

Министерство образования Республики Беларусь

Учреждение образования
БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
ИНФОРМАТИКИ И РАДИОЭЛЕКТРОНИКИ

Кафедра электронных вычислительных машин

Р.Х. Садыхов, О.Г. Маленко

ЛАБОРАТОРНЫЙ ПРАКТИКУМ
по курсам «Цифровая обработка сигналов и изображений»
и «Методы и средства обработки изображений»
для студентов специальностей 40 02 01 и 40 01 01
всех форм обучения

Минск 2002

УДК 621.391.2 (075.8)

ББК 32.811 я 7

С 14

Садыхов Р.Х.

С 14 Лабораторный практикум по курсам «Цифровая обработка сигналов и изображений» и «Методы и средства обработки изображений» для студентов специальностей 40 02 01 и 40 01 01 всех форм обучения / Р.Х. Садыхов, О.Г. Маленко. — Мн.: БГУИР, 2002. — 52 с.: ил.

ISBN 985-444-415-5

В настоящем лабораторном практикуме рассматривается обработка сигналов и изображений с помощью дискретных преобразований и нейронных сетей. Особое внимание уделяется оптимизации вычислений путем создания быстрых алгоритмов расчета и обучения. Рассматриваются нейросетевые классификаторы, позволяющие принимать решения в задачах распознавания и управления.

УДК 621.391.2 (075.8)
ББК 32.811 я 7

ISBN 985-444-415-5

© Р.Х. Садыхов, О.Г. Маленко, 2002
© БГУИР, 2002

1. ДИСКРЕТНЫЕ ПРЕОБРАЗОВАНИЯ

Лабораторная работа № 1.1 Изучение преобразования Фурье

Теоретическая часть

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

$$\int_{t_0}^{t_0+T} U_m(t)U_n(t)dt = \begin{cases} c, \forall m = n, \\ 0, \forall m \neq n. \end{cases} \quad (1.1)$$

При $c=1$ множество $\{U_n(t)\}$ называется ортонормированным.

В качестве иллюстрации общего метода ортогональных представлений рассмотрим случай, когда множество функций $\{U_n(t)\}$ есть множество тригонометрических функций $\{1, \cos n\omega_0 t, \sin n\omega_0 t\}$. Тогда разложение в ряд примет вид

$$x(t) = a_0 + \sum_{n=1}^{\infty} a_n \cos n\omega_0 t + \sum_{n=1}^{\infty} b_n \sin n\omega_0 t, \quad (1.2)$$

где ω_0 (рад/с) — основная угловая частота, которая связана с периодом T функции соотношением $T=2\pi/\omega_0$. Частоты $n\omega_0$ называют гармониками, так как они кратны основной частоте ω_0 .

Коэффициенты $\{a_0, a_n, b_n\}$ можно вычислить с учетом ортогональности множества функций $\{\cos n\omega_0 t, \sin n\omega_0 t\}$ на периоде T :

$$\int_T \cos n\omega_0 t \cos m\omega_0 t dt = \begin{cases} T/2, m = n, \\ 0, m \neq n; \end{cases}$$

$$\int_T \cos n\omega_0 t \sin m\omega_0 t dt = 0, \forall m, n;$$

$$\int_T \sin n\omega_0 t \sin m\omega_0 t dt = \begin{cases} T/2, m = n, \\ 0, m \neq n. \end{cases}$$

С учетом этих соотношений

$$a_0 = \frac{1}{T} \int_T x(t) dt; \quad a_n = \frac{2}{T} \int_T x(t) \cos n\omega_0 t dt; \quad (1.3)$$

$$b_n = \frac{2}{T} \int_T x(t) \sin n\omega_0 t dt.$$

Для получения представления $x(t)$ рядом Фурье в комплексной форме необходимо использовать соотношения в виде формулы Эйлера:

$$\begin{aligned}\cos n\omega_0 t &= \frac{1}{2}(e^{in\omega_0 t} + e^{-in\omega_0 t}); \\ \sin n\omega_0 t &= \frac{1}{2i}(e^{in\omega_0 t} - e^{-in\omega_0 t}); \quad i = \sqrt{-1}.\end{aligned}\quad (1.4)$$

Тогда

$$\begin{aligned}x(t) &= a_0 + \frac{1}{2} \sum_{n=1}^{\infty} \{a_n(e^{in\omega_0 t} + e^{-in\omega_0 t}) - ib_n(e^{in\omega_0 t} - e^{-in\omega_0 t})\} = \\ &= a_0 + \frac{1}{2} \sum_{n=1}^{\infty} \{(a_n - ib_n)e^{in\omega_0 t} + (a_n + ib_n)e^{-in\omega_0 t}\}.\end{aligned}\quad (1.5)$$

Введем коэффициент

$$C_n = \frac{1}{2}(a_n - ib_n).$$

Тогда

$$\begin{aligned}C_n &= \frac{1}{T} \int_T x(t) [\cos n\omega_0 t - i \sin n\omega_0 t] dt \quad \text{или} \quad C_n = \frac{1}{T} \int_T x(t) e^{-in\omega_0 t} dt; \\ C_{-n} &= C_n^* = \frac{1}{2}(a_n + ib_n).\end{aligned}\quad (1.6)$$

Следовательно,

$$\begin{aligned}x(t) &= a_0 + \sum_{n=1}^{\infty} [C_n e^{in\omega_0 t} + C_n^* e^{-in\omega_0 t}] = a_0 + \sum_{\substack{n=-\infty \\ n \neq 0}}^{\infty} C_n e^{in\omega_0 t}; \\ C_0 &= \frac{1}{T} \int_T x(t) dt = a_0; \\ x(t) &= \sum_{n=-\infty}^{\infty} C_n e^{in\omega_0 t}.\end{aligned}\quad (1.7)$$

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

$$C_x(k) = \frac{1}{N} \sum_{m=0}^{N-1} X(m) W^{km}, \quad (1.8)$$

где $k = \overline{0, N-1}$, $W = e^{-i2\pi/N}$, $i = \sqrt{-1}$,

$$X(m) = \sum_{k=0}^{N-1} C_x(k) W^{-km}. \quad (1.9)$$

Выражения (1.8), (1.9) составляют пару 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.

$$X(\pm m) = X(SN \pm m);$$

$$C_x(\pm k) = C_x(SN \pm k).$$

Рассмотрим основные свойства дискретного преобразования Фурье:

а) теорема линейности: дискретное преобразование Фурье является линейным, т.е. если $X(m) \leftrightarrow C_x(k)$ 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

$$C_x\left(\frac{N}{2} + l\right) = \overline{C_x\left(\frac{N}{2} - l\right)}, \forall l = \overline{0, N/2}. \quad (1.10)$$

Из (1.10) 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 вместо k — $(N/2+l)$, будем иметь

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

так как $W^{Nm} = -1$;

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

$$C_z(k) = W^{-kh} C_x(k). \quad (1.11)$$

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

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

Так как

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

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

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

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

Быстрое преобразование Фурье

Алгоритм быстрого преобразования Фурье значительно сокращает количество арифметических операций и объем памяти, необходимой для вычисления ДПФ.

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.

Рассмотрим случай вещественно-значной последовательности $\{X(m)\}$ при $N=8$. Из свойства комплексной сопряженности ДПФ следует, что

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

Тогда

$$C_x(k) = \frac{1}{8} \sum_{m=0}^7 X(m)W^{km}; \quad k = \overline{0, N-1};$$

$$W = e^{-i2\pi/8} = e^{-i\pi/4};$$

$$8C_x(k) = \sum_{m=0}^7 X(m) \cos\left(\frac{mk\pi}{4}\right) - i \sum_{m=0}^7 X(m) \sin\left(\frac{mk\pi}{4}\right). \quad (1.12)$$

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

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

То есть $W_4 = -W_0$;

$$W_5 = -W_1;$$

$$W_6 = -W_2;$$

$$W_7 = -W_3.$$

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

$$F = \begin{bmatrix} W_0 & W_0 & W_0 & W_0 & W_0 & W_0 & W_0 & W_0 \\ W_0 & -W_0 & W_0 & -W_0 & W_0 & -W_0 & W_0 & -W_0 \\ W_0 & W_2 & -W_0 & -W_2 & W_0 & W_2 & -W_0 & -W_2 \\ W_0 & -W_2 & -W_0 & W_2 & W_0 & -W_2 & -W_0 & W_2 \\ W_0 & W_1 & W_2 & W_3 & -W_0 & -W_1 & -W_2 & -W_3 \\ W_0 & -W_1 & W_2 & -W_3 & -W_0 & W_1 & -W_2 & W_3 \\ W_0 & W_3 & -W_2 & W_1 & -W_0 & -W_3 & W_2 & -W_1 \\ W_0 & -W_3 & -W_2 & -W_1 & -W_0 & W_3 & W_2 & W_1 \end{bmatrix} \begin{bmatrix} X_0 \\ X_1 \\ X_2 \\ X_3 \\ X_4 \\ X_5 \\ X_6 \\ X_7 \end{bmatrix} = \begin{bmatrix} A_1^1 & A_1^1 \\ B_1^1 & -B_1^1 \end{bmatrix} \begin{matrix} \rho \\ X \end{matrix} =$$

$$= \begin{matrix} A_1^1 \begin{Bmatrix} X_0 + X_4 \\ X_1 + X_5 \\ X_2 + X_6 \\ X_3 + X_7 \end{Bmatrix} \\ B_1^1 \begin{Bmatrix} X_0 - X_4 \\ X_1 - X_5 \\ X_2 - X_6 \\ X_3 - X_7 \end{Bmatrix} \end{matrix}. \quad (1.13)$$

