

Министерство образования Республики Беларусь
Учреждение образования
«Белорусский государственный университет
информатики и радиоэлектроники»

Кафедра электронных вычислительных машин

Р.Х. Садыхов, М.М. Лукашевич

ЛАБОРАТОРНЫЙ ПРАКТИКУМ

по дисциплинам «Цифровая обработка сигналов и изображений»
и «Методы и средства обработки изображений» для студентов
специальностей I-40 02 01 «Вычислительные машины, системы и сети»
и I-40 01 01 «Программное обеспечение информационных технологий»
всех форм обучения

В 2-х частях
Часть 1

Минск 2007

Содержание

ДИСКРЕТНЫЕ ПРЕОБРАЗОВАНИЯ

1. Лабораторная работа №1. Преобразование Фурье.....	4
2. Лабораторная работа №2. Операции корреляции и свертки.....	18
3. Лабораторная работа №3. Преобразование Уолша.....	23
4. Лабораторная работа №4. Вейвлет – преобразование.....	33

ДИСКРЕТНЫЕ ПРЕОБРАЗОВАНИЯ

Лабораторная работа №1 Преобразование Фурье

1. Цель работы

Изучение преобразования Фурье и его основных свойств, а также методики получения быстрого преобразования Фурье (БПФ).

2. Теоретические сведения

Ортогональные функции

Для лучшего понимания вопроса о рядах Фурье дадим определение ортогональным функциям. 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.

Ряд Фурье

Для теории формирования и обработки сигнала особое значение имеет возможность разложения заданного в виде функции сигнала по различным ортогональным системам функций.

Впервые в 1807 году французский математик и физик Жан Батист Жозеф Фурье показал, что любую произвольную функцию $x(t)$ можно представить в виде бесконечной суммы синусных и косинусных членов:

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

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 . В данном случае речь идет о системе ортогональных функций вида $\{1, \cos n\omega_0 t, \sin n\omega_0 t\}$.

Коэффициенты $\{a_0, a_n, b_n\}$ из формулы (1.2) можно вычислить с учетом ортогональности множества функций $\{\cos n\omega_0 t, \sin n\omega_0 t\}$ на периоде T :

$$\int_T \cos n\omega_0 t \cos m\omega_0 t dt = \begin{cases} T/2, & m = n, \\ 0, & m \neq n \end{cases} \quad (1.3)$$

$$\int_T \cos n\omega_0 t \sin m\omega_0 t dt = 0, \forall m, n \quad (1.4)$$

$$\int_T \sin n\omega_0 t \sin m\omega_0 t dt = \begin{cases} T/2, m = n, \\ 0, m \neq n \end{cases} \quad (1.5)$$

С учетом этих соотношений:

$$a_0 = \frac{1}{T} \int_T x(t) dt; \quad (1.6)$$

$$a_n = \frac{2}{T} \int_T x(t) \cos n\omega_0 t dt; \quad (1.7)$$

$$b_n = \frac{2}{T} \int_T x(t) \sin n\omega_0 t dt. \quad (1.8)$$

Раздел математики, устанавливающий соотношение между функцией $x(t)$ и коэффициентами a_n и b_n , называется гармоническим анализом, а представление (1.2) – рядом Фурье.

Компоненты ряда Фурье называются гармониками. Любая четная функция может быть разложена в ряд Фурье, состоящий из косинусов, а любая нечетная функция раскладывается в ряд из синусов. Для некоторых функций ряд Фурье может состоять лишь из нечетных гармоник.

В целом, любая полная система ортогональных функций может быть применена для разложения в ряды, которые соответствуют рядам Фурье. Например, часто используется разложение в ряды по функциям Уолша, Хаара, Лагерра, Бесселя и т.д.

Семейство преобразований Фурье

Преобразование Фурье (Fourier transform) – это разложение функций на синусоиды (далее косинусные функции также будем называть синусоидами, т.к. они отличаются от «настоящих» синусоид только фазой). Анализ Фурье закладывает основы многих методов, применяющихся в цифровой обработке сигналов и изображений (ЦОСиИ). По сути, преобразование Фурье (ПФ) позволяет сопоставить сигналу, заданному во временной области, его эквивалентное представление в частотной области. Обратно, если известна частотная характеристика сигнала, то обратное преобразование Фурье позволяет определить соответствующий сигнал во временной области.

Семейство преобразований Фурье (преобразование Фурье, ряды Фурье, дискретные ряды Фурье и дискретное преобразование Фурье) представлено на рис. 1.1 – 1.4.

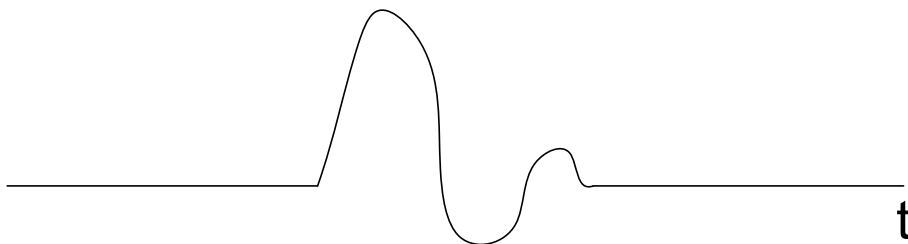


Рис. 1.1. Преобразование Фурье: сигнал непрерывный и аperiodический

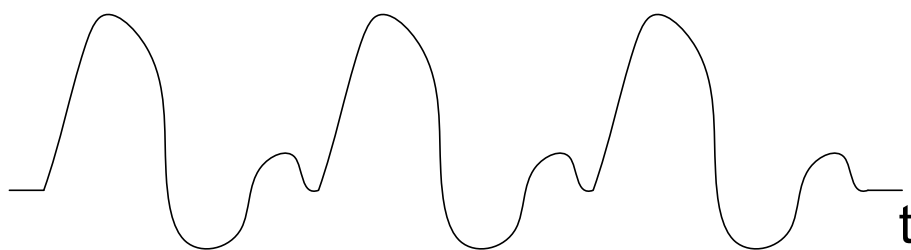


Рис. 1.2. Ряды Фурье: сигнал непрерывный и периодический

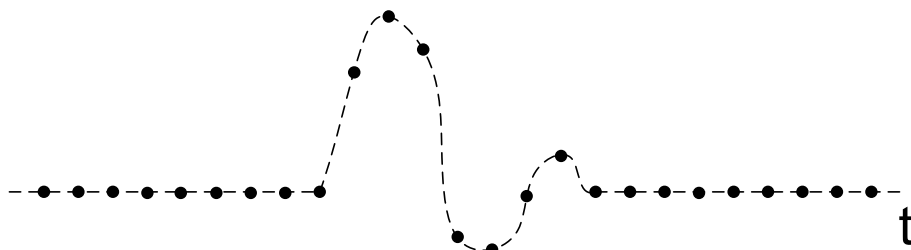


Рис. 1.3. Дискретные ряды Фурье: сигнал дискретный и аperiodический

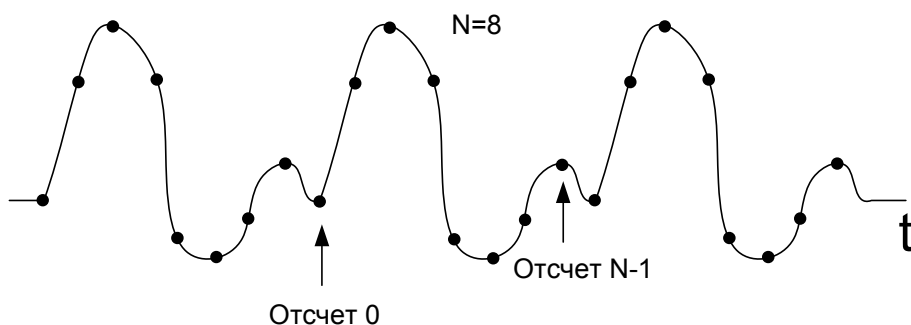


Рис. 1.4. Дискретное преобразование Фурье: (дискретные ряды Фурье) сигнал дискретный и периодический

Дискретное преобразование Фурье (ДПФ)

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) рис. 1.

Основная идея ДПФ ни чем не отличается от ПФ (см. рис. 1.5).

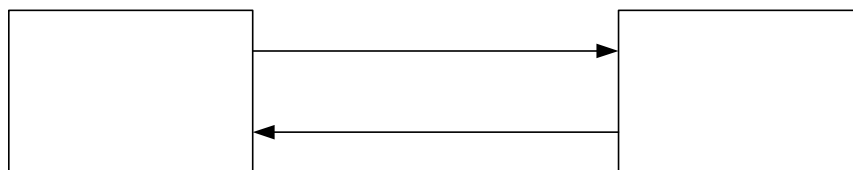


Рис. 1.5. Основная идея ДПФ

Для получения представления $x(t)$ (1.2) рядом Фурье в комплексной форме необходимо использовать соотношения в виде формулы Эйлера:

$$\begin{aligned}\cos n\omega_0 t &= \frac{1}{2}(e^{in\omega_0 t} + e^{-in\omega_0 t}); \\ \sin n\omega_0 t &= \frac{1}{2i}(e^{in\omega_0 t} - e^{-in\omega_0 t}); \quad i = \sqrt{-1}.\end{aligned}\quad (1.9)$$

Тогда

$$\begin{aligned}x(t) &= a_0 + \frac{1}{2} \sum_{n=1}^{\infty} \{a_n(e^{in\omega_0 t} + e^{-in\omega_0 t}) - ib_n(e^{in\omega_0 t} - e^{-in\omega_0 t})\} = \\ &= a_0 + \frac{1}{2} \sum_{n=1}^{\infty} \{(a_n - ib_n)e^{in\omega_0 t} + (a_n + ib_n)e^{-in\omega_0 t}\}\end{aligned}\quad (1.10)$$

Введем коэффициент

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

Тогда

$$\begin{aligned}C_n &= \frac{1}{T} \int_T x(t) [\cos n\omega_0 t - i \sin n\omega_0 t] dt \quad \text{или} \quad C_n = \frac{1}{T} \int_T x(t) e^{-in\omega_0 t} dt; \\ C_{-n} &= C_n^* = \frac{1}{2}(a_n + ib_n).\end{aligned}\quad (1.11)$$

