf{M}, \mathbf{y}) = f(\mathbf{x}, \mathbf{y} + \mathbf{N}) = f(\mathbf{x} + \mathbf{M}, \mathbf{y} + \mathbf{N})$$

Свойство симметрии относительно операции комплексного сопряжения

$$F(\mathbf{u}, \mathbf{v}) = F^*(-\mathbf{u}, -\mathbf{v})$$

Отсюда следует, что спектр центрально симметричен относительно начала координат

$$|F(\mathbf{u}, \mathbf{v})| = |F(-\mathbf{u}, -\mathbf{v})|$$

В случае двумерного преобразования спектр определяется как

$$|F(\mathbf{u}, \mathbf{v})| = [R^2(\mathbf{u}, \mathbf{v}) + I^2(\mathbf{u}, \mathbf{v})]^{1/2} \quad \text{где } R = \text{Re}(F), I = \text{Im}(F)$$

Фаза определяется как $\Phi(\mathbf{u}, \mathbf{v}) = \arctg[I(\mathbf{u}, \mathbf{v})/R(\mathbf{u}, \mathbf{v})]$

Энергетический спектр вычисляется как $P(\mathbf{u}, \mathbf{v}) = |F(\mathbf{u}, \mathbf{v})|^2$, а среднее значение будет равно

$$\bar{f}(x, y) = F(0, 0) = \frac{1}{MN} \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} f(x, y)$$

Представление в полярных координатах имеет вид

$$F(\mathbf{u}, \mathbf{v}) = |F(\mathbf{u}, \mathbf{v})| e^{-j\Phi(\mathbf{u}, \mathbf{v})}$$

Свойство разделения переменных означает, что для вычисления двумерного преобразования Фурье необходимо вначале вычислить одномерные преобразования по каждой строке изображения, а затем вычислить одномерные преобразования по каждому столбцу полученного промежуточного результата. Изменение порядка вычислений (сначала по столбцам, затем по строкам) приводит к тому же результату.

52. Двумерные свёртка и корреляция

Дискретная свертка двух функций $f(x,y)$ и $h(x,y)$ размерностью $M*N$ обозначается как $f(x,y)*h(x,y)$ и определяется как

$$f(x,y) * h(x,y) = \frac{1}{MN} \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} f(m,n)h(x-m, y-n)$$

Используя двумерное Фурье-представление, будем иметь

$$f(x,y)*h(x,y) \Leftrightarrow F(u,v)H(u,v)$$

$$f(x,y)h(x,y) \Leftrightarrow F(u,v)*H(u,v)$$

При вычислении двумерной свертки необходимо использовать технологию дополнения нулями. Пусть мы имеем два изображения $f(x,y)$ и $h(x,y)$ размерами $A*B$ и $C*D$ соответственно. Как и в одномерном случае необходимо рассматривать эти массивы как периодические с некоторыми периодами P в направлении x и Q в направлении y . Причем выбор $P \geq A+C-1$ $Q \geq B+D-1$ позволяет избежать ошибок перекрытия при вычислении двумерной свертки. При этом дополнение нулями функций $f(x,y)$, $h(x,y)$ в основном периоде осуществляется следующим образом

$$fe(x,y) = \begin{cases} f(x,y) & 0 \leq x \leq A-1 & 0 \leq Y \leq B-1 \\ 0 & A \leq x \leq P & B \leq Y \leq Q \end{cases}$$

$$he(x,y) = \begin{cases} h(x,y) & 0 \leq x \leq C-1 & 0 \leq Y \leq D-1 \\ 0 & C \leq x \leq P & D \leq Y \leq Q \end{cases}$$

Корреляционная функция двух функций $f(x,y)$ и $h(x,y)$ определяется соотношением

$$f(x,y) \circ h(x,y) = \frac{1}{MN} \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} f^*(m,n)h(x+m, y+n)$$

где f^* обозначает функцию, комплексно сопряженную функции f . Обычно мы имеем дело с действительными функциями и в этом случае $f^*=f$. Корреляция от свертки отличается комплексным сопряжением (в общем случае) и заменой минусов на плюсы во всех членах под знаком суммы. В остальном вычисление корреляционной функции идентично вычислению свертки, включая необходимость дополнения нулями.

Тогда теорема корреляции

$$f(x,y) \circ h(x,y) \Leftrightarrow F^*(u,v)H(u,v)$$

означает, что корреляционная функция в пространственной области может быть получена как результат обратного Фурье-преобразования, примененного к произведению $F^*(u,v)H(u,v)$, где F^* - функция, комплексно сопряженная к F .

Аналогично вычисление корреляционной функции в частотной области сводится к умножению в пространственной области.

$$F(u,v) \circ H(u,v) \Leftrightarrow f^*(x,y)h(x,y)$$

Указанные два утверждения составляют содержание теоремы о корреляции, причем все функции должны быть дополнены нулями.