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

$$A_1^1 = \begin{vmatrix} A_1^2 & A_1^2 \\ A_2^2 & -A_2^2 \end{vmatrix}; \quad (1.14)$$

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

Подставляя выражения для A_1^1 и B_1^1 в (1.13), получим

$$\begin{aligned}
 & A_1^2 \left\{ \begin{array}{l} (X_0 + X_4) + (X_2 + X_6) \\ (X_1 + X_5) + (X_3 + X_7) \end{array} \right\}; \\
 & A_2^2 \left\{ \begin{array}{l} (X_0 + X_4) - (X_2 + X_6) \\ (X_1 + X_5) - (X_3 + X_7) \end{array} \right\}; \\
 & B_1^2 \left\{ \begin{array}{l} (X_0 - X_4) + W_2(X_2 - X_6) \\ (X_1 - X_5) + W_2(X_3 - X_7) \end{array} \right\}; \\
 & B_2^2 \left\{ \begin{array}{l} (X_0 - X_4) - W_2(X_2 - X_6) \\ (X_1 - X_5) - W_2(X_3 - X_7) \end{array} \right\}.
 \end{aligned} \tag{1.15}$$

Наконец на последнем шаге получим

$$\begin{aligned}
 & [(X_0 + X_4) + (X_2 + X_6)] + [(X_1 + X_5) + (X_3 + X_7)] \\
 & [(X_0 + X_4) + (X_2 + X_6)] - [(X_1 + X_5) + (X_3 + X_7)] \\
 & [(X_0 + X_4) - (X_2 + X_6)] + W_2[(X_1 + X_5) - (X_3 + X_7)] \\
 & [(X_0 + X_4) - (X_2 + X_6)] - W_2[(X_1 + X_5) - (X_3 + X_7)] \\
 & [(X_0 - X_4) + W_2(X_2 - X_6)] + W_1[(X_1 - X_5) + W_2(X_3 - X_7)] \\
 & [(X_0 - X_4) + W_2(X_2 - X_6)] - W_1[(X_1 - X_5) + W_2(X_3 - X_7)] \\
 & [(X_0 - X_4) - W_2(X_2 - X_6)] + W_3[(X_1 - X_5) - W_2(X_3 - X_7)] \\
 & [(X_0 - X_4) - W_2(X_2 - X_6)] - W_3[(X_1 - X_5) - W_2(X_3 - X_7)].
 \end{aligned} \tag{1.16}$$

Описанный алгоритм удобно представить графически (рис. 1.1).

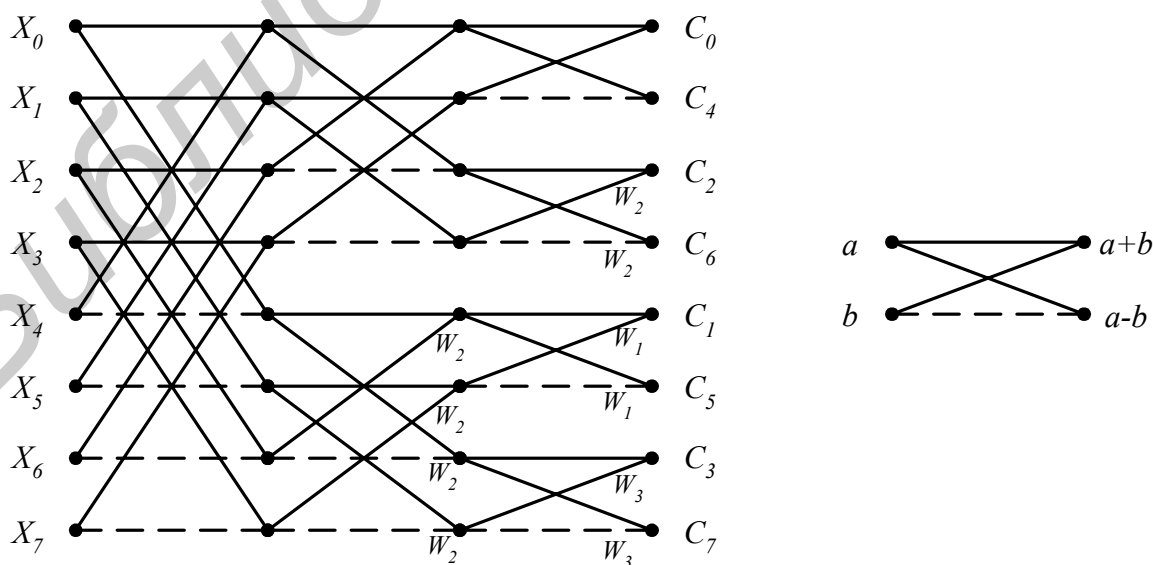


Рис. 1.1. Граф-схема быстрой процедуры вычисления коэффициентов преобразования Фурье

Для определения степеней W на одном 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

$$S_3 = (0, 4, 2, 6, 1, 5, 3, 7),$$

и таким образом имеем $W_0, W_4, W_2, W_6, W_1, W_5, W_3, W_7$ (табл. 1.1).

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 1.1

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

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

$$\begin{array}{cccc}
 0 & & & \\
 n_1 & & & \\
 n_2 & (n_1+n_2) & & \\
 n_3 & (n_1+n_3) & (n_2+n_3) & (n_1+n_2+n_3) \\
 n_4 & (n_1+n_4) & (n_2+n_4) & (n_1+n_2+n_4) \dots \\
 \cdot & & & \\
 \cdot & & & \\
 n_k & \dots & &
 \end{array}$$

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

$$\begin{array}{cccccccc}
 0 & & & & & & & \\
 8 & & & & & & & \\
 4 & 12 & & & & & & \\
 2 & 10 & 6 & 14 & & & & \\
 1 & 9 & 5 & 13 & 3 & 11 & 7 & 15 \\
 L_n = (0, 8, 4, 12, 2, 10, 6, 14, 1, 9, 5, 13, 3, 11, 7, 15).
 \end{array}$$

Для обработки исходных данных (которые предполагаются комплексными) с помощью алгоритма БПФ требуется $2N$ ячеек оперативной памяти. По-

этому 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

$$F_N = \prod_{i=1}^n F_i D_i. \quad (1.17)$$

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 :

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

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

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

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:

процедура БПФ(N, f)

1. If N равно 2, then do:

Begin

2. Замена $f(0)$ на $f(0)+f(1)$ и $f(1)$ на $f(0)-f(1)$.

3. Return

End

4. Else do:

Begin

5. Определите функцию g , как состоящую из всех отсчетов функции f с четным индексом, и функцию h , как состоящую из остальных отсчетов функции f .

6. Call procedure БПФ ($N/2, g$).

7. Call procedure БПФ ($N/2, h$).

8. Замена $f(t)$ на $g(t)+\exp(-i*2*\pi*t/N)h(t)$ for $t=0$ до $N-1$.

9. Замена $f(t+1)$ на $g(t)-\exp(-i*2*\pi*t/N)h(t)$ for $t=0$ до $N-1$.

End

Конец процедуры.

Процедура предусматривает обращение самой к себе при $N>2$, а при $N=2$ — использование следующих формул:

$$F(0)=f(0)+f(1);$$

$$F(1)=f(0)-f(1).$$

Совершенно очевидно, что последовательное деление N пополам можно реализовать лишь в случае, когда N есть некоторая степень числа 2, и только для таких значений существует простой алгоритм БПФ. Если $C(N)$ — трудоемкость для случая N отсчетов, то $C(N)=N\log_2 N+O(N)$, второй член в правой части которого представляет величины, линейно пропорциональные N .

Задание

1. Внимательно ознакомьтесь с теоретической частью.
2. Удостоверьтесь в правильности работы БПФ.
3. По имеющимся значениям величин постройте графики для входных функций $\sin(x)$, $\cos(x)$.
4. Постройте базовые функции для $N=8$ (включая синусные и косинусные составляющие).
5. Убедитесь в правильности работы преобразования Фурье и докажите, каким образом вы это установили, исходя из базовых составляющих.
6. Постройте входную функцию $\sin(x)$ или $\cos(x)$ для N частоты и опять покажите правильность работы преобразования.

Контрольные вопросы

1. Для чего используются ортогональные преобразования?
2. Дать определение ортогональным и ортонормальным функциям.
3. Доказать, что Фурье-базис является ортогональным.
4. Дать определение преобразования Фурье.
5. Каковы основные свойства преобразования Фурье?
6. Каким образом осуществляется быстрое преобразование Фурье?
7. В чем заключается преимущество быстрого преобразования Фурье?

Лабораторная работа № 1.2

Изучение операций корреляции и свертки

Теоретическая часть

Теорема свертки

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

$$Z(m) = \frac{1}{N} \sum_{h=0}^{N-1} X(h)Y(m-h), \quad m = \overline{0, N-1}, \quad (1.18)$$

то

$$C_z(k) = C_x(k)C_y(k).$$

Доказательство

Вычисляя $Z(m)$, получим

$$C_z(k) = \frac{1}{N} \sum_{m=0}^{N-1} Z(m)W^{km}. \quad (1.19)$$

Подставляя в уравнение свертки (1.18), 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.

Теорема корреляции

Если $X(m) \leftrightarrow C_x(k)$ и $Y(m) \leftrightarrow C_y(k)$, а их функция корреляции определяется соотношением

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

то $C_{\wedge z}(k) = \overline{C_x(k)} C_y(k)$.

Доказательство

По определению имеем

$$C_{\wedge z}(k) = \frac{1}{N} \sum_{m=0}^{N-1} Z(m) W^{km}. \quad (1.20)$$

Подставляя в (1.20) и меняя порядок суммирования, получаем

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

Применяя теорему сдвига, будем иметь

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

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, $C_{\wedge z}(k) = C_y(k) \overline{C_x(k)}$.