Следовательно,

$$\begin{aligned}x(t) &= a_0 + \sum_{n=1}^{\infty} [C_n e^{in\omega_0 t} + C_n^* e^{-in\omega_0 t}] = a_0 + \sum_{\substack{n=-\infty \\ n \neq 0}}^{\infty} C_n e^{in\omega_0 t}; \\ C_0 &= \frac{1}{T} \int_T x(t) dt = a_0; \\ x(t) &= \sum_{n=-\infty}^{\infty} C_n e^{in\omega_0 t}.\end{aligned}\quad (1.12)$$

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

$$\boxed{C_x(k) = \frac{1}{N} \sum_{m=0}^{N-1} X(m) W^{km}}, \quad \text{где } k = \overline{0, N-1}, \quad W = e^{-i2\pi/N}, \quad i = \sqrt{-1}, \quad (1.13)$$

$$\boxed{X(m) = \sum_{k=0}^{N-1} C_x(k) W^{-km}}. \quad (1.14)$$

Выражения (1.13), (1.14) 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 $X(m) \leftrightarrow C_x(k)$, $Y(m) \leftrightarrow C_y(k)$ 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

$$C_x\left(\frac{N}{2} + l\right) = \overline{C_x\left(\frac{N}{2} - l\right)}, \forall l = \overline{0, N/2}. \quad (1.15)$$

Из (1.13) 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

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

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

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

$$C_z(k) = W^{-kh} C_x(k). \quad (1.16)$$

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

Можно выделить следующие области применения ДПФ:

- цифровой спектральный анализ
 - анализаторы спектра
 - обработка речи
 - обработка изображений
 - распознавание образов
- проектирование фильтров
 - вычисление импульсной характеристики по частотной
 - вычисление частотной характеристики по импульсной

- быстрое преобразование Фурье (БПФ) – простой алгоритм для эффективного вычисления ДПФ.

ДИСКРЕТНОЕ ПРЕОБРАЗОВАНИЕ ФУРЬЕ (ДПФ)

- Периодический сигнал может быть разложен на сумму выбранных должным образом косинусоидальных и синусоидальных функций (Жан Батист Жозеф Фурье, 1807).
- ДПФ работает с конечным числом (N) оцифрованных по времени отсчетов $X(m)$. Когда эти группы отсчетов повторяются, они становятся периодическими с точки зрения преобразования.
- Комплексный спектральный выход ДПФ $S(k)$ является результатом свертки входных отсчетов с базисными функциями синуса и косинуса.

Быстрое преобразование Фурье (БПФ)

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 (табл. 1.1).

Таблица 1.1

Эффективность БПФ

N	Умножений при ДПФ	Умножений при БПФ	Эффективность БПФ
256	65 536	1 024	64 : 1
512	262 144	2 304	114 : 1
1 024	1 048 576	5 120	205 : 1
2 048	4 194 304	11 264	372 : 1
4 096	16 777 216	24 576	683 : 1

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» специальными функциями.

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

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

Тогда

$$\begin{aligned} C_x(k) &= \frac{1}{8} \sum_{m=0}^7 X(m) W^{km}; \quad k = \overline{0, N-1}; \\ W &= e^{-i2\pi/8} = e^{-i\pi/4}; \\ 8C_x(k) &= \sum_{m=0}^7 X(m) \cos\left(\frac{mk\pi}{4}\right) - i \sum_{m=0}^7 X(m) \sin\left(\frac{mk\pi}{4}\right). \end{aligned} \quad (1.17)$$

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

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

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

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

$$= \begin{bmatrix} A_1^1 \begin{Bmatrix} X_0 + X_4 \\ X_1 + X_5 \\ X_2 + X_6 \\ X_3 + X_7 \end{Bmatrix} \\ B_1^1 \begin{Bmatrix} X_0 - X_4 \\ X_1 - X_5 \\ X_2 - X_6 \\ X_3 - X_7 \end{Bmatrix} \end{bmatrix}. \quad (1.18)$$

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

$$A_1^1 = \begin{bmatrix} A_1^2 & A_1^2 \\ A_2^2 & -A_2^2 \end{bmatrix}; \quad (1.19)$$

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

Подставляя выражения для A_I^1 и B_I^1 в (1.18) получим

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

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

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

$$B_2^2 \begin{Bmatrix} (X_0 - X_4) - W_2 (X_2 - X_6) \\ (X_1 - X_5) - W_2 (X_3 - X_7) \end{Bmatrix}. \quad (1.20)$$

Наконец, на последнем шаге получим

$$\begin{aligned}
& [(X_0 + X_4) + (X_2 + X_6)] + [(X_1 + X_5) + (X_3 + X_7)] \\
& [(X_0 + X_4) + (X_2 + X_6)] - [(X_1 + X_5) + (X_3 + X_7)] \\
& [(X_0 + X_4) - (X_2 + X_6)] + W_2[(X_1 + X_5) - (X_3 + X_7)] \\
& [(X_0 + X_4) - (X_2 + X_6)] - W_2[(X_1 + X_5) - (X_3 + X_7)] \\
& [(X_0 - X_4) + W_2(X_2 - X_6)] + W_1[(X_1 - X_5) + W_2(X_3 - X_7)] \\
& [(X_0 - X_4) + W_2(X_2 - X_6)] - W_1[(X_1 - X_5) + W_2(X_3 - X_7)] \\
& [(X_0 - X_4) - W_2(X_2 - X_6)] + W_3[(X_1 - X_5) - W_2(X_3 - X_7)] \\
& [(X_0 - X_4) - W_2(X_2 - X_6)] - W_3[(X_1 - X_5) - W_2(X_3 - X_7)]
\end{aligned} \tag{1.21}$$

Описанный алгоритм удобно представить графически (рис. 1.6).

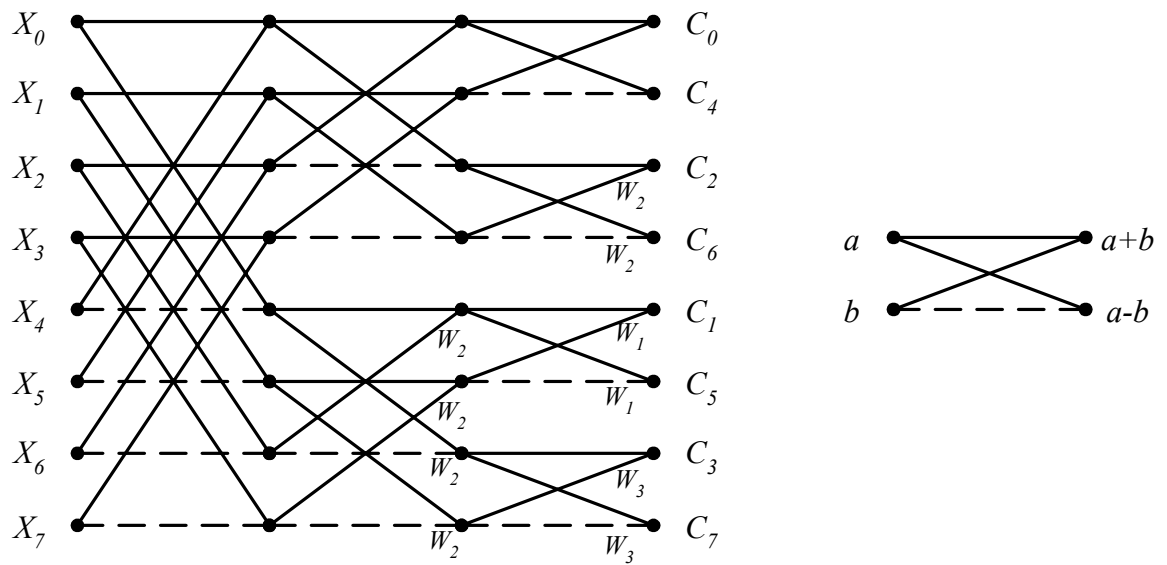


Рис. 1.6. Граф-схема быстрой процедуры вычисления коэффициентов преобразования Фурье

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

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

Для получения S_2 необходимо выполнить двоичную инверсию каждой $(n-1)$ -разрядной последовательности множества S_l , 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. 1.2).

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 1.2

Значения степени W

Номер итерации	Степени W ($N=16$)
1	$W_0 W_0 W_0 W_0 W_0 W_0 W_0 W_0 W_0 W_0 W_0 W_0 W_0 W_0 W_0 W_0$
2	$W_0 W_0 W_0 W_0 W_0 W_0 W_0 W_0 W_4 W_4 W_4 W_4 W_4 W_4 W_4 W_4$
3	$W_0 W_0 W_0 W_0 W_4 W_4 W_4 W_4 W_2 W_2 W_2 W_2 W_6 W_6 W_6 W_6$
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

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

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. 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

$$F_N = \prod_{i=1}^n F_i D_i. \quad (1.22)$$

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 :

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

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

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

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.

АЛГОРИТМ БПФ(a, N)

```
{
1. Если длина вектора равна 1, вернуть  $a$ .
2. Разбить вектор  $a$  на четную часть  $a^{чет} = (a_0, a_2, \dots, a_{N-2})$ 
   и нечетную  $a^{нечет} = (a_1, a_3, \dots, a_{N-1})$ .
3. Рекурсивно вызвать БПФ на каждой из частей
    $b^{чет} = \text{БПФ}(a^{чет})$ 
    $b^{нечет} = \text{БПФ}(a^{нечет})$ 
4. Объединение результатов.
   а. (инициализация) Присвоить  $\omega_N$  значение главного корня из
      единицы
      
$$\omega_N = e^{\frac{2\pi i}{N}} = \cos \frac{2\pi}{N} + i \sin \frac{2\pi}{N}$$

   б. (инициализация) Присвоить  $\omega = 1$ 
   с. В цикле вычислить левую и правую часть одновременно:
      for(  $j=0$ ;  $j < N/2$ ;  $j++$ )
      {
          $y_j = b_j^{чет} + \omega b_j^{нечет}$ 
          $y_{j+N/2} = b_j^{чет} - \omega b_j^{нечет}$ 
      }
5. Вернуть вектор  $y$ .
}
```

Ниже изображено дерево рекурсий, рис. 1.7. Каждый уровень, начиная снизу, соответствует проходу алгоритма по всему вектору и объединению сначала одиночных элементов в пары, затем пар в четверки и так далее до конца. Обратите внимание на то, что порядок индексов на верхнем уровне не соответствует нижнему. Это естественно, если учесть, что нечетные индексы после бабочки идут в правую половину вектора, а четные – в левую.