Если изображения сравниваемые совпадают, то

$$f(x,y) \circ f(x,y) \Leftrightarrow F^*(u,v)F(u,v) \Leftrightarrow |F(u,v)|^2$$

Это соотношение представляет собой энергетический спектр.

Аналогично $|f(x,y)|^2 \Leftrightarrow F(u,v) \circ F(u,v)$

53. Основы фильтрации в частотной области.

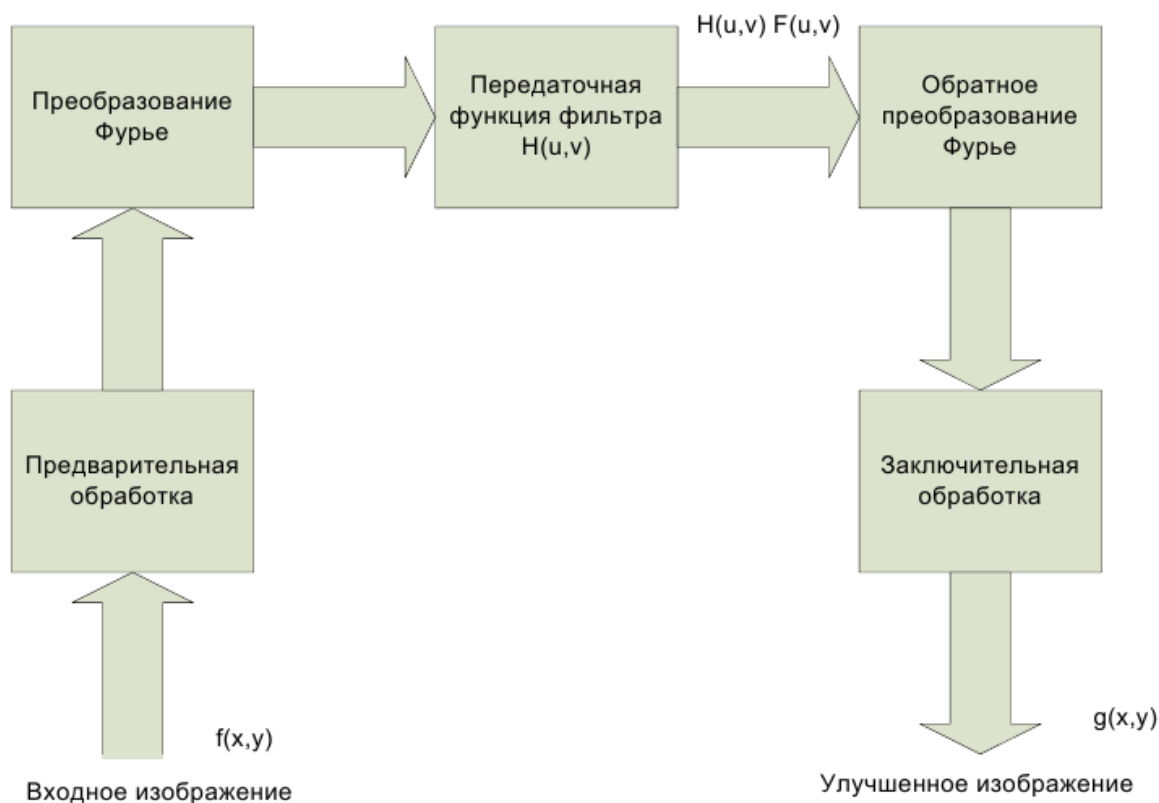


Рис. 19.1

$$F[f(x,y)(-1)^{x+y}] = F(u-M/2, v-N/2)$$

Процедура фильтрации в частотной области состоит из следующих шагов:

1. Исходное изображение умножается на $(-1)^{x+y}$, чтобы его Фурье-преобразование в соответствии с (19.1) оказалось центрированным.
2. Вычисляется прямое ДПФ $F(u,v)$ изображения, полученного после шага 1.
3. Функция $F(u,v)$ умножается на $H(u,v)$ (передаточная функция фильтра).
4. Вычисляется обратное ДПФ от результата шага 3.
5. Выделяется вещественная часть результата шага 4
6. Результат шага 5 умножается на $(-1)^{x+y}$,

Причина, по которой множитель $H(u,v)$ называется фильтром, состоит в том, что он подавляет некоторые частоты преобразования, оставляя при этом другие без изменения.

Аналогия с повседневной жизнью возникает при рассмотрении сетчатого фильтра (сито), который пропускает некоторые предметы и не пропускает другие в соответствии с их размерами.