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

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

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

Тогда

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

Матричное представление корреляции и свертки

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

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

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

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

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

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

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

$$\begin{bmatrix} \hat{Z}(0) \\ \hat{Z}(1) \\ \hat{Z}(2) \\ \text{M} \\ \hat{Z}(N-2) \\ \hat{Z}(N-1) \end{bmatrix} = \frac{1}{N} \begin{bmatrix} X(0) & X(1) & X(2) & \Lambda & X(N-1) \\ X(N-1) & X(0) & X(1) & \Lambda & X(N-2) \\ X(N-2) & X(N-1) & X(0) & \Lambda & X(N-3) \\ \text{M} & \text{M} & \text{M} & \Lambda & \text{M} \\ X(2) & X(3) & X(4) & \Lambda & X(1) \\ X(1) & X(2) & X(3) & \Lambda & X(0) \end{bmatrix} \begin{bmatrix} Y(0) \\ Y(1) \\ Y(2) \\ \text{M} \\ Y(N-2) \\ Y(N-1) \end{bmatrix}.$$

В свою очередь соотношение свертки можно записать в общем виде как

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

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 вычислений корреляции будет иметь вид рис. 1.2

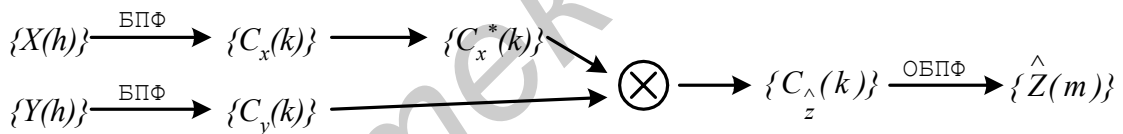


Рис. 1.2. Схема вычисления корреляции

В свою очередь схему вычисления свертки можно представить как на рис. 1.3

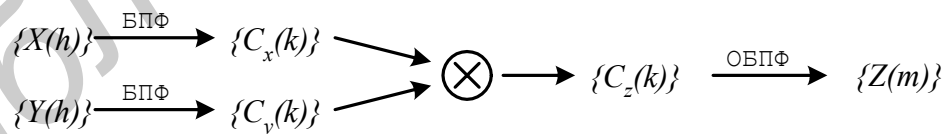


Рис. 1.3. Схема вычисления свертки

Задание

1. Внимательно ознакомиться с теоретической частью.
2. Изобразить график функции $\cos(x)$.
3. Изобразить график функции $\sin(x)$.
4. Написать программу реализации одномерной свертки на основе уже реализованных предварительно функций прямого и обратного БПФ.

5. Написать программу определения корреляции двух функций на основе БПФ.
6. Воспользовавшись графиками функций в качестве исходных данных, получить их свертку и корреляцию.
7. Построить графики в соответствии с полученными результатами.
8. Сделать соответствующие выводы.

Контрольные вопросы

1. Для чего используются понятия «корреляция» и «свертка» в такой области, как цифровая обработка сигналов и изображений?
2. Привести описание операции «свертка».
3. Привести описание операции «корреляция».
4. Каким образом связаны понятия «свертка», «корреляция» и БПФ?
5. Привести теорему свертки.
6. Привести теорему корреляции.

Лабораторная работа № 1.3

Изучение преобразования Уолша

Теоретическая часть

В 1923 году американский ученый Уолш получил полную систему ортонормированных функций, которая дополняет систему функций Радемахера. Множество функций Уолша обычно разделяется на три группы, отличающиеся порядком расположения в системе.

Общеприняты следующие упорядочения:

- 1) упорядочение по частоте (по Уолшу);
- 2) упорядочение по Пэли (диадическое);
- 3) упорядочение по Адамару, где под частотой функции понимается число пересечений нулевого уровня в единицу времени.

Функции Уолша определяются как $Wal_W(i, t)$.

Если через S_i обозначить частоту функции $Wal_W(i, t)$, то

$$S_i = \begin{cases} 0, & i = 0, \\ i/2, & i = \text{четное}, \\ (i+1)/2, & i = \text{нечетное}, \end{cases}$$

причем $Wal_W(J, \theta) = \prod_{v=1}^n [rad(v, \theta)]^{J_v}$. Это соотношение справедливо, когда соот-

ветствующие 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.

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

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

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

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

$$0 \oplus 0 = 0$$

$$1 \oplus 0 = 1$$

$$0 \oplus 1 = 1$$

$$1 \oplus 1 = 0.$$

Например, код Грея, соответствующий двоичному числу 101101, может быть образован как на рис. 1.4. Трехразрядный код Грея показан в табл. 1.2.

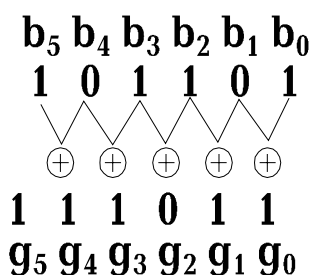


Рис. 1.4. Преобразование двоичного кода в код Грея

Таблица 1.2

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, соответствующего коду Грея 1001011, показан на рис. 1.5.

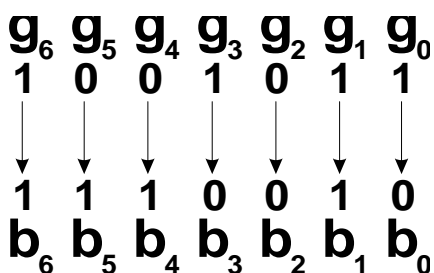


Рис. 1.5. Преобразование кода Грея в двоичный код

Функции Радемахера (рис. 1.6) представляют собой неполную систему ортонормированных функций. Функция Радемахера определенной частоты с индексом m , обозначаемая $rad(m,t)$, имеет последовательности прямоугольных импульсов и содержит 2^{m-1} периодов на полуоткрытом интервале $[0,1)$, принимая значения $+1$ и -1 . Исключение составляет функция $rad(0,t)$, которая имеет

вид единичного импульса. Функции Радемахера — периодические с периодом 1, т.е. $rad(m,t)=rad(m,t+1)$, и могут быть получены с помощью рекуррентного соотношения

$$rad(m,t)=rad(1,2^{m-1}t),$$

$$\text{где } rad(1,t) = \begin{cases} 1, & t \in [0, 1/2), \\ -1, & t \in [1/2, 1). \end{cases}$$

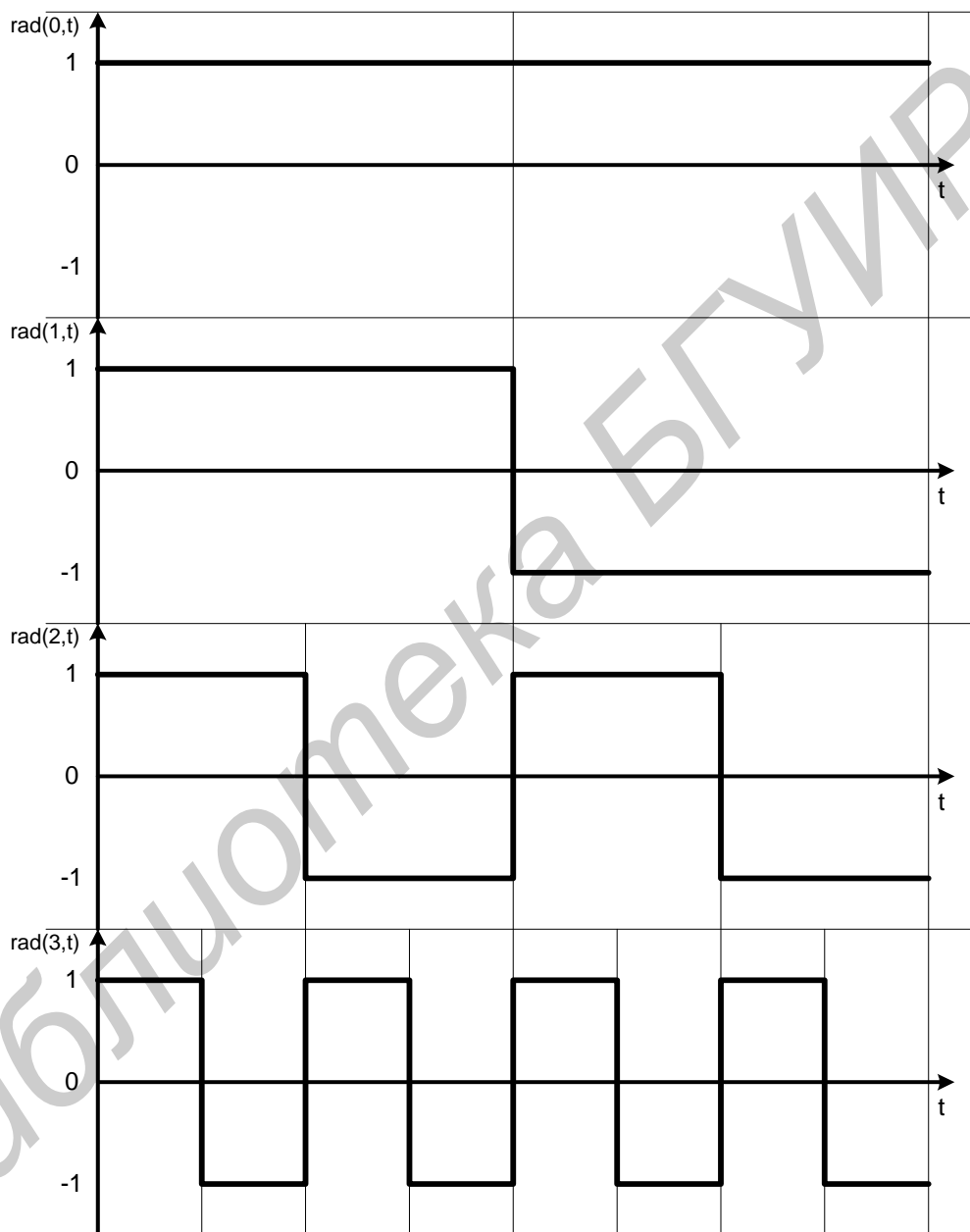


Рис. 1.6. Функции Радемахера для $N=8$

Обозначив для краткости $rad(m,t)=r_m(t)$, для $N=8$ будем иметь:

если $r_1(t) = ++++----$

$r_2(t) = ++--++--$

$r_3(t) = +-+-+--+$,

где «+» соответствует +1, а «-» соответствует -1,

то по системе Уолша (упорядочение по частоте следования)

$$\begin{aligned}
 W(0,t) &= +++++++ \\
 W(1,t) &= r_1 = ++++---- \\
 W(2,t) &= r_1 r_2 = ++----++ \\
 W(3,t) &= r_2 = +- - - + + - - \\
 W(4,t) &= r_2 r_3 = + - - + + - - + \\
 W(5,t) &= r_1 r_2 r_3 = + - - + - + + - \\
 W(6,t) &= r_1 r_3 = + - + - - + - + \\
 W(7,t) &= r_3 = + - + - + - + -
 \end{aligned}$$

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.

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

В свою очередь элементы матрицы Адамара можно получить из соотношения

$$h_{uv} = (-1)^{\sum_{i=0}^{n-1} u_i v_i} \quad u, v = 0, 1, \dots, N-1.$$

Свойства системы Уолша:

1. Ортогональность

Функции Уолша являются ортогональными на интервале $[0, 1]$

$$\int_0^1 Wal_\alpha(t) Wal_\beta(t) dt = \begin{cases} 0, & \alpha \neq \beta, \\ 1, & \alpha = \beta. \end{cases}$$

2. Любая функция Уолша есть периодическая с периодом 1:

$$Wal_\alpha(t) = Wal_\alpha(t + 1).$$

3. Так как функции Уолша принимают только значения +1 и -1, то модуль функции Уолша равен 1 $|Wal_\alpha(t)| = 1$, а среднее значение функции Уолша для всех $\alpha \neq 0$ равно нулю

$$\int_0^1 Wal_\alpha(t) dt = 0, \quad \text{где } \alpha \neq 0.$$

4. Мультипликативность

Произведение двух функций Уолша всегда дает новую функцию Уолша из этой же системы

$$Wal_{\alpha}(t)Wal_{\beta}(t) = Wal_{\gamma}(t), \text{ где } \gamma = \alpha \oplus \beta.$$

Пару преобразований Уолша можно записать в виде

$$x(t) = \sum_{k=0}^{N-1} C_k Wal(k, t);$$

$$C_k = \frac{1}{N} \sum_{t=0}^{N-1} x(t) Wal(k, t).$$

Быстрое преобразование Уолша можно получить также с помощью разбиения матриц. Графическая схема алгоритма показана на рис. 1.7. 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

$$\begin{bmatrix} C_x(0) \\ C_x(1) \\ C_x(2) \\ C_x(3) \end{bmatrix} = \frac{1}{8} H(2) \begin{bmatrix} x_1(0) \\ x_1(1) \\ x_1(2) \\ x_1(3) \end{bmatrix} \text{ и } \begin{bmatrix} C_x(4) \\ C_x(5) \\ C_x(6) \\ C_x(7) \end{bmatrix} = \frac{1}{8} H(2) \begin{bmatrix} x_1(4) \\ x_1(5) \\ x_1(6) \\ x_1(7) \end{bmatrix},$$

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

$$\begin{bmatrix} C_x(0) \\ C_x(1) \\ C_x(2) \\ C_x(3) \end{bmatrix} = \frac{1}{8} \begin{bmatrix} H(1) & H(1) \\ H(1) & -H(1) \end{bmatrix} \cdot \begin{bmatrix} x_1(0) \\ x_1(1) \\ x_1(2) \\ x_1(3) \end{bmatrix} \quad \text{и}$$

$$\begin{bmatrix} C_x(4) \\ C_x(5) \\ C_x(6) \\ C_x(7) \end{bmatrix} = \frac{1}{8} \begin{bmatrix} H(1) & H(1) \\ H(1) & -H(1) \end{bmatrix} \cdot \begin{bmatrix} x_1(4) \\ x_1(5) \\ x_1(6) \\ x_1(7) \end{bmatrix}.$$

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 $N \log_2 N$.

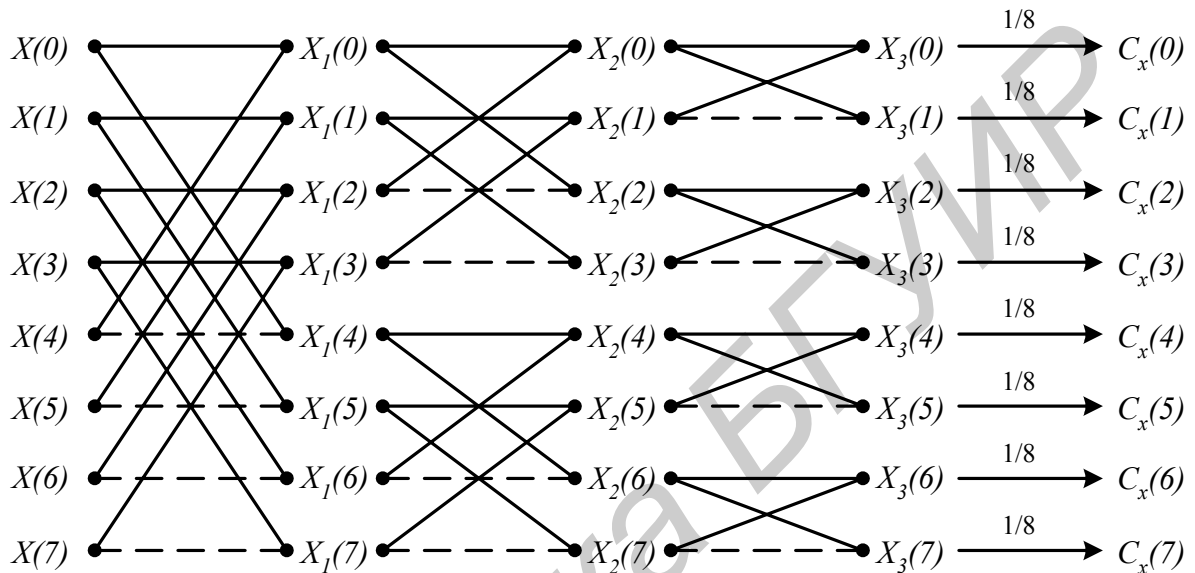


Рис. 1.7. Процедура быстрого алгоритма Уолша

Задание

1. Внимательно ознакомьтесь с теоретической частью.
2. Удостоверьтесь в правильности работы преобразования Уолша.
3. По имеющимся значениям величин постройте графики для входных функций $\sin(x)$, $\cos(x)$.
4. Постройте базовые функции для $N=8$.
5. Убедитесь в правильности работы преобразования Уолша и докажите, каким образом вы это установили, исходя из базовых составляющих.
6. Постройте матрицу преобразования для N -отсчетов и опять покажите правильность работы преобразования Уолша.

Контрольные вопросы

1. Для чего используются ортогональные преобразования?
2. Доказать, что базис Уолша является ортогональным.
3. Дать определение преобразованию Уолша.
4. Каковы основные свойства преобразования Уолша?
5. Каким образом осуществляется быстрое преобразование Уолша?
6. В чем заключается преимущество быстрого преобразования Уолша?

Лабораторная работа № 1.4

Изучение преобразования Хаара

Теоретическая часть

Множество функций Хаара $har(n, m, t)$ (рис. 1.8), 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 $0 \leq r < \log_2 N$, $1 \leq m \leq 2^r$.

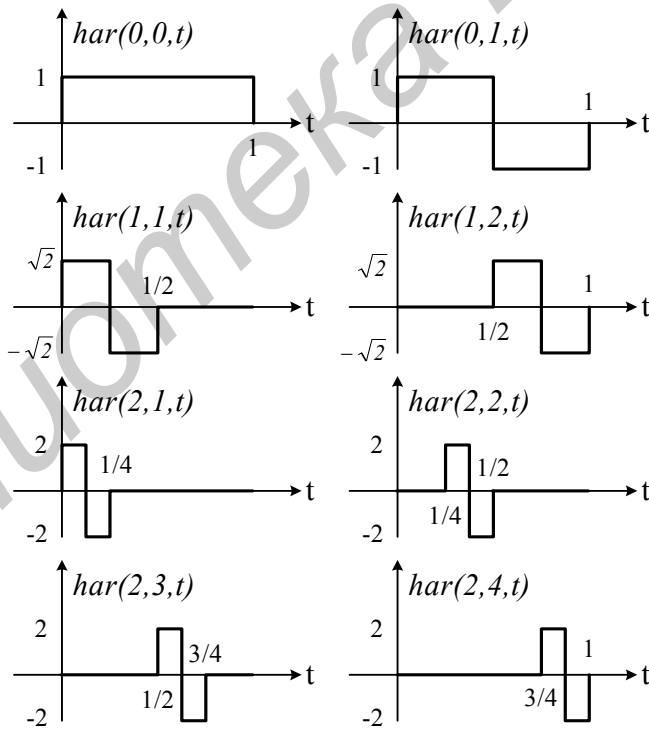


Рис. 1.8. Функции Хаара для $N=8$

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

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

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.