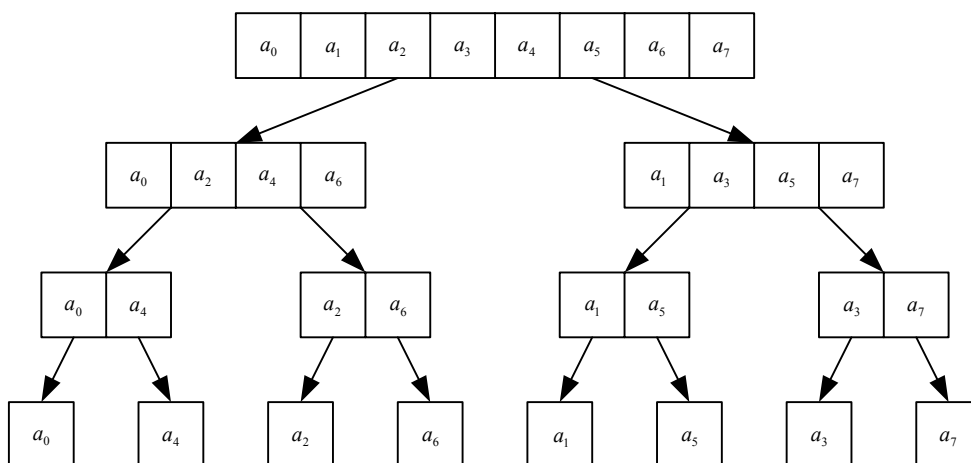


Рис. 1.7. Дерево рекурсий для 8 элементов

Чтобы техника работала на вход алгоритма должен поступать вектор нижнего уровня $(a_0, a_4, a_2, a_6, a_1, a_5, a_3, a_7)$ – тогда объединения произойдут в правильном порядке, как изображено выше, и дадут в результате вектор (a_0, a_1, \dots, a_7) . Таким образом, перед вызовом процедуры следует провести перестановку элементов массива.

3. Задание

1. Ознакомьтесь с теоретической частью.
2. В соответствии с вариантом (табл. 3.1) для заданной функции выполните прямое и обратное БПФ - напишите программу на языке С, С++. Результатом работы программы должны быть: график заданной функции, график по результатам прямого БПФ, график по результатам обратного БПФ.
3. Промоделируйте данную работу в MATLAB, сравните полученные результаты.

4. Напишите отчет.

Содержание отчета:

- исходные данные;
- краткое описание алгоритма работы программы;
- график заданной функции, график по результатам прямого БПФ, график по результатам обратного БПФ;
- результаты моделирования в MATLAB;
- анализ и пояснение полученных результатов;
- выводы.

Таблица 3.1

Варианты задания

№ варианта	Заданная функция	N для БПФ
1	$y=\cos(3x)+\sin(2x)$	8
2	$y=\sin(3x)+\cos(x)$	16
3	$y=\cos(2x)+\sin(5x)$	8
4	$y=\sin(2x)+\cos(7x)$	16
5	$y=\cos(x)+\sin(x)$	8
6	$y=\sin(x)+\cos(4x)$	16
7	$y=\cos(5x)+\sin(6x)$	8
8	$y=\sin(5x)+\cos(x)$	16

4. 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 №2

Операции свертки и корреляции

1. Цель работы

Изучение операций корреляции и свертки, их основных свойств, а также методики их получения с помощью быстрого преобразования Фурье (БПФ) на основе теорем о корреляции и свертке.

2. Теоретические сведения

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.

Как было сказано в теоретической части предыдущей лабораторной работы, сигнал может быть разложен на группу составляющих, называемых импульсами. В результате, импульсное разложение представляет способ поточечного анализа сигнала. Один из методов такого разложения – разложение Фурье.

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

Теорема свертки

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

$$Z(m) = \frac{1}{N} \sum_{h=0}^{N-1} X(h)Y(m-h), \quad m = \overline{0, N-1}, \quad (2.1)$$

то

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

Доказательство

Вычисляя $Z(m)$, получим

$$C_z(k) = \frac{1}{N} \sum_{m=0}^{N-1} Z(m)W^{km}. \quad (2.2)$$

Подставляя в (2.2) соотношение свертки (2.1), 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.

Теорема корреляции

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

$$\hat{Z}(m) = \frac{1}{N} \sum_{h=0}^{N-1} X(h) Y(m+h), \text{ где } m = \overline{0, N-1}, \quad (2.3)$$

то $C_z(k) = \overline{C_x(k)} C_y(k)$,

где $\overline{C_x(k)}$ - комплексное сопряженное $C_x(k)$

Доказательство

По определению имеем

$$C_z(k) = \frac{1}{N} \sum_{m=0}^{N-1} \hat{Z}(m) W^{km}. \quad (2.4)$$

Подставляя (2.3) в (2.4) и меняя порядок суммирования, получаем

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

Применяя теорему сдвига, будем иметь

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

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, $C_z(k) = C_y(k) \overline{C_x(k)}$.

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

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

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

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

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

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

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

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

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

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

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

В общем виде корреляцию двух последовательностей можно записать как

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

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

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

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 вычислений корреляции будет иметь вид рис.2.1.

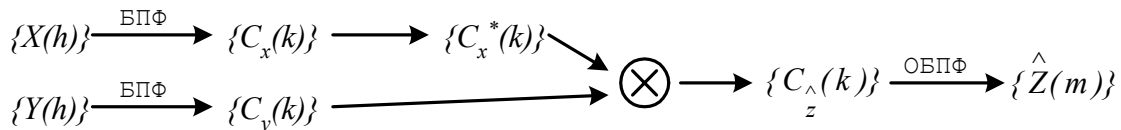


Рис. 2.1. Схема вычисления корреляции

В свою очередь, схему вычисления свертки можно представить как показано на рис.2.2.

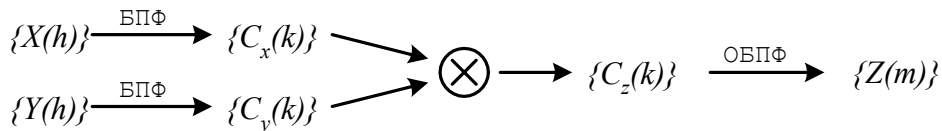


Рис. 2.2. Схема вычисления свертки

3. Задание

1. Ознакомьтесь с теоретической частью.
2. Для заданных функций (табл. 2.1) написать на языке C, C++ программу реализации одномерной свертки на основе уже реализованных предварительно функций прямого и обратного БПФ. Результатом работы программы: графики заданных функции, график результата свертки.
3. Для заданных функций написать на языке C, C++ программу реализации корреляции на основе уже реализованных предварительно функций прямого и обратного БПФ. Результатом работы программы должны быть: графики заданных функции, график результата корреляции.
4. Промоделируйте данную работу в MATLAB, сравните полученные результаты.
5. Напишите отчет.
Содержание отчета:
 - исходные данные;
 - краткое описание алгоритма работы программы;
 - графики заданных функций, графики результатов свертки и корреляции;
 - результаты моделирования в MATLAB;
 - анализ и пояснение полученных результатов;
 - выводы.

Таблица 2.1

Варианты задания

№ варианта	Заданная функция	N для БПФ
1	$y=\cos(3x); z=\sin(2x)$	8
2	$y=\sin(3x); z=\cos(x)$	16
3	$y=\cos(2x); z=\sin(5x)$	8
4	$y=\sin(2x); z=\cos(7x)$	16
5	$y=\cos(x); z=\sin(x)$	8
6	$y=\sin(x); z=\cos(4x)$	16
7	$y=\cos(5x); z=\sin(6x)$	8
8	$y=\sin(5x); z=\cos(x)$	16

4. Контрольные вопросы

1. Для чего используются понятия “корреляция” и “свертка” в такой области, как цифровая обработка сигналов и изображений?
2. Привести описание операций “свертка” и “корреляция”.
3. Каким образом связаны понятия “свертка”, “корреляция” и БПФ?
4. Привести теорему свертки, теорему корреляции.

Лабораторная работа №3

Преобразование Уолша

1. Цель работы

Изучение преобразования Уолша и его основных свойств, а также методики получения быстрого преобразования Уолша (БПУ).

2. Теоретические сведения

Особый класс систем ортогональных функций составляют системы кусочно-постоянных функций, таких как функции Уолша, Адамара и Хаара. Эти системы имеют большое практическое значение, особенно для цифровых систем, поскольку они характеризуются высокоэффективными алгоритмами быстрых преобразований.

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 (способы определения функций Уолша):

- упорядочение по частоте (по Уолшу с помощью функций Радемахера);
- упорядочение по Пэли (диадическое);
- упорядочение по Адамару, где под частотой функции понимается число пересечений нулевого уровня в единицу времени.

Рассмотрим способ, основанный на взаимосвязи функций Уолша с функциями Радемахера и способ, основанный на матрицах Адамара.

Функции Радемахера получаются из синусоидальных функций с помощью соотношения

$$r_k(t) = \text{sign}[\sin(2^k \pi t)], \quad 0 \leq t < 1, \quad (3.1)$$

где аргумент t – безразмерное время, т.е. время, нормированное к произвольному интервалу T_0 , а целое положительное число k – порядок функции. Символом sign (сигнум-функция) обозначается функция:

$$\text{sign} x = \begin{cases} 1 & \text{при } x > 0, \\ -1 & \text{при } x \leq 0. \end{cases} \quad (3.2)$$

Функции Радемахера принимают одно из двух значений ± 1 и имеют вид меандра.

Функции Радемахера ортонормированны с единичной весовой функцией на интервале $0 \leq t < 1$, т.к. для любых двух функций $r_m(t)$, $r_n(t)$ имеют место соотношения:

$$\int_0^1 r_m(t)r_n(t)dt = \begin{cases} 1 & \text{при } m = n, \\ 0 & \text{при } m \neq n. \end{cases} \quad (3.3)$$

Все функции Радемахера являются нечетным и относительно середины интервала определения и не могут быть использованы для аппроксимации сигналов, четных относительно этой точки. Иными словами, система функций Радемахера – неполная. На рис. 3.1 приведены функции Радемахера для $N = 8$.