Пусть $f(x,y)$ обозначает входное изображение после шага 1 и пусть $F(u,v)$ есть его Фурье-образ. Тогда Фурье-образ выходного изображения определяется соотношением $G(u,v) = H(u,v)F(u,v)$

Умножение функций двух переменных H и F осуществляется поэлементно. Это означает, что первый элемент H умножается на первый элемент F , второй элемент H –

на второй элемент F и так далее. Фильтрованное изображение получается вычислением обратного преобразования Фурье от Фурье-образа $G(u,v)$

$$g_{\text{фильтр}}(x,y) = F^{-1} [G(u,v)]$$

Искомое изображение получается выделением действительной части из последнего результата и умножения на $(-1)^{x+y}$, чтобы скомпенсировать эффект от умножения входного изображения на ту же величину.

Обратное Фурье-преобразование в общем случае является комплексным. Однако, в случае вещественного входного изображения и вещественной передаточной функции фильтра мнимые части всех значений обратного Фурье-преобразования должны равняться нулю. Однако в связи с ошибками округления при вычислениях на практике значения обратного Фурье-преобразования, как правило, содержат паразитную мнимую составляющую, которой необходимо при решении прикладных задач пренебречь.

Процедура фильтрации, приведенная на рис.19.1, представлена в общем виде, включая предварительную и заключительную обработку. Помимо умножения на $(-1)^{x+y}$ такая обработка может включать обрезание входного изображения так, чтобы его размеры приняли ближайшие четные значения по отношению к исходным (это необходимо для правильного центрирования Фурье-преобразования), яркостное масштабирование, преобразование форматов входных и выходных данных. Возможны многоступенчатые процедуры фильтрации, а также разнообразные операции предварительной и заключительной обработки. Существенным является то, что метод фильтрации основан на некотором изменении Фурье-образа изображения посредством передаточной функции фильтра и последующем обращении результата для получения обработанного выходного изображения. метод фильтрации основан на некотором изменении Фурье-образа изображения посредством передаточной функции фильтра и последующем обращении результата для получения обработанного выходного изображения.

54. Фильтр, основанные на порядковых статистиках.

Фильтры, основанные на порядковых статистиках, относятся к классу нелинейных пространственных фильтров. Отклик фильтра определяется предварительным упорядочиванием (ранжированием), значений пикселей, покрываемых маской фильтра и последующим выбором значения, находящегося на определенной позиции упорядоченной последовательности (т. е. имеющего определенный ранг). При этом фильтрация сводится к замещению исходного значения пикселя (в центре маски) на полученное значение отклика фильтра. Из указанного класса наиболее известен медианный фильтр, который заменяет значение пикселя на значение медианы распределения яркости всех пикселей в окрестности, включая исходный. Медианные фильтры особенно эффективны при фильтрации импульсных шумов, иногда называемых шумами «соль и перец», которые выглядят как наложение на изображение случайных черных и белых точек.

Медиана набора чисел есть такое число ξ , что половина чисел из набора меньше или равны ξ , а другая половина – больше и равны ξ . Чтобы выполнить медианную фильтрацию для элемента изображения, необходимо сначала упорядочить по возрастанию значения пикселей внутри окрестности, затем найти значение медианы и наконец присвоить полученное значение обрабатываемому элементу. Так для окрестности размером 3×3 медианой будет пятое значение по величине, для окрестности 5×5 – тринадцатое значение и так далее.

Аналитически действие медианного фильтра определяется как

$$\hat{f}(x, y) = \underset{s, t \in S_{xy}}{med} \{g(s, t)\}$$

Причем при вычислении медианы значение в самой точке (т. е. в центре окружности) также учитывается.

Если несколько элементов в окрестности имеют одинаковые значения, то значения будут сгруппированы. Например, пусть в окрестности 3×3 имеют место следующие значения (10, 20, 20, 20, 15, 20, 20, 25, 100). После упорядочения они будут расположены следующим образом (10, 15, 20, 20, 20, 20, 20, 25, 100), а, следовательно, медианой будет значение 20.

Хотя медианный фильтр значительно более распространен в обработке изображений, чем остальные виды фильтров, тем не менее, он не является единственным. Медиана представляет собой 50-ый перцентиль упорядоченного набора чисел.

Широкая популярность медианных фильтров обусловлена тем, что они прекрасно приспособлены для подавления импульсных шумов и при этом приводят к меньшему размыванию по сравнению с линейными сглаживающими фильтрами.