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 номеров столбцов 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

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

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. $\{0,1\} \rightarrow \{1,0\}$, что приводит к матрице, совпадающей с $H_2^*(3)$.

Графические схемы прямого и обратного быстрого преобразования Хаара представлены соответственно на рис 1.9 и 1.10.

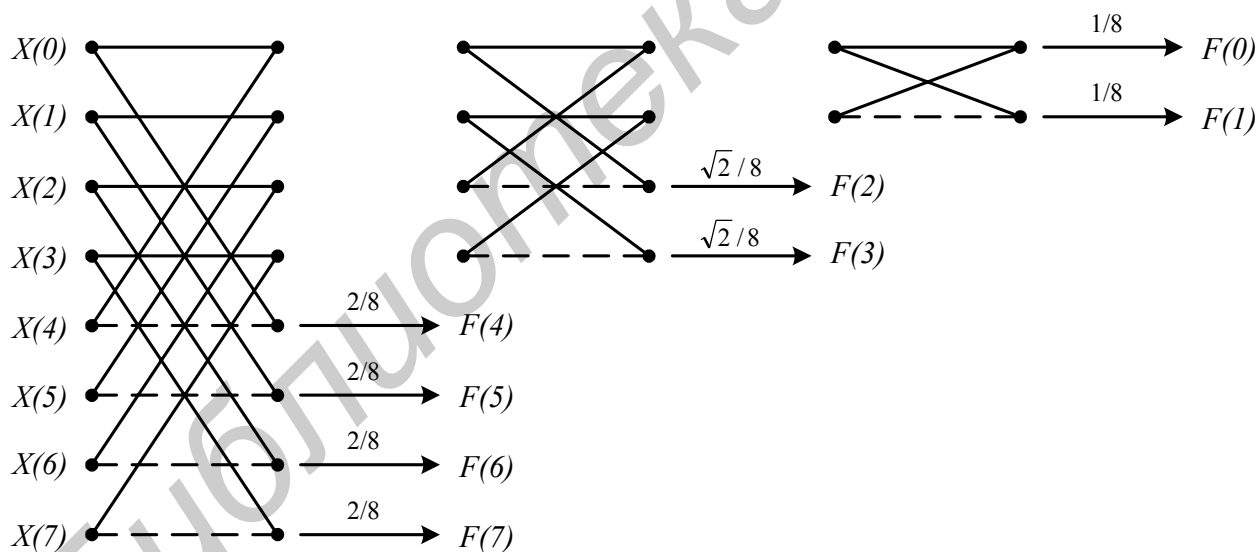


Рис. 1.9. Граф-схема прямого быстрого преобразования Хаара

Тогда спектр можно определить как

$$\begin{aligned} P_h(0) &= F^2(0); \\ P_h(1) &= F^2(1); \\ P_h(2) &= \sum_{k=2}^3 F^2(k); \end{aligned}$$

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

В общем виде можно записать

$$P_h(0) = F^2(0);$$

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

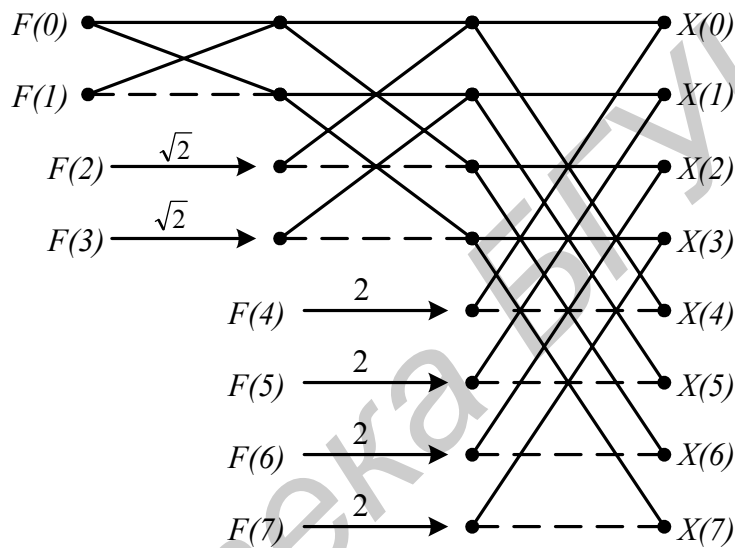


Рис. 1.10. Граф-схема быстрого обратного преобразования Хаара

Задание

1. Внимательно ознакомьтесь с теоретической частью.
2. Разберитесь в работе алгоритма Кули-Тьюки.
3. Постройте графики для входных произвольных прямоугольных функций.
4. Постройте базовые функции для $N=8$.
5. Убедитесь в правильности работы преобразования Хаара и докажите, каким образом вы это установили, исходя из базовых составляющих.
6. Построить матрицу преобразования для N -отсчетов и опять показать правильность работы преобразования.

Контрольные вопросы

1. Для чего используются ортогональные преобразования?
2. Доказать, что базис Хаара является ортогональным.
3. Дать определение преобразования Хаара.

4. Каковы основные свойства преобразования Хаара?
5. Каким образом осуществляется быстрое преобразование Хаара?
6. В чем заключается преимущество быстрого преобразования Хаара?

Литература

1. Ярославский Л.П. Введение в цифровую обработку изображений. — М.: Сов. радио, 1979.
2. Садыхов Р.Х., Чеголин П.М., Шмерко В.П. Методы и средства обработки сигналов в дискретных базисах. — Мн.: Наука и техника, 1987.
3. Прэрт У. Цифровая обработка изображений. — М.: Мир, 1982.
4. Даджион Д., Мерсеро Р. Цифровая обработка многомерных сигналов. — М.: Мир, 1988.
5. Беллман Р. Введение в теорию матриц. — М.: Наука, 1976.
6. Рабинер Л., Гоулд Б. Теория и применение цифровой обработки сигналов. — М.: Мир, 1978.
7. Абламейко С.В., Лагуновский Л.М. Обработка изображений: технология, методы, применение. — Мн.: Институт технической кибернетики НАНБ, 1999.
8. Садыхов Р.Х., Мачнев А.Г. Системные процессоры цифровой обработки изображений в двумерных базисах. — Мн.: Институт технической кибернетики НАНБ, 1996.
9. Эндрюс Г. Применение вычислительных машин для обработки изображений. — М.: Наука, 1977.
10. Залманзон Л.А. Преобразования Фурье, Уолша, Хаара и их применение в управлении, связи и других областях. — М.: Наука, 1989.

2. НЕЙРОННЫЕ СЕТИ

Общие теоретические сведения

Основным элементом нейронной сети является нейрон, который осуществляет операцию нелинейного преобразования суммы произведений входных сигналов на весовые коэффициенты:

$$y = f\left(\sum_{i=1}^n w_i x_i + T\right), \quad (2.1)$$

где $X=(x_1, x_2, \dots, x_n)$ — вектор входного сигнала; $W=(w_1, w_2, \dots, w_n)$ — весовой вектор; T — порог; f — так называемая функция активации.

Схема нейронного элемента изображена на рис 2.1. Каждому i -му входу нейрона соответствует весовой коэффициент w_i (синапс), который характеризует силу синаптической связи по аналогии с биологическим нейроном.

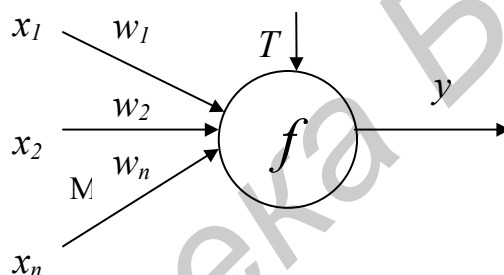


Рис. 2.1. Нейрон

Весовой вектор W , порог T и пороговая функция f определяют поведение любого нейрона — его реакцию на входные данные. Величина веса w_i определяет степень влияния входа i на выход нейрона, а знак — характер влияния. Каждый из входов связан с некоторым источником информации (рецептор, формирующий признаки, распределительная ячейка, выход другого нейрона). Положительные веса характерны для возбуждающих связей, способствующих повышению активности нейрона. Отрицательные, как правило, соответствуют тормозящим связям.

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 [8] вводят дополнительный вход нейрона x_0 , всегда равный 1, и обозначают порог как его вес w_0 . Это позволяет упростить выражение (2.1) 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 [1,7]. 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. 2.2)

$$y = \frac{1}{(1 + e^{-x})}; \quad (2.2)$$

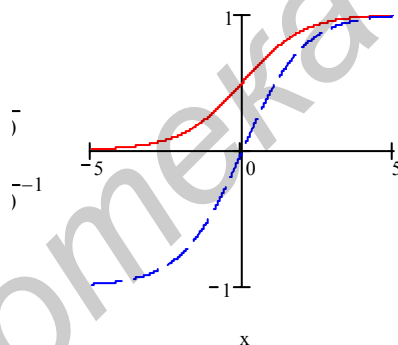


Рис. 2.2. Сигмоидные функции активации

5) биполярная сигмоидная функция (см. рис. 2.2)

$$y = \frac{2}{(1 + e^{-x})} - 1; \quad (2.3)$$

6) гиперболический тангенс (рис. 2.3)

$$y = th(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}}. \quad (2.4)$$

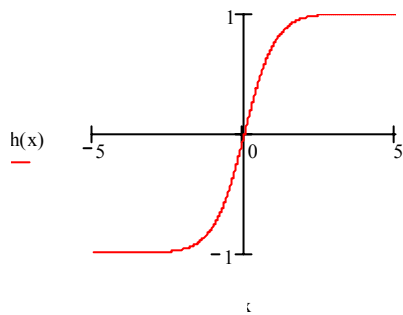


Рис. 2.3. Гиперболический тангенс

Для классификаторов чаще всего применяют функции (2.2)–(2.4), поскольку они нелинейные и хорошо дифференцируются.