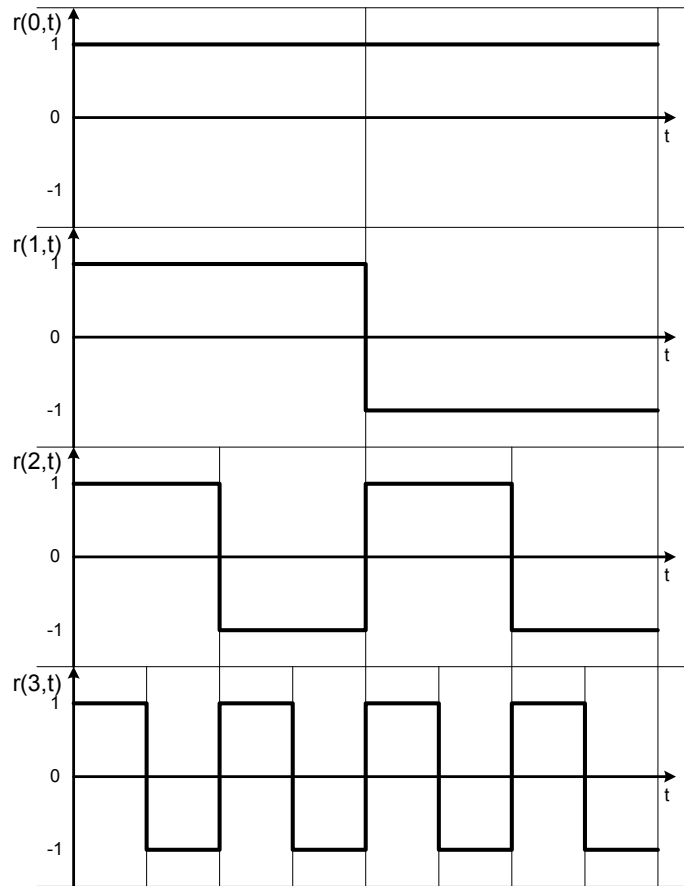


Рис. 3.1 Функции Радемахера для $N=8$

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

Если $r_1(t) = + + + + - - - -$

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

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

где “+” соответствует +1, а “-” соответствует -1.

Функции Уолша, образующие полную ортонормированную систему, можно сформировать, образуя произведения соответствующих функций Радемахера. Первые восемь функций Уолша представлены на рис. 3.2.

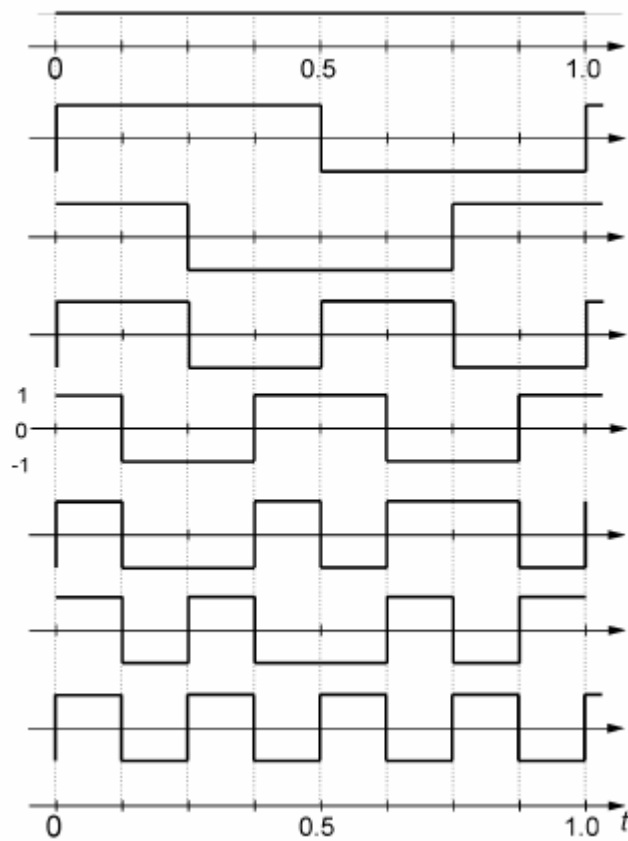


Рис. 3.2. Первые восемь функций Уолша

В свою очередь функции Уолша можно представить следующим образом:

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

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:

$$wal(n, t) = \prod_{k=0}^{r-1} [r_k(t)]^{n_{r-k+1} \oplus n_{r-k}} \quad (3.4)$$

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:

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

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

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

На основе двоичного представления числа $n=6$ не сложно 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.

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

$$0 \oplus 0 = 0$$

$$1 \oplus 0 = 1$$

$$0 \oplus 1 = 1$$

$$1 \oplus 1 = 0$$

Например, код Грея, соответствующий двоичному числу 101101, может быть образован как на рис. 3.3. Трехразрядный код Грея показан в табл. 3.1.

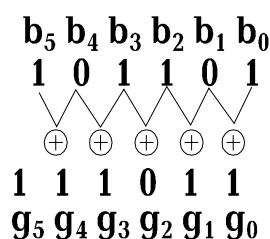


Рис. 3.3. Преобразование двоичного кода в код Грея

Таблица 3.1

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. 3.4.

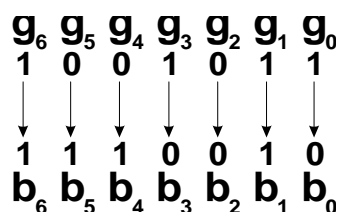


Рис. 3.4. Преобразование кода Грея в двоичный код

Приведем некоторые **свойства функций Уолша**.

1. Функции Уолша *ортонормированы* 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 $wal(i, t)$.

Способ нумерации функций в системе называется упорядочением. Функции Уолша, сформированные в соответствии с выражением (3.4), 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 $N = 2^n$ функций, всегда можно представить в виде произведения степеней первых n 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.

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

Далее (табл. 3.2, 3.3) приводятся нумерация функций Уолша в базисе из 16 функций при различных способах упорядочения и нумерация для базиса из 8 функций. Сравнение таблиц показывает, что нумерация одних и тех же функций в упорядочении Адамара меняется в зависимости от размерности базиса.

Таблица 3.2

Нумерация функций Уолша
при различных способах
упорядочения, N=16

wal(n,t)	had(h,t)	pal(p,t)
0	0	0
1	8	1
2	12	3
3	4	2
4	6	6
5	14	7
6	10	5
7	2	4
8	3	12
9	11	13
10	15	15
11	7	14
12	5	10
13	13	11
14	9	9
15	1	8

Таблица 3.3

Нумерация функций Уолша
при различных способах
упорядочения, N=8

wal(n,t)	had(h,t)	pal(p,t)
0	0	0
1	4	1
2	6	3
3	2	2
4	3	6
5	7	7
6	5	5
7	1	4

Быстрое преобразование Уолша можно получить с помощью технологии разбиения матриц. Графическая схема алгоритма показана на рис. 3.5. 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

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

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

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

$$\begin{aligned}
8C_x(0) &= x_2(0) + x_2(1) = x_3(0); \\
8C_x(1) &= x_2(0) - x_2(1) = x_3(1); \\
8C_x(2) &= x_2(2) + x_2(3) = x_3(2); \\
8C_x(3) &= x_2(2) - x_2(3) = x_3(3); \\
8C_x(4) &= x_2(4) + x_2(5) = x_3(4); \\
8C_x(5) &= x_2(4) - x_2(5) = x_3(5); \\
8C_x(6) &= x_2(6) + x_2(7) = x_3(6); \\
8C_x(7) &= x_2(6) - x_2(7) = x_3(7).
\end{aligned}$$

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 $N \log_2 N$.

На рис. 3.5 приведена граф-схема процедуры быстрого алгоритма Уолша.

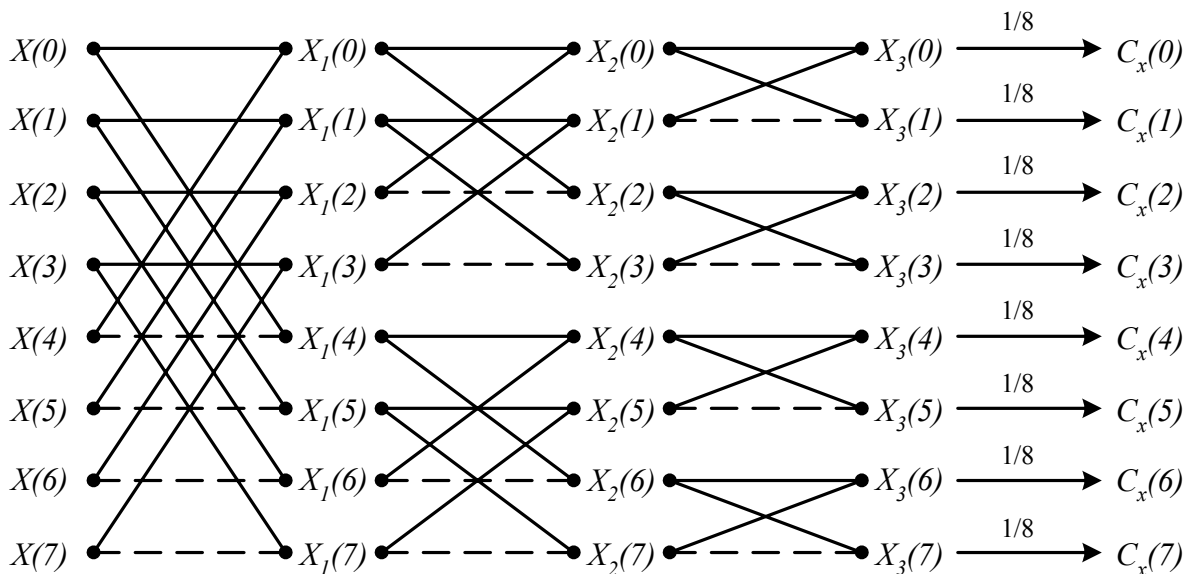


Рис. 3.5. Процедура быстрого алгоритма Уолша

3. Задание

1. Ознакомьтесь с теоретической частью.

2. Реализовать на языке C, C++ БПУ. Исходными данными для БПУ будет одна из функций Уолша, упорядоченная по частоте, и одна из функций Уолша, упорядоченная по Адамару (т.е. одна из 8 гармоник, $N=8$). Для каждой функции выполнить БПУ и вывести результат (2 массива по 8 чисел). Функции Уолша, упорядоченные по частоте, получать в результате произведения функций

Радемахера в нужной степени (функции Радемахера задавать в качестве исходных данных в коде программы), функции Уолша, упорядоченные по Адамару, задавать в качестве исходных данных в коде программы.