Использование 100-го перцентилля приводит к фильтру, основанному на выборе максимального значения (фильтр максимума), который задается выражением

$$\hat{f}(x, y) = \underset{(s, t) \in S_{xy}}{\max} \{g(s, t)\}$$

Такой фильтр полезен при обнаружении наиболее ярких точек на изображении. Использование 0-го перцентиля приводит к фильтру, основанному на выборе минимального значения (фильтр минимума)

$$\hat{f}(x, y) = \min_{(s,t) \in S_{xy}} \{g(s, t)\}$$

Такой фильтр полезен при обнаружении наиболее темных точек на изображении. Применение фильтра срединной точки заключается в вычислении среднего между максимальным и минимальным значениями в соответствующей окрестности

$$\hat{f}(x, y) = \frac{1}{2} [\max_{(s,t) \in S_{xy}} \{g(s, t)\} + \min_{(s,t) \in S_{xy}} \{g(s, t)\}]$$

Этот фильтр объединяет методы порядковых статистик и усреднения. Этот фильтр лучше всего работает при наличии таких случайно распределенных шумов, как гауссов или равномерный.

Предположим, что мы удалили $d/2$ наименьших и $d/2$ наибольших значений яркости из множества всех значений функции $g(s,t)$ в окрестности S_{xy} . Пусть $g_r(s,t)$ представляет оставшиеся элементы изображения, количество которых равно $(mn-d)$.

Фильтр, действие которого заключается в усреднении оставшихся значений, называется фильтром усеченного среднего

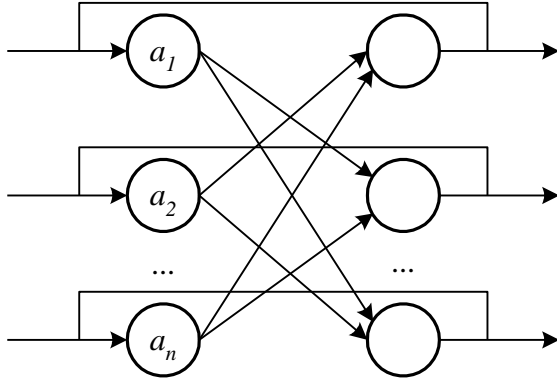
$$\hat{f}(x, y) = \frac{1}{mn - d} \sum_{(s,t) \in S_{xy}} q_r(s, t),$$

$$\text{где } d = \overline{0, mn - 1}$$

При $d=0$ фильтр усеченного среднего сводится к среднеарифметическому фильтру. В случае $d/2=(mn-1)/2$ фильтр трансформируется в медианный.

55. Нейронная сеть Хопфилда.

Сеть Хопфилда – однослойная, симметричная, нелинейная, автоассоциативная нейронная сеть, которая запоминает бинарные / биполярные образы. Сеть характеризуется наличием обратных связей. Топология сети Хопфилда показана на рис. 1.1. Информация с выхода каждого нейрона поступает на вход всех остальных нейронов. Образы для данной модификации сети Хопфилда кодируются биполярным вектором, состоящим из 1 и -1 .



Обучение.

Обучение сети осуществляется в соответствии с соотношением

$$w_{ij} = \begin{cases} \sum_{k=1}^m a_i^k a_j^k, & i \neq j, \text{ для } i, j = \overline{1, n}, \\ 0, & i = j \end{cases} \quad (1.1)$$

где w_{ij} – вес связи от i -го нейрона к j -му;

n – количество нейронов в сети;

m – количество образов, используемых для обучения сети;

a_i^k – i -й элемент k -го образа из обучающей выборки.

Матрица весовых коэффициентов

$$W = \begin{bmatrix} w_{11} & w_{12} & \dots & w_{1n} \\ w_{21} & w_{22} & \dots & w_{2n} \\ \dots & \dots & \dots & \dots \\ w_{n1} & w_{n2} & \dots & w_{nn} \end{bmatrix}. \quad (1.2)$$

В качестве матрицы весовых коэффициентов Хопфилд использовал симметричную матрицу ($w_{ij}=w_{ji}$) с нулевой главной диагональю ($w_{ii}=0$). Последнее условие соответствует отсутствию обратной связи нейронного элемента на себя. В качестве функции активации нейронных элементов может использоваться как пороговая, так и непрерывная функции, например сигмоидная или гиперболический тангенс.

Будем рассматривать нейронную сеть Хопфилда с дискретным временем. Тогда при использовании пороговой функции активации она называется *нейронной сетью с дискретным состоянием и временем*. Нейронная сеть с непрерывной функцией активации называется нейронной сетью с непрерывным состоянием и дискретным временем. При использовании непрерывного времени модель Хопфилда называется непрерывной.