Веса и порог конкретного нейрона являются настраиваемыми параметрами. Они содержат знания нейрона, определяющие его поведение. 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. 2.4).

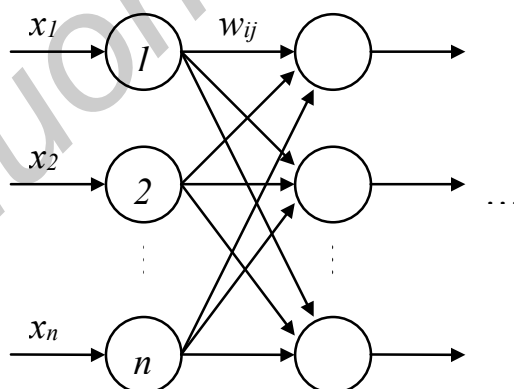


Рис. 2.4. Распределительный слой

У любой нейронной сети можно выделить 2 режима работы: обучение и воспроизведение. На этапе обучения настраиваются веса и пороги всех слоев. Воспроизведение является этапом обработки информации, следующим за обучением, при этом веса и пороги, как правило, не изменяются. Большое влияние на функционирование сети оказывает ее топология — архитектура слоев и связей между нейронами.

На способе обработки информации решающим образом сказывается наличие или отсутствие в сети обратных связей. Если обратных связей нет (каждый нейрон получает информацию только от нейронов предыдущих слоев), то обработка информации происходит в одном направлении за число тактов, равное числу слоев. При наличии обратных связей информация может проходить через сеть много раз до достижения какого-либо условия. В случае, если условие достигнуто, говорят, что сеть стабилизировалась. В общем случае сходимость не гарантируется. Тем не менее наличие обратных связей позволяет решать задачи с привлечением меньшего числа нейронов, что ускоряет процесс обучения.

Существуют следующие разновидности способов обучения нейронных сетей:

1. Обучение с учителем. В этом случае для каждого входного вектора x^k задан идеальный выходной вектор y^k , т.е. обучающая выборка задается парами векторов $\{x^k, y^k\}$, определяющими вход-выход. Задача сети в таких условиях — аппроксимировать таблично заданную функцию. Такое обучение широко используют в классификаторах, при этом x^k представляет собой входной вектор признаков, а y^k кодирует номер класса.

2. Обучение с подкреплением. Используется в случае, когда информация о выходах известна не полностью. Например, вместо эталонных ответов известно лишь, хуже или лучше данная конфигурация сети справляется с задачей. Цель — научить сеть правильной линии поведения.

3. Обучение без учителя. Сети предлагается самостоятельно найти скрытые закономерности в массиве векторов x^k и найти более компактное его представление.

В табл. 2.1 приведены примеры различных классов нейронных сетей и решаемые с их помощью задачи.

Таблица 2.1

Способ обработки информации	Обучение с учителем	Обучение без учителя
Без обратных связей	Многослойный персептрон, сеть РБФ (аппроксимация функций, классификация)	Конкурентные сети, сети Кохонена (сжатие данных, выделение признаков)
С обратными связями	Рекуррентные аппроксиматоры (предсказание временных рядов, самообучение)	Сеть Хопфилда (ассоциативная память)

Лабораторная работа № 2.1

Сеть Хопфилда

Краткие теоретические сведения

Сеть Хопфилда — однослойная, симметричная, нелинейная, автоассоциативная нейронная сеть, которая запоминает бинарные / биполярные образы. Сеть характеризуется наличием обратных связей. Топология сети Хопфилда показана на рис. 2.5. 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 .

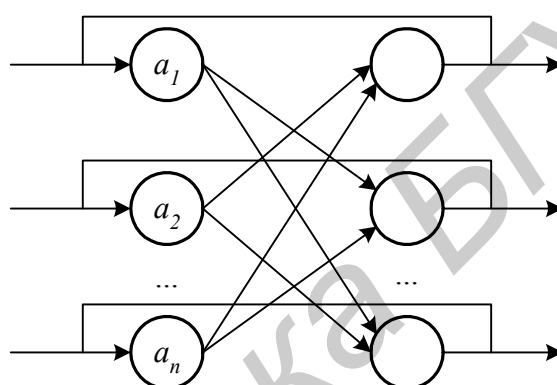


Рис. 2.5. Топология сети Хопфилда

Обучение. Обучение сети осуществляется в соответствии с соотношением

$$w_{ij} = \begin{cases} \sum_{k=1}^m a_i^k a_j^k, & i \neq j, \text{ для } i, j = \overline{1, n}, \\ 0, & i = j \end{cases} \quad (2.5)$$

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

$$W = \begin{bmatrix} w_{11} & w_{12} & \dots & w_{1n} \\ w_{21} & w_{22} & \dots & w_{2n} \\ \dots & \dots & \dots & \dots \\ w_{n1} & w_{n2} & \dots & w_{nn} \end{bmatrix}.$$

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 функция, 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

$$a_i(t+1) = f \left(\sum_{j=1}^n w_{ij} a_j(t) \right), \quad (2.6)$$

где $a_j(t)$ — выход j -го нейрона в момент времени t , f — бинарная / биполярная функция активации;

$$f(x) = \begin{cases} 1, & x > 0, \\ -1, & x \leq 0. \end{cases} \quad (2.7)$$

При работе в синхронном режиме на один такт работы сети все нейроны одновременно меняют состояние по формуле (2.7). 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 (2.7) 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

$$m = \frac{n}{2 \ln n + \ln \ln n},$$

где n — количество нейронов, что следует отнести к недостаткам этой сети.

Задание

1. Произвести обучение сети Хопфилда на заданный тип образов. Рекомендуемое количество нейронов сети не менее 32 (для запоминания 3 образов). Примеры типов образов: шестнадцатеричные числа из 8 цифр, слова из 4 букв, бинарные изображения размером 6×6.

2. Подать на вход сети ряд тестовых образов. Вывести полученные результаты в заданном виде.

3. Убедиться, что сеть сходится к ближайшему образу из обучающей выборки.

4. Добиться некорректного функционирования и объяснить его причины.

Контрольные вопросы

1. Топология сети Хопфилда.
2. Обучение сети Хопфилда.
3. Процесс воспроизведения информации в сети Хопфилда.
4. Зависимость максимального количества образов, запоминаемых сетью, от ее размера.
5. В чем причина некорректной работы при запоминании подобных образов?
6. Варианты использования сети Хопфилда.

Лабораторная работа № 2.2

Многослойный персептрон

Краткие теоретические сведения

Многослойный персептрон 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 [1]. Для этого достаточно одного скрытого слоя нейронов с сигмоидной функцией активации (2.2)–(2.4), т.е. многослойный персептрон обычно состоит из 3 слоев: первого распределительного, второго скрытого и третьего выходного (рис. 2.6).

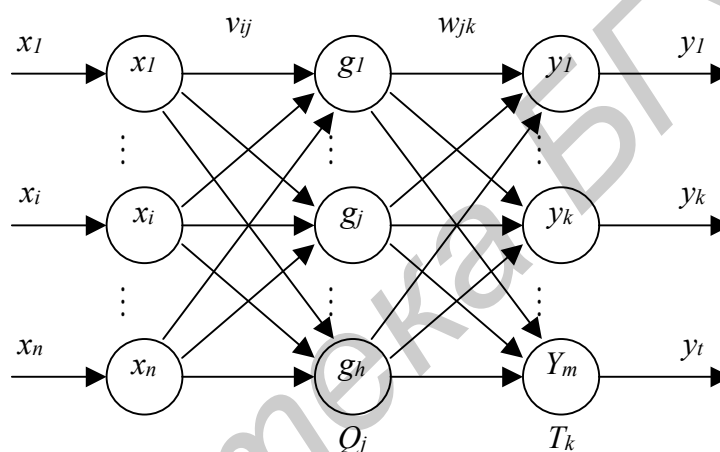


Рис. 2.6. Многослойный персептрон

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. Персептрон функционирует по следующим формулам:

$$g_j = f\left(\sum_{i=1}^n v_{ij}x_i + Q_j\right), \quad (2.9)$$

$$y_k = f\left(\sum_{j=1}^h w_{jk}g_j + T_k\right). \quad (2.10)$$

В качестве функции активации используется одна из функций (2.2)–(2.4). Вид функции определяет диапазон чисел, в котором работает сеть. В дальнейшем будет использоваться сигмоидная функция (2.2), имеющая область значений от 0 до 1.

Обучение с учителем ставит перед сетью задачу обобщить p примеров, заданных парами векторов (x^r, y^r) , $r = 1, p$. Вектор $x^r = (x_1^r, x_2^r, \dots, x_n^r)$ в случае задачи классификации задает входной образ (вектор признаков), а вектор $y^r = (y_1^r, y_2^r, \dots, y_m^r)$, 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 персептрона проводится с помощью алгоритма обратного распространения ошибки [6], 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. На первом этапе происходит начальная инициализация знаний сети. Простейший вариант такой инициализации — присвоить всем весам и порогам случайные значения из диапазона $[-1, 1]$.

2. Далее 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 (2.9), (2.10).