3. Пояснить результаты работы программы.

4. Промоделируйте данную работу в MATLAB, сравните полученные результаты.

5. Напишите отчет.

Содержание отчета:

- исходные данные;
- краткое описание алгоритма работы программы;
- заданная функция, результаты БПУ;
- результаты моделирования в MATLAB;
- анализ и пояснение полученных результатов;
- выводы.

4. 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 №4

Вейвлет-преобразование

1. Цель работы

Изучение основ вейвлет-анализа, непрерывного вейвлет-преобразования (НВП), дискретизированного НВП, дискретного вейвлет-преобразования (ДВП), методики быстрого вейвлет-преобразования (БВП) с помощью алгоритма Маллата и лифтинговой схемы.

2. Теоретические сведения

Методы обработки стационарных сигналов

Большинство сигналов, встречающихся на практике, представлено во *временной области*, т.е. сигнал есть функция времени. Таким образом, при отображении сигнала на графике одной из координат (независимой) является ось времени, а другой координатой (зависимой) – ось амплитуд. Таким образом, получаем *амплитудно-временное представление* сигнала. Для большинства приложений обработки сигналов это представление не является наилучшим. Во многих случаях наиболее значимая информация скрыта в частотной области сигнала. Частотный спектр есть совокупность частотных (спектральных) компонент и он отображает наличие тех или иных частот в сигнале. Частота измеряется в Герцах (Гц), или в числе периодов в секунду. На рис. 4.1 представлены три косинусоиды: 3, 10 и 50 Гц соответственно.

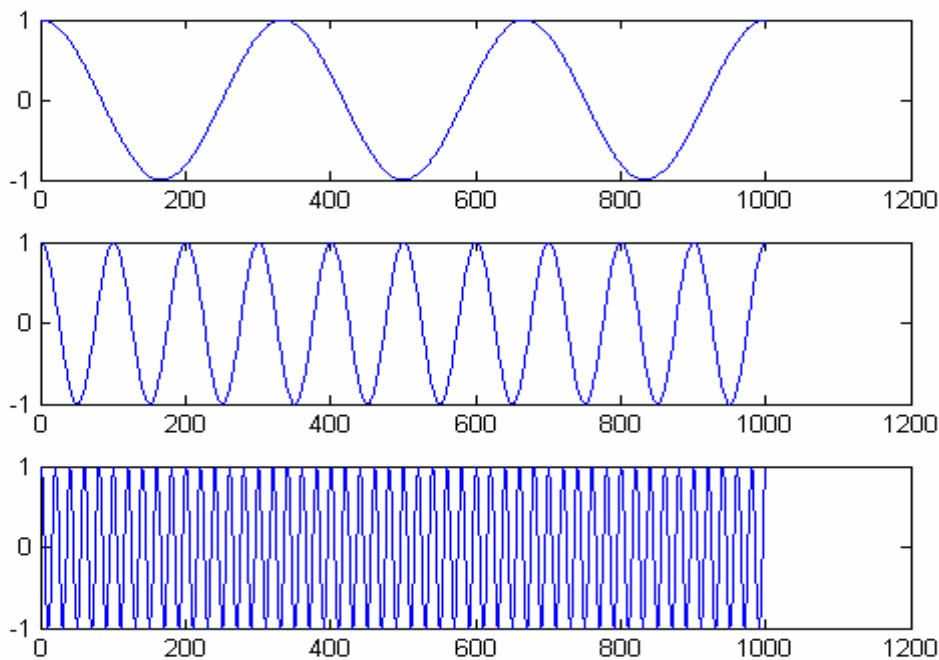


Рис. 4.1 Косинусоиды с частотой 3, 10 и 50 Гц соответственно

Классическим методом частотного анализа сигналов является преобразование Фурье, суть которого пояснялась в лабораторной работе №1.

Результат преобразования Фурье – амплитудно-частотный спектр, по которому можно определить присутствие некоторой частоты в исследуемом сигнале. Результаты преобразования Фурье трех косинусоид (3, 10 и 50 Гц соответственно) представлены ниже на рис. 4.2.

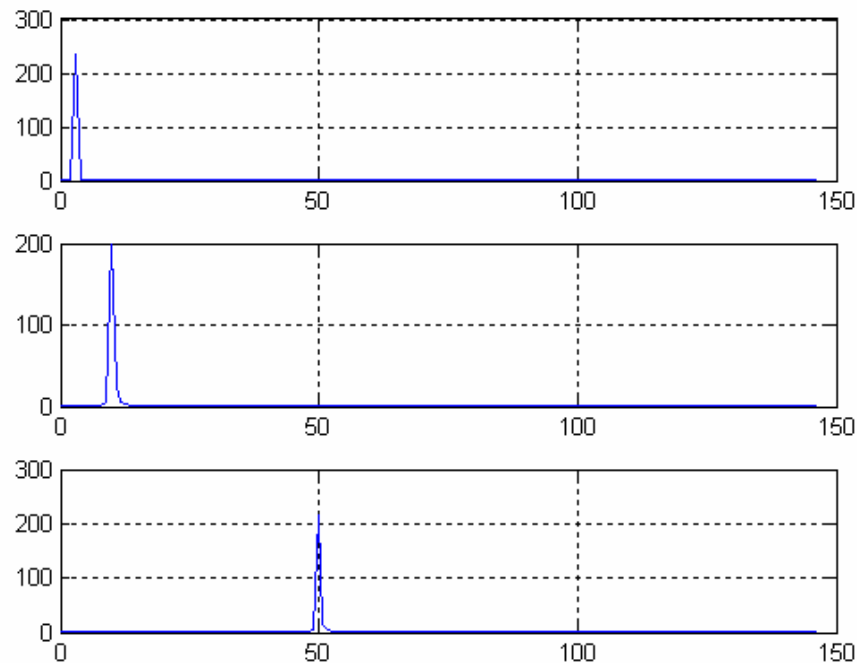


Рис. 4.2. Результаты преобразования Фурье косинусоид с частотой 3, 10 и 50 Гц соответственно

Приведем еще один пример стационарного сигнала $x(t) = \cos(2\pi 10t) + \cos(2\pi 25t) + \cos(2\pi 50t) + \cos(2\pi 100t)$, его график и результаты преобразования Фурье (рис. 4.3).

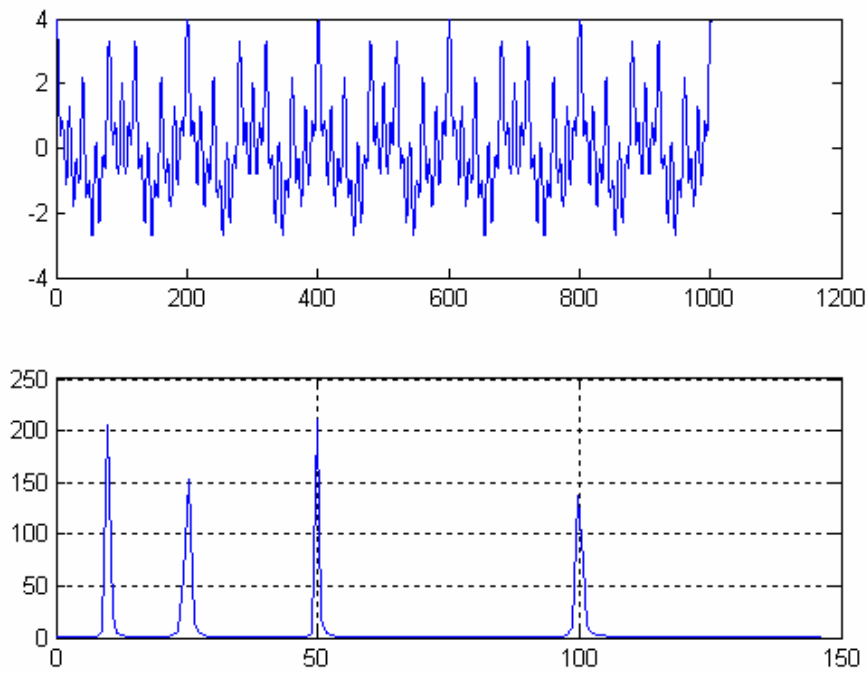


Рис. 4.3. Стационарный сигнал, результаты ПФ данного сигнала

Большинство реальных сигналов имеет сложные частотно-временные характеристики. Как правило, такие сигналы состоят из близких по времени, короткоживущих высокочастотных компонент и долговременных, близких по частоте низкочастотных компонент.

Сигнал на рис. 4.4 нестационарный. Его частота непрерывно изменяется во времени (четыре частотные компоненты соответствуют частотам 100, 50, 25, 10 Гц.)

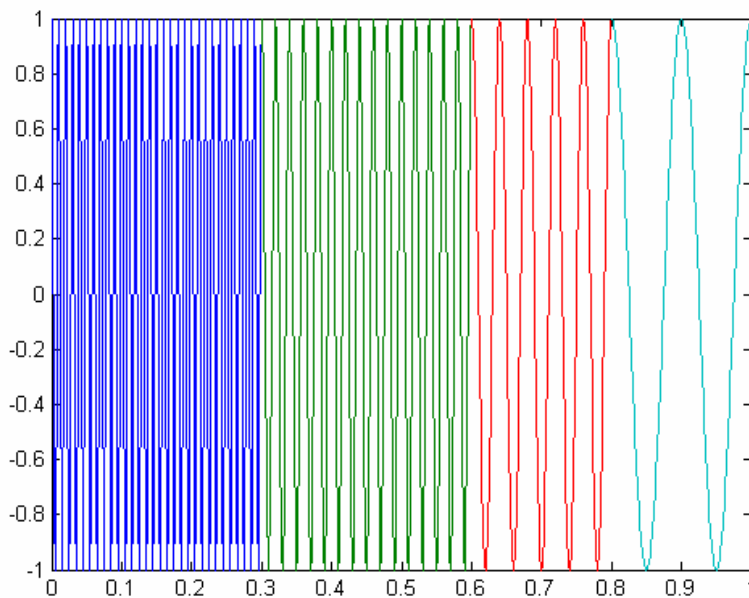


Рис. 4.4. Нестационарный сигнал

Спектр этого сигнала (ПФ) будет иметь вид (рис. 4.5):