Для описания функционирования таких сетей Хопфилд использовал аппарат статистической физики. При этом каждый нейрон имеет два состояния активности ($1, -1$), которые аналогичны значениям спина некоторой частицы. Весовой коэффициент w_{ji} можно интерпретировать как вклад поля j -частицы в величину потенциала i -частицы. Хопфилд показал, что поведение такой сети аналогично поведению лизингового спинового стекла. При этом он ввел понятие вычислительной энергии, которую можно интерпретировать в виде ландшафта с долинами и впадинами. Структура соединений сети определяет очертания ландшафта. Нейронная сеть выполняет вычисления, следуя по пути, уменьшающему вычислительную энергию сети. Это происходит до тех пор, пока путь не приведет на дно впадины. Данный процесс аналогичен скатыванию капли жидкости по склону, когда она минимизирует свою потенциальную энергию в поле тяготения. Впадины и долины в сети Хопфилда соответствуют наборам информации, которую хранит сеть. Если процесс начинается с приближенной или неполной информации, то он следует по пути, который ведет к ближайшей впадине. Это соответствует операции ассоциативного распознавания.

Матрица весов является диагонально симметричной, причем все диагональные элементы равны 0 .

Воспроизведение. Нейронная сеть Хопфилда может функционировать синхронно и асинхронно. Для воспроизведения используется соотношение

$$a_i(t+1) = f\left(\sum_{j=1}^n w_{ij} a_j(t)\right), \quad (1.3)$$

где $a_j(t)$ – выход j -го нейрона в момент времени t , а f – бинарная / биполярная функция активации;

$$f(x) = \begin{cases} 1 & x > 0, \\ -1 & x \leq 0. \end{cases} \quad (1.4)$$

При работе в синхронном режиме на один такт работы сети все нейроны одновременно меняют состояние по формуле (1.4). В случае асинхронной работы состояние меняет только один случайный нейрон. Итерации продолжаются до тех пор, пока сеть не придет в стабильное состояние.

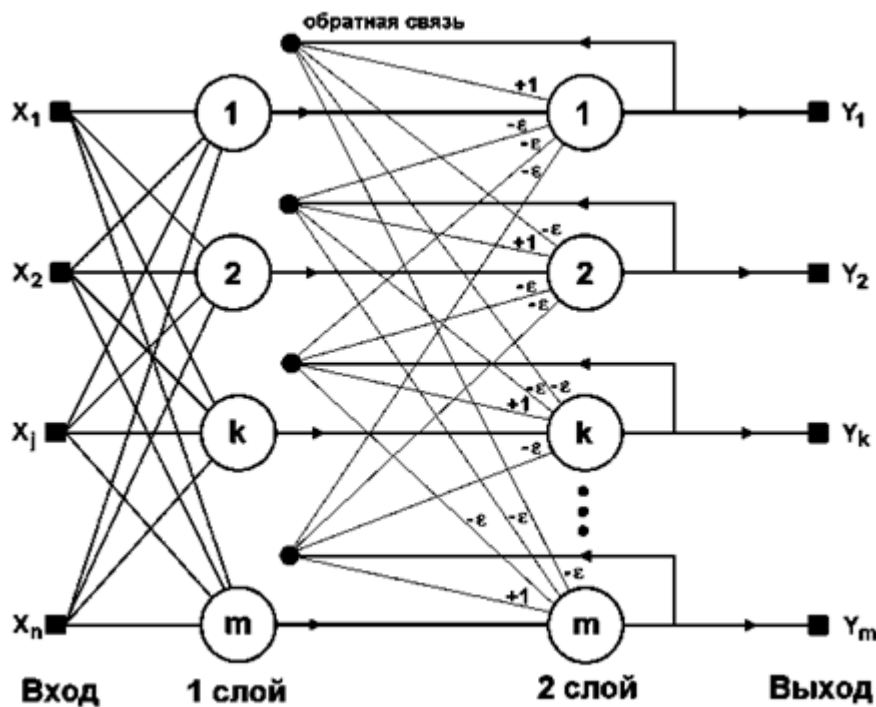
Во время воспроизведения исходным вектором $a(0)$ является некоторый тестовый образ, не совпадающий с образами из обучающей выборки. В процессе функционирования по формуле (1.4) сеть должна прийти в состояние, соответствующее образу из обучающей выборки, наиболее похожему на тестовый.

Максимальное количество образов, которое можно запомнить в матрице W , не превышает

$$m = \frac{n}{2 \ln n + \ln \ln n}, \quad (1.5)$$

где n – количество нейронов, что следует отнести к недостаткам этой сети.

56. Нейронная сеть Хэмминга



Когда нет необходимости, чтобы сеть выдавала образец в явном виде и достаточно, скажем, получать номер образца, ассоциативную память успешно реализует *сеть Хэмминга*. Данная сеть характеризуется, по сравнению с *сетью Хопфилда*, более экономным

использованием памяти и меньшим объемом вычислений, что становится очевидным из ее структуры (см. [рис. 8.2](#)).

Рис. 8.2.