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:

$$w_{jk}(t+1) = w_{jk}(t) + \alpha \frac{\partial E}{\partial w_{jk}(t)}, \quad (2.11)$$

$$T_k(t+1) = T_k(t) + \alpha \frac{\partial E}{\partial T_k(t)}, \quad (2.12)$$

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

$$E = \frac{1}{2} \sum_{k=1}^m (y_k^r - y_k)^2. \quad (2.13)$$

Ошибка k -го нейрона выходного слоя определяется как

$$d_k = \frac{\partial E}{\partial y_k} = y_k^r - y_k. \quad (2.14)$$

Выразим производные из формул (2.11), (2.12) через легко вычисляемые величины. Определим взвешенную сумму, аргумент функции активации как

$$S_k = \sum_{j=1}^h w_{jk} g_j + T_k. \quad (2.15)$$

Можно представить:

$$\frac{\partial E}{\partial w_{jk}} = \frac{\partial E}{\partial y_k} \frac{\partial y_k}{\partial S_k} \frac{\partial S_k}{\partial w_{jk}}, \quad (2.16)$$

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

$$\frac{\partial E}{\partial w_{jk}} = d_k f'(S_k) g_j. \quad (2.17)$$

Аналогично (2.16), с учетом того, что $\frac{\partial S_k}{\partial T_k} = 1$, получаем

$$\frac{\partial E}{\partial T_k} = \frac{\partial E}{\partial y_k} \frac{\partial y_k}{\partial S_k} \frac{\partial S_k}{\partial T_k} = d_k f'(S_k). \quad (2.18)$$

Веса и пороги скрытого слоя также корректируются по формулам, аналогичным (2.11), (2.12), с учетом (2.17) и (2.18). При этом главной трудностью является определение ошибки нейрона скрытого слоя. Эту ошибку явно опреде-

лить по формуле, аналогичной (2.14), нельзя, однако существует возможность рассчитать ее через ошибки нейронов выходного слоя (отсюда произошло название алгоритма обратного распространения ошибки):

$$e_j = \frac{\partial E}{\partial g_j} = \sum_{k=1}^m \frac{\partial E}{\partial y_k} \frac{\partial y_k}{\partial S_k} \frac{\partial S_k}{\partial g_j} = \sum_{k=1}^m d_k f'(S_k) w_{jk}. \quad (2.19)$$

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

$$f'(S_k) = \left(\frac{1}{1 + e^{-S}} \right)' = f(S_k)(1 - f(S_k)) = y_k(1 - y_k). \quad (2.20)$$

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 (2.11), (2.12) для двух слоев, использующих сигмоидную функцию:

$$w_{jk} = w_{jk} + \alpha y_k (1 - y_k) d_k g_j, \quad (2.21)$$

$$T_k = T_k + \alpha y_k (1 - y_k) d_k, \quad (2.22)$$

$$v_{ij} = v_{ij} + \beta g_j (1 - g_j) e_j x_i, \quad (2.23)$$

$$Q_j = Q_j + \beta g_j (1 - g_j) e_j. \quad (2.24)$$

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 2 алгоритма обучения [8], 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 крите-

рия максимальную по модулю ошибку на выходе d_k , полученную на шаге 2. Условием прекращения обучения в этом случае будет

$$\max |d_k| < D, k = \overline{1, m}, r = \overline{1, p}, \quad (2.25)$$

где D — достаточно маленькая константа — величина максимальной ошибки, которую требуется достичь в процессе обучения. Если условие (2.25) 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 [8].

Оптимальное соотношение между этими параметрами оценивают как

$$h \sim \sqrt{\frac{p}{n}}. \quad (2.26)$$

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 (2.13), а гарантируется определение только локального минимума. Проблемы, возникающие в процессе градиентного спуска, можно проанализировать на примере функции ошибки, схематически изображенной на рис. 2.7.

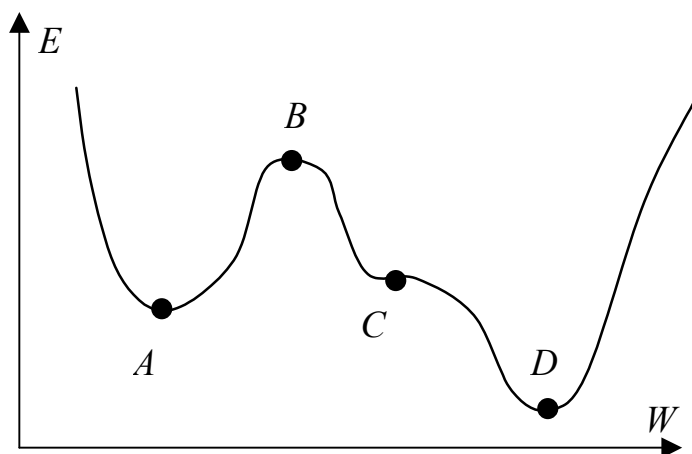


Рис. 2.7. Схематическая функция ошибки сети

На рис. 2.7 показаны четыре критические точки, производная функции ошибки в которых близка к нулю. Точка *A* соответствует локальному минимуму. Признаком достижения локального минимума в процессе обучения является полное прекращение уменьшения ошибки (2.25). В этом случае может помочь повторное обучение с другим начальным распределением знаний (тут может помочь случайная инициализация). Точка *B* — локальный максимум. В случае попадания в окрестность такой точки скорость резко падает, затем снова быстро растет. Не существует способа предсказать, в какую сторону лучше двигаться из точки с производной, близкой к нулю. Точка *C* — точка перегиба, характеризуется длительным уменьшением скорости. Точка *D* — глобальный минимум — цель алгоритма.

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 (2.11) в момент времени t будет вычисляться как

$$\Delta w(t) = \alpha \frac{\partial E}{\partial w} + \mu \Delta w(t-1), \quad (2.27)$$

где μ — параметр, определяющий величину влияния момента. С использованием (2.27) 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 [7,8] позволяют улучшить работу сети в условиях плохой сходимости. Однако они услож-

няют 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.

Задание

1. Обучить трехслойный персептрон на заданное количество классов. В обучающей выборке каждый класс может быть представлен одним идеальным образом заданного типа. Векторы y^r могут состоять из всех нулей и одной единицы, положение которой определяет номер класса. В такой конфигурации номер класса будет определяться номером самого активного нейрона на выходе, а степень уверенности — величиной на данном выходе. Вывести количество шагов, затраченных на обучение.

2. Подать на вход тестовые образы и убедиться, что сеть корректно разбивает их на классы.

3. Поэкспериментировать с константами α , β . В первом приближении их можно взять равными 1, затем постепенно увеличивать, достигая большей скорости обучения.

Контрольные вопросы

1. Топология многослойного персептрона.
2. Процесс обучения.
3. Процесс воспроизведения.
4. Процедура обратного распространения ошибки.
5. Каким образом можно улучшить качество распознавания?
6. Достоинства и недостатки данного типа нейронной сети.

Лабораторная работа № 2.3

Сеть РБФ

Краткие теоретические сведения

Сеть РБФ (радиальная базисная функция) является аналогом многослойного персептрона (рис. 2.8). 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.

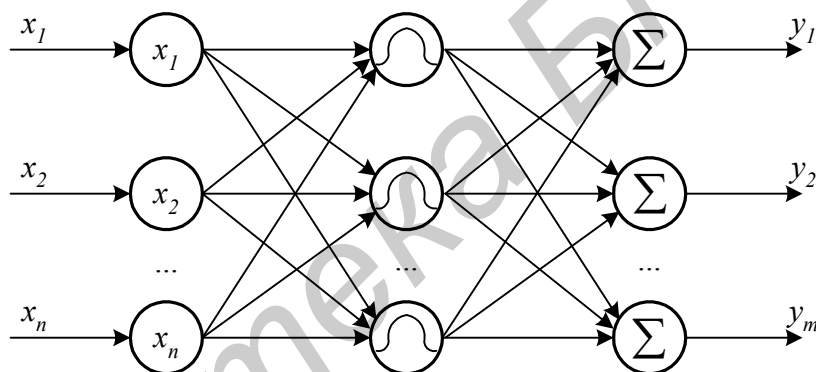


Рис. 2.8. Топология сети РБФ

Упрощение работы и обучения достигается за счет введения вместо скрытого слоя нейронов слоя РБФ ячеек. Классический закон, по которому такая ячейка функционирует, определяется формулой гауссова колокола

$$g_j = \exp\left(\frac{-|x - t^j|^2}{\sigma_j^2}\right), \quad (2.28)$$

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. 2.9). 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.

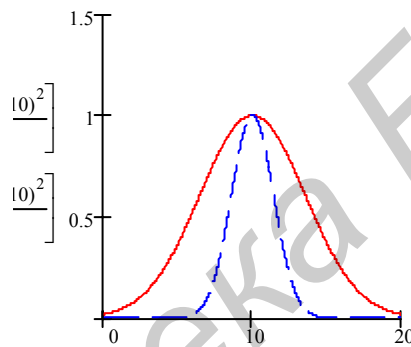


Рис. 2.9. Примеры функций РБФ ячеек с одинаковым центром и разным разбросом

Выходной слой РБФ сети обычно состоит из суммирующих ячеек

$$y_k = \sum_{j=1}^h w_{jk} g_j. \quad (2.29)$$

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 2.2 значительно упрощаются. Формулы (2.9) и (2.10) заменяются соответственно на (2.28) и (2.29). Формула (2.21) упрощается до

$$w_{jk} = w_{jk} + \alpha d_k g_j, \quad (2.30)$$

поскольку функция активации в выходном слое сети РБФ линейная и ее производная равна 1. Для коррекции весовых коэффициентов выходных суммирующих ячеек используется только (2.30).

Контроль завершения алгоритма обучения производится аналогично.

Воспроизведение. Сеть функционирует по формулам (2.28) и (2.29).

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.