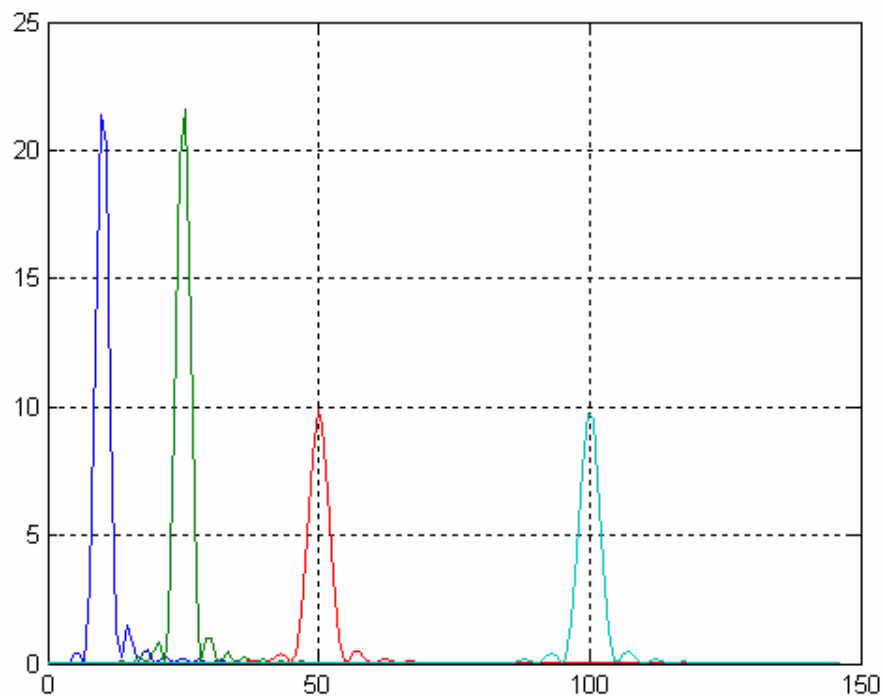


Рис. 4.5. Спектр нестационарного сигнала

В данном случае появляются небольшие «ложные» частоты и неодинаковости амплитудных пиков, поэтому ПФ не подходит для нестационарных сигналов.

Для анализа таких сигналов необходим метод, способный обеспечить хорошее разрешение и по частоте, и по времени. Первое требуется для локализации низкочастотных составляющих, второе – для разрешения компонент высокой частоты. Вейвлет - преобразование благодаря хорошей приспособленности к анализу нестационарных сигналов стало мощной альтернативой преобразованию Фурье.

Преобразование Фурье представляет сигнал, заданный во временной области, в виде разложения по ортогональным базисным функциям (синусам и косинусам), выделяя, таким образом, частотные компоненты. Недостаток преобразования Фурье заключается в том, что частотные компоненты не могут быть локализованы во времени, что накладывает ограничения на применимость данного метода к ряду задач (например, в случае изучения динамики изменения частотных параметров сигнала на временном интервале).

Существует два подхода к анализу нестационарных сигналов такого типа. Первый – оконное преобразование Фурье (short-time Fourier transform). Следуя по этому пути, мы работаем с нестационарным сигналом, как со стационарным, предварительно разбив его на сегменты (окна), статистика которых не меняется со временем. Второй подход – вейвлет преобразование. В этом случае нестационарный сигнал анализируется путем разложения по базисным

функциям, полученным из некоторого прототипа путем сжатий, растяжений и сдвигов. Функция прототип называется материнским или анализирующим вейвлетом.

Краткий обзор оконного преобразования Фурье

В случае, когда не возникает вопрос о локализации временного положения частот, метод Фурье дает хорошие результаты. Но при необходимости определить временной интервал присутствия частоты приходится применять другие методы.

Одним из таких методов является обобщенный метод Фурье (оконное преобразование Фурье, ОПФ). Этот метод состоит из следующих этапов:

1. в исследуемой функции создается «окно» – временной интервал, для которого функция $f(x) \neq 0$, и $f(x) = 0$ для остальных значений;
2. для этого «окна» вычисляется преобразование Фурье;
3. «окно» сдвигается, и для него также вычисляется преобразование Фурье.

При ОПФ сигнал делится на отрезки («окна»), в пределах которых его можно считать стационарным. «Пройдя» таким «окном» вдоль всего сигнала, получается некоторая трехмерная функция, зависящая от положения «окна» и частоты (рис. 4.6).

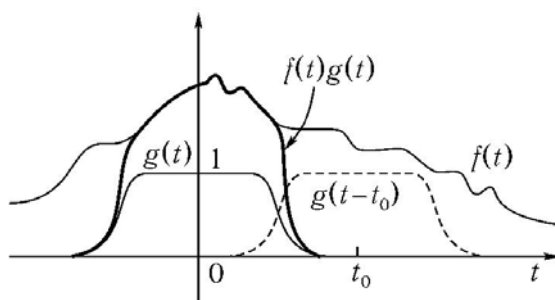


Рис. 4.6. Оконное преобразование Фурье:

функция $f(t)$ перемножается с оконной функцией $g(t)$, и вычисляются коэффициенты произведения $f(t)g(t)$. Затем процедура повторяется для сдвигов окна $g(t-t_0)$, $g(t-2t_0)$, ...

Данный подход позволяет определить факт присутствия в сигнале любой частоты и интервал ее присутствия. Это значительно расширяет возможности метода по сравнению с классическим преобразованием Фурье, но существуют и определенные недостатки. Согласно следствиям принципа неопределенности Гейзенберга в данном случае нельзя утверждать факт наличия частоты ω_0 в сигнале в момент времени t_0 – можно лишь определить, что спектр частот (ω_1, ω_2) присутствует в интервале (t_1, t_2) . Причем разрешение по частоте (по времени) остается постоянным вне зависимости от области частот (времен), в которых производится исследование. Узкое окно обеспечивает временное разрешение, а широкое – частотное. Проблема состоит в том, что приходится

выбирать окно «раз и навсегда», то есть для анализа сигнала, тогда как разные его участки могут потребовать применения разных окон.

Поэтому, если, например, в сигнале существенна только высокочастотная составляющая, то увеличить разрешение можно только, изменив параметры метода. В качестве метода, не обладающего подобного рода недостатками, был предложен аппарат вейвлет - анализа.

Основные положения вейвлет-анализа

Различают дискретный и непрерывный вейвлет-анализ, аппарат которых можно применять как для непрерывных, так и для дискретных сигналов.

Сигнал анализируется путем разложения по базисным функциям, полученным из некоторого прототипа путем сжатий, растяжений и сдвигов. Функция-прототип называется анализирующим (материнским) вейвлетом.

Вейвлет-функция должна удовлетворять двум условиям:

1. Среднее значение (интеграл по всей прямой) равно 0.
2. Функция быстро убывает при $t \rightarrow \infty$.

Обычно, функция-вейвлет обозначается буквой ψ . В общем случае **непрерывное вейвлет-преобразование (CWT)** функции $f(t)$ выглядит так:

$$\Psi_x^\omega(\tau, s) = \frac{1}{\sqrt{|s|}} \int x(t) \psi^* \left(\frac{t - \tau}{s} \right) dt, \quad (4.1)$$

где τ - параметр сдвига, s - параметр масштаба, функция $\psi^* \left(\frac{t - \tau}{s} \right)$ - это «материнский вейвлет», то есть функция преобразования.

Слово вейвлет означает маленькая волна. Под маленькой понимается то, что эта функция (окно) имеет конечную ширину. Термин «материнский» означает, что функции с различной шириной носителя, используемые в преобразовании, порождаются одной базовой функцией - материнским вейвлетом. То есть материнский вейвлет является прототипом для всех оконных функций.

Основные вейвлетообразующие функции, или материнские вейвлеты, приведены в табл. 4.1.

Наиболее распространенные вещественные базисы конструируются на основе производных функции Гаусса ($g_0(t) = \exp(-t^2/2)$). Это обусловлено тем обстоятельством, что функция Гаусса имеет наилучшие показатели локализации как во временной, так и в частотной областях.

На рис. 4.7 показаны вейвлеты первых четырех порядков и модули их спектральной плотности. При $n=1$ получаем вейвлет первого порядка, называемый WAVE-вейвлетом с равным нулю нулевым моментом. При $n=2$ получаем МНАТ-вейвлет, называемый «мексиканская шляпа» (mexican hat - похож на сомбреро). У него нулевой и первый моменты равны нулю. Он имеет лучшее разрешение, чем WAVE-вейвлет.

Совместное использование вейвлетов $g_1 - g_4$ для ВП существенно повышает точность вейвлет-анализа.

Таблица 4.1

Основные материнские вейвлеты

Вейвлеты	Аналитическая запись $\psi(t)$	Спектральная плотность $\psi(\omega)$
Вещественные непрерывные базисы		
Гауссовы: – первого порядка, или WAVE-вейвлет, – второго порядка, или МНАТ-вейвлет «мексиканская шляпа» – <u>mexican hat</u>), – n -го порядка,	$-t \exp(-t^2/2)$ $(1-t^2) \exp(-t^2/2)$ $(-1)^n \frac{d^n}{dt^n} [\exp(-t^2/2)]$	$(i\omega) \sqrt{2\pi} \exp(-\omega^2/2)$ $(i\omega)^2 \sqrt{2\pi} \exp(-\omega^2/2)$ $(-1)^n (i\omega)^n \sqrt{2\pi} \exp(-\omega^2/2)$
DOG – <u>d</u> ifference of <u>g</u> aussians	$e^{-t^2/2} - 0,5e^{-t^2/8}$	$\sqrt{2\pi}(e^{-\omega^2/2} - e^{-2\omega^2})$
LP-Littlewood & Paley	$(\pi t)^{-1}(\sin 2\pi t - \sin \pi t)$	$\begin{cases} (2\pi)^{-1/2}, \pi \leq t \leq 2\pi, \\ 0, \text{ в противном случае} \end{cases}$
Вещественные дискретные		
НААР-вейвлет	$\begin{cases} 1, & 0 \leq t \leq 1/2, \\ \geq -1, & 1/2 \leq t \leq 1, \\ 0, & t < 0, t > 1. \end{cases}$	$ie^{i\omega/2} \frac{\sin^2 \omega/4}{\omega/4}$
ФНАТ-вейвлет, или «французская шляпа» (<u>F</u> rench <u>h</u> at – похож на цилиндр)	$\begin{cases} 1, & t \leq 1/3, \\ \geq -1/2, & 1/3 \leq t \leq 1, \\ 0, & t > 1. \end{cases}$	$\frac{4 \sin^3 \omega/3}{3 \omega/3}$
Комплексные		
Морле (Morlet)	$e^{i\omega_0 t} e^{-t^2/2}$	$\sigma(\omega) \sqrt{2\pi} e^{-(\omega-\omega_0)^2/2}$
Пауля (Paul) (чем больше n , тем больше нулевых моментов имеет вейвлет)	$\Gamma(n+1) \frac{i^n}{(1-n)^{n+1}}$	$\sigma(\omega) \sqrt{2\pi} (\omega)^n e^{-\omega}$