Сеть состоит из двух слоев. Первый и второй слои имеют по m нейронов, где m — число образцов. Нейроны первого слоя имеют по n синапсов, соединенных с входами сети (которые образуют фиктивный нулевой слой). Нейроны второго слоя связаны между собой ингибиторными (отрицательными *обратными*) синаптическими связями. Единственный синапс с положительной *обратной связью* для каждого нейрона соединен с его же аксоном.

Идея работы сети состоит в нахождении расстояния Хэмминга от тестируемого образа до всех образцов. Расстоянием Хэмминга называется число отличающихся битов в двух бинарных векторах. Сеть должна выбрать образец с минимальным расстоянием Хэмминга до неизвестного входного сигнала, в результате чего будет активизирован только один выход сети, соответствующий именно этому образцу.

На стадии инициализации весовым коэффициентам первого слоя и порогу активационной функции присваиваются следующие значения:

$$w_{ik} = \frac{x_i^k}{2}, \quad i = 0, \dots, n-1, k = 0, \dots, m-1,$$

$$T_k = \frac{n}{2}, \quad k = 0, \dots, m-1.$$

Здесь x_i^k - i -й элемент k -го образца.

Весовые коэффициенты тормозящих синапсов во втором слое берут равными некоторой величине $0 < \varepsilon < 1/m$. Синапс нейрона, связанный с его же аксоном, имеет вес +1.

Алгоритм функционирования сети Хэмминга следующий:

1. На входы сети подается неизвестный вектор

$$X = \{x_i | i = 0, \dots, n\},$$

исходя из которого рассчитываются состояния нейронов первого слоя (верхний индекс в скобках указывает номер слоя):

$$y_j^{(1)} = s_j^{(1)} = \sum_{i=0}^{n-1} w_{ij} x_i + T_j, \quad j = 0, \dots, m-1.$$

После этого полученными значениями инициализируются значения аксонов второго слоя:

$$y_j^{(2)} = y_j^{(1)}, \quad j = 0, \dots, m-1.$$

2. Вычисляются новые состояния нейронов второго слоя:

$$s_j^{(2)}(p+1) = y_j(p) - \varepsilon \sum_{k=0}^{m-1} y_k^{(2)}(p), \quad k \neq j, \quad j = 0, \dots, m-1$$

и значения их аксонов:

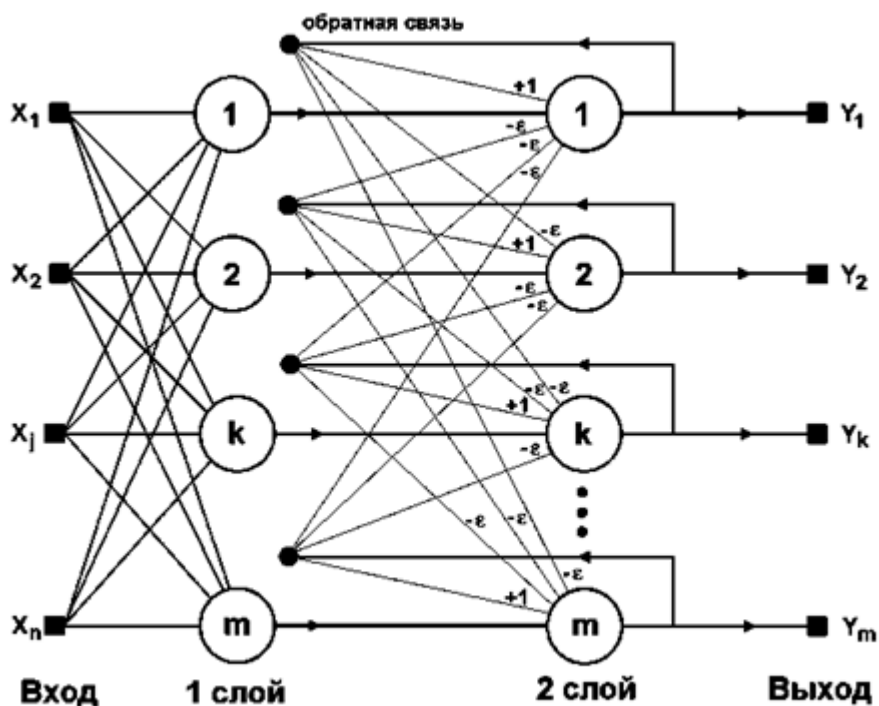
$$y_j^{(2)}(p+1) = f[s_j^{(2)}(p+1)], \quad j = 0, \dots, m-1.$$

Активационная функция f имеет вид порога, причем величина F должна быть достаточно большой, чтобы любые возможные значения аргумента не приводили к насыщению.

3. Проверить, изменились ли выходы нейронов второго слоя за последнюю итерацию. Если да — перейти к шагу 2. Иначе — конец процедуры.