Задание

1. Обучить сеть РБФ на заданное количество классов аналогично многослойному персептрону. Вывести количество шагов, затраченных на обучение.
2. Подать на вход тестовые образы, убедиться, что сеть корректно разбивает их на классы.
3. Сделать вывод о скорости обучения сети РБФ по сравнению с многослойным персептроном.

Контрольные вопросы

1. Топология сети РБФ.
2. Обучение сети, выбор количества РБФ ячеек.
3. Процесс воспроизведения информации в сети РБФ.
4. Каким образом можно улучшить работу сети?
5. Достоинства и недостатки данного типа нейронной сети.

Лабораторная работа № 2.4

Конкурентная нейронная сеть

Краткие теоретические сведения

Самоорганизующиеся нейронные сети обучаются без учителя. Они способны адаптироваться к входным данным, используя содержащиеся в этих данных зависимости. Такие сети используются для нахождения более компактного описания данных (сжатия), кластеризации, выделения признаков.

Конкурентная сеть является простейшей самоорганизующейся нейронной сетью (рис. 2.10).

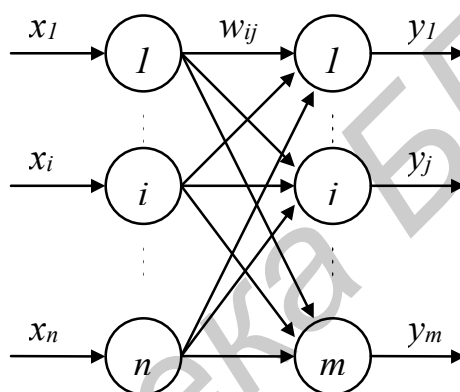


Рис. 2.10. Конкурентная нейронная сеть

Первый слой является распределительным. Нейроны второго слоя функционируют по формуле

$$y_j = \sum_{i=1}^n w_{ij} x_i = |w^j| |x| \cos \alpha, \quad (2.31)$$

где $x = (x_1, x_2, \dots, x_i, \dots, x_n)$ — входной вектор; $w^j = (w_{1j}, w_{2j}, \dots, w_{ij}, \dots, w_{nj})$ — вектор весовых коэффициентов нейрона, а $|x|$ и $|w^j|$ — их модули, α — угол между ними.

Обучение. При обучении нейронной сети при подаче каждого входного вектора определяется нейрон-победитель, для которого (2.31) максимально. Для этого нейрона синаптические связи усиливаются по формуле

$$w_{ij}(t+1) = w_{ij}(t) + \beta(x_i - w_{ij}(t)), \quad (2.32)$$

где β — скорость обучения.

Смысл этой формулы в том, что вектор весовых коэффициентов нейрона — победителя «поворачивается» в сторону входного вектора, тем самым актив-

ность нейрона усиливается. Удобно работать с нормированными входными и весовыми векторами, когда их модуль равен 1. Нормировка уравнивает шансы в конкуренции нейронов с разным модулем вектора весовых коэффициентов. Выражение (2.31) для нормированных векторов будет выглядеть как

$$y_j = \sum_{i=1}^n w_{ij} x_i = \cos \alpha, \quad (2.33)$$

а выражение (2.32):

$$w_{ij}(t+1) = \frac{w_{ij}(t) + \beta(x_i - w_{ij}(t))}{|w^j(t) + \beta(x - w^j(t))|}. \quad (2.34)$$

Случайное начальное распределение весовых коэффициентов может привести к тому, что некоторые нейроны никогда не станут победителями, так как их весовые векторы окажутся удаленными от всех входных векторов. Существует ряд модификаций алгоритма обучения, позволяющих устранить этот недостаток. Хорошие результаты на практике показало частотно-зависимое конкурентное обучение. Согласно нему, нейрон-победитель определяется по минимуму произведения евклидова расстояния между входным и весовым вектором и количеством побед данного нейрона f_j :

$$d_v = \min_j (|x - w^j| f_j). \quad (2.35)$$

Шансы нейрона на победу уменьшаются с количеством побед, что дает преимущество другим нейронам.

Конкурентное обучение продолжается до тех пор, пока максимум евклидова расстояния между любым входным вектором и соответствующим ему вектором весов нейрона-победителя не достигнет заданного малого значения.

Конкурентная сеть позволяет разбить входную выборку нормированных векторов на m (количество выходных нейронов сети) кластеров, расположенных на поверхности гиперсферы в пространстве признаков единичного радиуса. Входные векторы, приводящие к победе одного и того же нейрона, относят к одному кластеру.

Кохонен предложил внести в правило конкурентного обучения (2.32) информацию о расположении нейронов в выходном слое. Для этого нейроны упорядочиваются в одномерные или двухмерные решетки. Вводится функция, корректирующая изменение весов в зависимости от расстояния до нейрона-победителя $h(t, k, j)$ — сила влияния между нейроном-победителем k и нейроном j в момент времени t . Для $j=k$ эта функция всегда равна 1, и уменьшается с рас-

том расстояния между k и j в решетке. С течением времени радиус влияния обычно сужается. С использованием этой функции веса меняются для всех нейронов сети, а не только для нейрона-победителя:

$$w_{ij}(t+1) = w_{ij}(t) + \beta h(t, k, j)(x_i - w_{ij}(t)). \quad (2.36)$$

В качестве функции $h(t, k, j)$ может использоваться гауссовый колокол (2.28) с параметром σ , зависящим от времени, или функция вида «мексиканской шляпы».

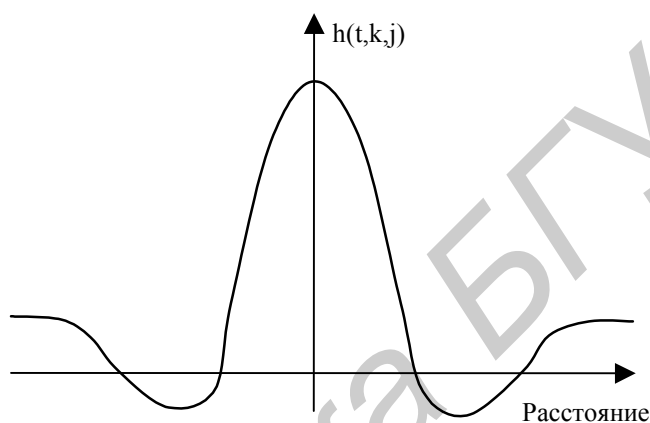


Рис. 2.11. Пример функции расстояния в сетях Кохонена

В результате модификации конкурентного обучения сеть Кохонена не только кластеризует входные примеры, но и упорядочивает их в виде одномерной или двумерной решетки. Это позволяет получить дополнительную информацию о близости кластеров. Если два кластера проецируются на соседние нейроны решетки, это говорит об их близости в исходном пространстве признаков. Обратное неверно. Из-за уменьшения размерности пространства отношение близости сохраняется только для ограниченного числа кластеров.

Задание

1. Обучить конкурентную сеть с использованием правила (2.35) на количество образов, превышающих количество нейронов сети. Рекомендуется использовать нормированные векторы.
2. Убедиться, что похожие образы были спроецированы сетью в один кластер (подача их на вход активизирует один и тот же нейрон).
3. Подать на вход тестовые образы, отличные от образов из обучающей выборки. Сделать выводы.

Контрольные вопросы

1. Смысл самообучения.
2. Обучение конкурентной нейронной сети.
3. Определение нейрона-победителя.
4. Сеть Кохонена.
5. Использование самоорганизующихся сетей.
6. Достоинства и недостатки данного типа нейронной сети.

Литература

1. Bishop C.M. Neural Networks for Pattern Recognition. — Oxford: Clarendon press, 1995. — 482 p.
2. Hopfield J.J. Neural networks and physical systems with emergent collective computational abilities // Proc. Natl. Acad. Sci. USA. — 1982. — Vol. 79. — P. 2554.
3. Kohonen T. Self-organization and associative memory. — Springer-Verlag, 1989. — 312 p.
4. Kohonen T. Self-organized formation of topologically correct feature maps // Biol. Cybernetics. — 1982. — Vol. 43. — P. 56–69.
5. Kohonen T. Self-organizing maps. — Springer-Verlag, 1995. — 362 p.
6. Rumelhart D.E., Hinton G.E., Williams R.J. Learning internal representation by error propagation: McClelland J.L. and Rumelhart D.E. (Eds). Parallel Distributed Processing: Exploration in the Microstructure of Cognition. — MIT Press, Cambridge MA. — 1986. — Vol. 1. — P. 318–362.
7. Головкин В.А. Нейронные сети: обучение, организация и применение. Учеб. пособие для вузов. — М.:ИПРЖР, 2001. — 256 с.
8. Ежов А.А., Шумский С.А. Нейрокомпьютинг и его применение в экономике и бизнесе. — М.: Мир, 1998. — 222 с.

Учебное издание

**Садыхов Рауф Хосровович,
Маленко Олег Григорьевич**

ЛАБОРАТОРНЫЙ ПРАКТИКУМ
по курсам «Цифровая обработка сигналов и изображений»
и «Методы и средства обработки изображений»
для студентов специальностей 40 02 01 и 40 01 01
всех форм обучения

Редактор Н.А. Бебель
Корректор Е.Н. Батурчик
Компьютерная верстка Т.В. Шестакова

Подписано в печать 1/16.		Формат	60x84
Бумага офсетная. Уч.-изд. л. 2,6.	Печать ризографическая. Тираж 100 экз.	Усл. печ. л. Заказ 232.	

Издатель и полиграфическое исполнение:
Учреждение образования «Белорусский государственный университет
информатики и радиоэлектроники»
Лицензия ЛП № 156 от 05.02.2001
Лицензия ЛВ № 509 от 03.08.2001