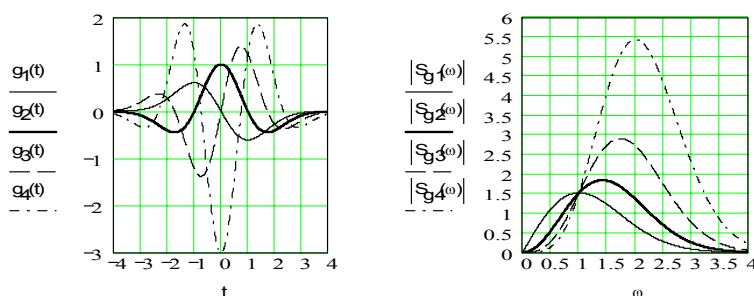


Рис. 4.7. Вейвлеты первых четырех порядков и модули их спектральной плотности

Наиболее простой пример *дискретного* вейвлета – это HAAR-вейвлет. Недостатком его являются несимметричность формы и негладкость – резкие границы в t -области, вследствие чего возникает бесконечное чередование «лепестков» в частотной области, хотя и убывающих как $1/\omega$.

LR -вейвлет, имеющий, наоборот, резкие границы в ω -области, можно считать другим предельным случаем.

Среди комплексных вейвлетов в наиболее часто используется базис, основанный на хорошо локализованном и во временной и в частотной областях вейвлете Морле. Характерный параметр ω_0 позволяет изменять избирательность базиса. Вещественная и мнимая части $\psi(t)$ – это амплитудно-модулированные колебания.

Выбор конкретного материнского вейвлета (будь то непрерывный или дискретный) целиком зависит от характера поставленной задачи и от конкретного анализируемого сигнала. Разные сигналы удастся анализировать тем или иным способом, и критерием успеха обычно служит простота получаемого разложения. При этом решающим фактором оказываются интуиция и практический опыт исследователя.

При ВП мы не имеем частотного параметра, как это было при ОПФ. Вместо него здесь имеется параметр масштаба, который можно определить как величину, обратную частоте. Параметр масштаба в вейвлет-анализе имеет аналогию с масштабом географических карт. Большое значение масштаба соответствуют малому количеству деталей, глобальному представлению сигнала, а низкие значения масштаба позволяют различить детали. Аналогично, в терминах частоты, низкие частоты соответствуют глобальной информации о сигнале (которая содержится на всей его протяженности), а высокие частоты – детальной информации, скрытым особенностям, которые имеют обычно малую протяженность. Масштабирование, как математическая операция, расширяет или сжимает сигнал. Поэтому, при $s > 1$ расширяет сигнал, а при $s < 1$ сжимает его.

В настоящее время большинство вычислений выполняют компьютеры. Это означает, что ни ПФ, ни ОПФ, ни НВП не могут быть практически вычислены путем взятия интегралов. Так как интегралы при вычислении заменяются суммами, то достаточно логичным будет применить дискретизацию частотно-временной области, то есть перейти к **дискретизированному НВП**. Параметр масштаба s дискретизируется на логарифмической сетке, а параметр времени дискретизируется в соответствии с параметром масштаба, что означает, что на разных масштабах имеет место различная частота дискретизации. Чем больше s , тем меньше временных отсчетов мы берем – поэтому для низких частот получаем хорошую частотную и скудную временную информацию, а для высоких частот наоборот. Дискретизированное НВП будет определяться следующим выражением:

$$\Psi_x(j, k) = \int x(t) \psi_{j,k}(t) dt, \quad (4.2)$$

Из-за некоторых недостатков от дискретизированного НВП перешли к **дискретному вейвлет-преобразованию (ДВП)**.

Для начала определим понятие *субполосное кодирование (subband coding)*. Это результат свертки сигнала с несколькими полосовыми фильтрами и децимацией результата. Совокупность набора фильтров и дециматоров называется *банком фильтров*. Каждый получившийся в результате преобразования сигнал несет в себе информацию о спектральной составляющей исходного сигнала при некотором пространственном (временном) масштабе. Для обратного синтеза сигнала выполняется операция интерполяции субполосных сигналов, фильтрация и их сложение. Использование субполосного кодирования вкупе с *кратномасштабным анализом (multiresolution analysis)*, который представляет собой процесс декомпозиции сигнала на различных частотах и различном разрешении одновременно, позволяет получить масштабно-временное представление сигнала. Разрешение сигнала изменяется за счет фильтрации сигнала и является мерой количества детальной информации в нем, масштаб изменяется за счет интерполяции и децимации. Интерполяция соответствует увеличению частоты дискретизации сигнала, а децимация понижению. Эти процедуры осуществляются добавлением новых отсчетов между существующими отсчетами сигнала и удалением некоторых отсчетов из сигнала соответственно.

В ходе ДВП сигнал анализируется в различных частотных полосах с различным разрешением путем декомпозиции на грубую аппроксимацию (полусуммы соседних значений сигнала) и детали (полуразности соседних значений сигнала). Таким образом, определены два множества функций: масштабирующие функции и вейвлеты, соответствующие низкочастотным (НЧ) и высокочастотным (ВЧ) фильтрам.

$$\text{масштабирующие коэффициенты: } a_{j-1,k} = \frac{1}{\sqrt{2}}(a_{j,2k} + a_{j,2k+1}) \quad (4.3)$$

$$\text{вейвлет коэффициенты: } b_{j-1,k} = \frac{1}{\sqrt{2}}(a_{j,2k} - a_{j,2k+1}) \quad (4.4)$$

Эта особенность была замечена Маллатом и изложена им в одной из своих работ.

$$Y_{low}[k] = \sum_n x(n)h(2k - n) \quad (4.5)$$

$$Y_{high}[k] = \sum_n x(n)g(2k - n) \quad (4.6)$$

Эти два выражения определяют два фильтра: низкочастотный и высокочастотный. Добавляя к этой группе фильтров два дециматора, использование которых обусловлено избыточностью данных получаемых с выходов фильтров, получаем банк фильтров анализа (рис. 4.8). Следует отметить, что операция децимации в данном случае удаляет из сигнала нечетные компоненты и оставляет четные. Единичное применение такого банка

фильтров к сигналу делит его на две части, в одной из которых содержатся высокие частоты сигнала, а в другой низкие. В матричном виде это выглядит так:

$$\begin{bmatrix} A_n \\ D_n \end{bmatrix} x = \begin{bmatrix} a_{n-1} \\ d_{n-1} \end{bmatrix}, \quad (4.7)$$

где A_n, D_n фильтрующие матрицы размерностью $2^{n-1} \times 2^n$, построенные на основании коэффициентов фильтра в зависимости от разрядности x и претерпевшие децимацию – удаление нечетных строк; x – исходный сигнал; a_{n-1} и d_{n-1} – низкочастотная и высокочастотная составляющие сигнала x .

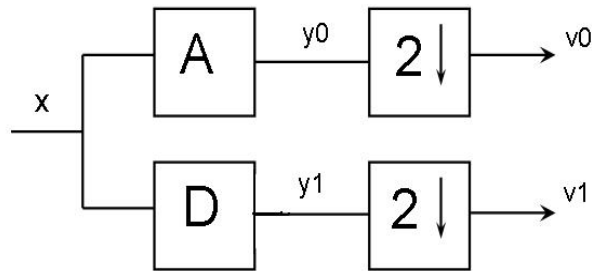


Рис. 4.8. ДВП. Процедура анализа:

A, D – фильтрующие матрицы (низкочастотная и высокочастотная),
 $2\downarrow$ – дециматор, $v0$ и $v1$ – a_{n-1} и d_{n-1} соответственно

Так как строки комбинированной матрицы (состоящей из A_n, D_n) взаимно ортогональны, так же, как и столбцы, то квадратная матрица инвертируема. Процедура инвертирования в этом случае соответствует процедуре транспонирования матрицы. Полученная матрица $[A^T \ D^T]$ представляет собой банк фильтров синтеза сигнала. Умножение этой матрицы на исходную справа дает нам единичный вектор I . Следовательно, мы имеем ортогональный банк фильтров синтеза – анализа (рис. 4.9).

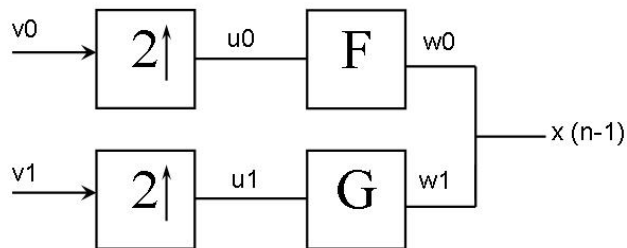


Рис. 4.9. ДВП. Процедура анализа:

$v0$ и $v1$ – a_{n-1} и d_{n-1} соответственно, $2\uparrow$ – интерполятор,
 F и G – добавляющий и вычитающий фильтры соответственно

Интерполятор представляет собой узел, который вставляет между двумя значениями сигнала третье – нуль. Работа фильтров F и G восстанавливает исходный сигнал x . Таким образом, две эти схемы вместе образуют банк фильтров ДВП.

Аналогично появлению в свое время быстрого ПФ (БПФ), появилось **быстрое вейвлет – преобразование (БВП, FWT)**. БВП вытекает из классического ДВП рекурсивным применением последнего, что возможно благодаря принципу кратномасштабного анализа (рис. 4.10).