Из оценки алгоритма видно, что роль первого слоя весьма условна: воспользовавшись один раз на шаге 1 значениями его весовых коэффициентов, сеть больше не обращается к нему, поэтому первый слой может быть вообще исключен из сети.

Сеть Хэмминга



Когда нет необходимости, чтобы сеть в явном виде выдавала образец, то есть достаточно, скажем, получать номер образца, ассоциативную память успешно реализует сеть Хэмминга. Данная сеть характеризуется, по сравнению с сетью Хопфилда, меньшими затратами на память и объемом вычислений, что становится очевидным из ее структуры (Рис. 3).

Сеть состоит из двух слоев. Первый и второй слои имеют по m нейронов, где m – число образцов. Нейроны первого слоя имеют по n синапсов, соединенных со входами сети (образующими фиктивный нулевой слой). Нейроны второго слоя связаны между собой ингибиторными (отрицательными обратными) синаптическими связями. Единственный синапс с положительной обратной связью для каждого нейрона соединен с его же аксоном.

Идея работы сети состоит в нахождении расстояния Хэмминга от тестируемого образа до всех образцов. Расстоянием Хэмминга называется число отличающихся битов в двух бинарных векторах. Сеть должна выбрать образец с минимальным расстоянием Хэмминга до неизвестного входного сигнала, в результате чего будет активизирован только один выход сети, соответствующий этому образцу.

На стадии инициализации весовым коэффициентам первого слоя и порогу активационной функции присваиваются следующие значения:

$$w_{ik} = \frac{x_i^k}{2}, i=0\dots n-1, k=0\dots m-1 \quad (5)$$

$$T_k = n / 2, k = 0\dots m-1 \quad (6)$$

Здесь x_i^k – i -ый элемент k -ого образца.

Весовые коэффициенты тормозящих синапсов во втором слое берут равными некоторой величине $0 < \varepsilon < 1/m$. Синапс нейрона, связанный с его же аксоном имеет вес $+1$.

Алгоритм функционирования сети Хэмминга следующий:

1. На входы сети подается неизвестный вектор $X = \{x_i; i=0\dots n-1\}$, исходя из которого рассчитываются состояния нейронов первого слоя (верхний индекс в скобках указывает номер слоя):

$$y_j^{(1)} = s_j^{(1)} = \sum_{i=0}^{n-1} w_{ij} x_i + T_j, j=0\dots m-1 \quad (7)$$

После этого полученными значениями инициализируются значения аксонов второго слоя:

$$y_j^{(2)} = y_j^{(1)}, j = 0\dots m-1 \quad (8)$$

2. Вычислить новые состояния нейронов второго слоя:

$$s_j^{(2)}(p+1) = y_j^{(2)}(p) - \varepsilon \sum_{k=0}^{m-1} y_k^{(2)}(p), k \neq j, j = 0\dots m-1 \quad (9)$$

и значения их аксонов:

$$y_j^{(2)}(p+1) = f[s_j^{(2)}(p+1)], j = 0\dots m-1 \quad (10)$$

Активационная функция f имеет вид порога (рис. 2б), причем величина F должна быть достаточно большой, чтобы любые возможные значения аргумента не приводили к насыщению.

3. Проверить, изменились ли выходы нейронов второго слоя за последнюю итерацию. Если да – перейди к шагу 2. Иначе – конец.

57. РБФ – сеть.

Сеть РБФ (радиальная базисная функция) является аналогом многослойного персептрона (рис. 3.1). Скорость обучения такой сети гораздо выше, причем допускается полностью аналитический подход к расчету весовых коэффициентов. Однако эти положительные моменты сопровождаются рядом недостатков, главным из которых является ухудшение точности аппроксимации. Сеть обладает хорошей обобщающей способностью только для ограниченного класса аппроксимируемых функций. В качестве классификатора такая сеть может с успехом применяться в случае хорошей кластеризации классов в пространстве признаков.

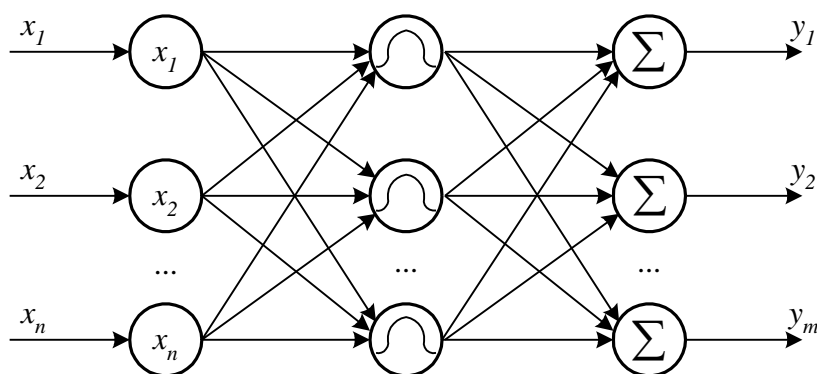


Рис. 3.1. Топология сети РБФ

Упрощение работы и обучения достигается за счет введения вместо скрытого слоя нейронов слоя РБФ ячеек. Классический закон, по которому такая ячейка функционирует, определяется формулой гауссового колокола:

$$g_j = \exp\left(\frac{-|x - t^j|^2}{\sigma_j^2}\right),$$

(3.1)

где x – входной вектор; t^j – вектор, определяющий математическое ожидание (центр кластера в пространстве признаков) РБФ ячейки; σ_j – среднеквадратическое отклонение или параметр, зависящий от величины разброса образов класса от его центра (рис. 3.1). В данном выражении евклидово расстояние между векторами x и t^j вычисляется как

$$|x - t^j|^2 = (x_1 - t_1^j)^2 + (x_2 - t_2^j)^2 + \dots + (x_n - t_n^j)^2.$$

Обучение. РБФ ячейки обучаются путем подбора центра и отклонения каждой из них. Для классификатора в качестве центра выбирается центр кластера в пространстве признаков, компактно содержащего образы одного и того же класса. В простейшем случае, если класс задается одним идеальным образом,

этот образ и будет являться вектором t – центром РБФ ячейки. Параметр разброса каждой ячейки выбирается в зависимости от величины радиуса кластера или расстояния до соседних центров. Ряд авторов рекомендует выбирать σ как половину расстояния до ближайшего центра ячейки, соответствующей другому классу. Количество РБФ ячеек выбирается таким образом, чтобы покрыть гауссовыми колоколами все классы.

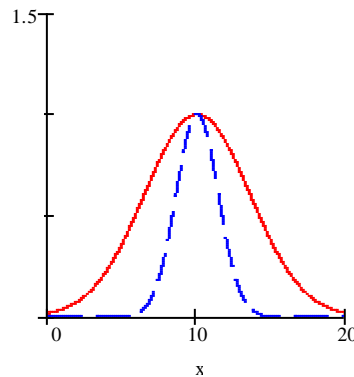


Рис. 4.2. Примеры функций РБФ ячеек с одинаковым центром и разным разбросом

Выходной слой РБФ сети обычно состоит из суммирующих ячеек

$$y_k = \sum_{j=1}^h w_{jk} g_j. \quad (3.2)$$

Это позволяет при определенных условиях использовать систему линейных уравнений для определения весов выходного слоя. В других обстоятельствах можно использовать алгоритм градиентного спуска для настройки весов выходного слоя (функция активации в данном случае линейная). С учетом того что обучается только выходной слой нейронной сети, а скрытый уже настроен, обучение такой сети проходит на порядок быстрее, чем аналогичного многослойного персептрона.

В случае обучения выходного слоя по алгоритму градиентного спуска формулы расчета из лабораторной работы 2 значительно упрощаются. Формулы (2.1) и (2.2) заменяются соответственно на (3.1) и (3.2). Формула (2.13) упрощается до

$$w_{jk} = w_{jk} + \alpha d_k g_j, \quad (3.3)$$

поскольку функция активации в выходном слое сети РБФ линейная и ее производная равна 1. Для коррекции весовых коэффициентов выходных суммирующих ячеек используется только (3.3).

Контроль завершения алгоритма обучения производится аналогично.

Воспроизведение. Сеть функционирует по формулам (3.1) и (3.2).

Очевидно, что функция на выходе РБФ сети будет представлять собой суперпозицию гауссовых колоколов. В этом заключается ограничение данного класса классификаторов. Кроме этого, при неудачном выборе признаков большой проблемой является выбор количества РБФ ячеек, определение их центров и отклонения. С уменьшением числа РБФ ячеек улучшается обобщение данных сетью, но могут проявляться большие ошибки в эталонных точках.

58. Функция когерентности

Функция когерентности $\gamma_{xy}(f)$ определяется путем нормировки взаимной

$$\gamma_{xy}^2(f) = \frac{|S_{xy}(f)|^2}{S_{xx}(f)S_{yy}(f)}$$

спектральной плотности:

и для любых частот f удовлетворяет неравенствам:

$$0 \leq \gamma_{xy}^2(f) \leq 1.$$

Функция когерентности является аналогом коэффициента корреляции в частотной области и отражает степень линейной взаимосвязи гармонических компонент рассматриваемых процессов. Чем ближе функция когерентности к

единице на данной частоте f , тем больше совпадение гармонических составляющих на этой частоте. Как правило, именно функция когерентности, а не взаимная спектральная плотность, используется в практических приложениях для анализа связанности процессов в частотной области.