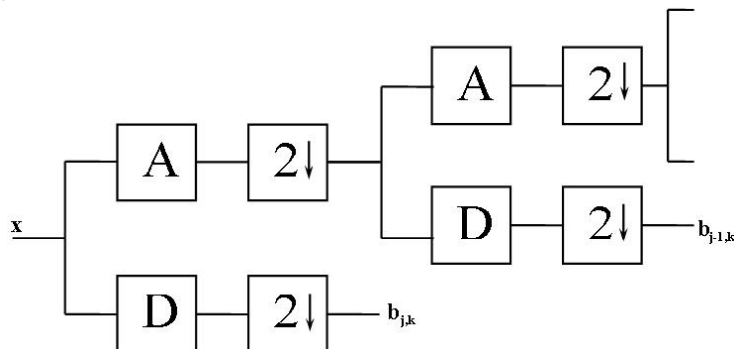


Рис. 4.10. БВП. Логарифмическое дерево банков фильтров

Сигнал x поступает на вход первого банка фильтров в дереве. После преобразования и децимации коэффициенты ВЧ фильтра $b_{j,k}$ сохраняются, так как они являются выходом на своем уровне «логарифмического дерева». Выходы же НЧ фильтра после децимации, когда в выходном сигнале остались только четные компоненты, поступают на вход следующего банка фильтров, где процедура повторяется. Выходом второго уровня являются коэффициенты $b_{j+1,k}$. Эта схема будет повторяться $L = \log_2 N$ раз, где N это число компонент в исходном сигнале. В итоге мы получим $L-1$ векторов с ВЧ фильтров и общее среднее – выход последнего НЧ фильтра, то есть полную декомпозицию сигнала. Применение такого рекурсивного алгоритма и есть БВП. Это преобразование подходит для любого сигнала, длина которого есть степень двойки, а также для любых вейвлетов, относящихся к классу ортогональных. Следует также добавить, что БВП обладает и обратным преобразованием, которое строится с использованием инверсного ДВП, когда в соответствие каждому блоку анализа на каждом уровне ставится блок синтеза. Таким образом, имеет место обратная рекурсия. Впервые такая связь между вейвлетами и фильтрами была открыта Стивеном Маллатом, поэтому в его честь этот алгоритм называют «алгоритмом Маллата» или «пирамидой Маллата».

Другим подходом к реализации БВП является *лифтинговая схема (lifting scheme)*. Подобная реализация БВП позволяет конструировать очень многие типы вейвлетов, включая ортогональные, биортогональные, вейвлет Хаара и другие, вне зависимости от преобразования Фурье.

Лифтинговая схема, как и алгоритм Маллата, включает в себя два блока: блок анализа и блок синтеза сигнала. Рассмотрим блок анализа (рис. 4.11).

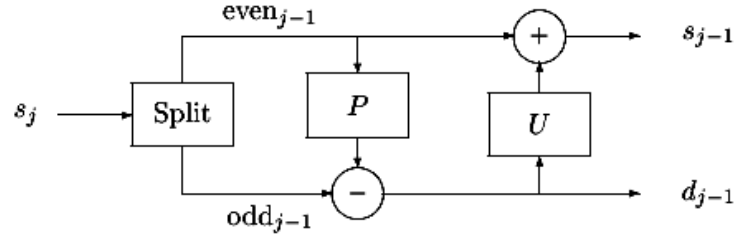


Рис. 4.11. Лифтинговая схема, блок анализа

На рисунке показано, что вектор входных значений s_j поступает на вход блока *разделения* (*Split*), который представляет собой устройство разбиения входного сигнала на две подпоследовательности по определенному закону с целью максимально «декоррелировать» сигнал. Одним из необходимых условий при проведении такой операции является возможность обратного восстановления сигнала из двух его подпоследовательностей, что используется на этапе анализа в *блоке объединения* (*Merge*) (рис. 4.12). Достаточно часто применяется разделение сигнала на две последовательности, состоящие из четных (*even*) и нечетных (*odd*) отсчетов исходного сигнала. Такая процедура получила название *Lazy wavelet Transform* – «ленивое» (не требующее больших усилий) вейвлет преобразование, которое задается выражением:

$$(even_{j-1}; odd_{j-1}) := Split(s_j) \quad (4.8)$$

Следующий блок в схеме анализа – это *блок предсказания* (*Predict*). На предыдущем шаге мы разделили исходный сигнал на две подпоследовательности четных и нечетных отсчетов. Если структура сигнала имеет локальную корреляцию, то четные и нечетные отсчеты будут сильно коррелированы. То есть, другими словами, имея одну подпоследовательность, должна быть возможность предсказать другую с заданной точностью. Всегда используется подпоследовательность четных отсчетов, чтобы предсказать подпоследовательность нечетных. В данном случае (вейвлет Хаара) нечетный отсчет $s_{j;2l+1}$ будет использовать своего левого соседа $s_{j;2l}$ как предсказание. Тогда уточняющая разность $d_{j-1;l}$ будет являться разностью нечетного отсчета и предсказания:

$$d_{j-1;l} = s_{j;2l+1} - s_{j;2l} \quad (4.9)$$

Таким образом, операция предсказания определяется следующим выражением:

$$d_{j-1} = odd_{j-1} - P(even_{j-1}) \quad (4.10)$$

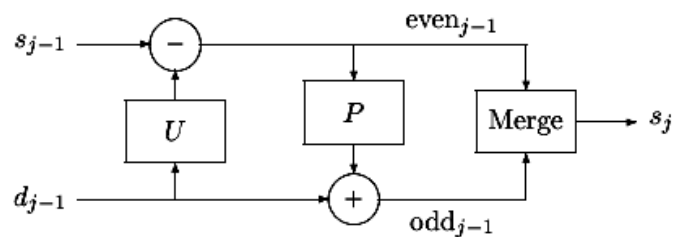


Рис. 4.12. Лифтинговая схема, блок синтеза

И, наконец, последний блок – *блок обновления (Update)*. Коэффициенты, полученные на предыдущем этапе, необходимы чтобы «поднять» или «подтянуть» («обновить») оставшуюся часть отсчетов сигнала, так как вторая половина будет сохранена и анализироваться не будет. После процедуры четная подпоследовательность будет содержать информацию обо всем сигнале, так как, используя оператор предсказания, множество s будет дополняться информацией о множестве d , поскольку к этому моменту вместо множества d в памяти хранятся коэффициенты d^* преобразования (характеризующие множество d). Обновление четной подпоследовательности определяется следующим выражением:

$$s_{j-1} = even_{j-1} + U(d_{j-1}). \quad (4.11)$$

Получаемые коэффициенты s будут рекурсивно поступать на вход схемы или переходить на более низкий уровень точно такой же схемы, с целью произвести полную декомпозицию сигнала. Процедура будет повторяться до тех пор, пока множество s не будет содержать единственный последний коэффициент s_{00} , который теперь будет именоваться – *общее среднее*.

3. Задание

1. Ознакомьтесь с теоретической частью.
 2. Для заданного сигнала в соответствии с вариантом реализуйте БВП с помощью алгоритма Маллата или используя лифтинговую схему (язык программирования C, C++).
 3. Пояснить результаты работы программы.
 4. Промоделируйте данную работу в MATLAB, сравните полученные результаты.
 5. Напишите отчет.
- Содержание отчета:
- исходные данные (заданный сигнал);
 - краткое описание алгоритма работы программы;
 - результаты работы программы;
 - результаты моделирования в MATLAB;

- анализ и пояснение полученных результатов;
- выводы.

4. Контрольные вопросы

1. Назовите основные недостатки преобразования Фурье при анализе нестационарных сигналов.
2. Поясните суть оконного преобразования Фурье, его преимущества по сравнению с преобразованием Фурье и недостатки по сравнению с вейвлет-преобразованием.
3. Поясните основные положения вейвлет-анализа.
4. Назовите основные этапы реализации алгоритма Маллата.
5. Назовите основные этапы реализации лифтинговой схемы.

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. Прэтт 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. Систематические 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.
11. Гонсалес Р., Вудс Р. Цифровая обработка изображений. - М.: Техносфера, 2005.
12. Сэломон Д. Сжатие данных, изображений и звука: Учеб. пособие для вузов. - М.: Техносфера, 2004.
13. Уэлстид С. Фракталы и вейвлеты для сжатия изображений в действии: Учеб. пособие. - М.: Триумф. 2003.

Министерство образования Республики Беларусь
Учреждение образования
«Белорусский государственный университет
информатики и радиоэлектроники»

Кафедра электронных вычислительных машин

Р.Х. Садыхов, М.М. Лукашевич

ЛАБОРАТОРНЫЙ ПРАКТИКУМ

по дисциплинам «Цифровая обработка сигналов и изображений»
и «Методы и средства обработки изображений» для студентов
специальностей I-40 02 01 «Вычислительные машины, системы и сети»
и I-40 01 01 «Программное обеспечение информационных технологий»
всех форм обучения

В 2-х частях

Часть 2

Минск 2007

Содержание

Общие теоретические сведения.....	4
1. Лабораторная работа №1. Нейронная сеть Хопфилда.....	10
2. Лабораторная работа №2. Многослойный персептрон.....	15
3. Лабораторная работа №3. Сеть РБФ.....	23
4. Лабораторная работа №4. Конкурентная нейронная сеть.....	27
Литература.....	31

НЕЙРОННЫЕ СЕТИ

Общие теоретические сведения

Приведем одно из определений искусственных нейронных сетей [1]:

Искусственная нейронная сеть (ИНС) – это существенно параллельно распределенный процессор, который обладает способностью к сохранению и репрезентации опытного знания. Она сходна с мозгом в двух аспектах:

- знание приобретается сетью в процессе обучения;
- для сохранения знания используются силы межнейронных соединений, называемые также синаптическими соединениями.

Работа нейронной сети (НС) состоит в преобразовании входных сигналов во времени, в результате чего меняется внутреннее состояние сети и формируются выходные воздействия. Обычно НС оперирует цифровыми, а не символьными величинами. Большинство моделей НС требуют обучения. В общем случае, **обучение** — это такой выбор параметров сети, при котором сеть лучше всего справляется с поставленной проблемой. Обучение — это задача многомерной оптимизации и для ее решения существует множество алгоритмов.

Современные искусственные НС демонстрируют такие ценные свойства, как:

1. **Обучаемость.** Выбрав одну из моделей НС, создав сеть и выполнив алгоритм обучения, мы можем обучить сеть решению задачи, которая ей по силам.

2. **Способность к обобщению.** После обучения сеть становится нечувствительной к малым изменениям входных сигналов (шуму или вариациям входных образов) и дает правильный результат на выходе.

3. **Способность к абстрагированию.** Если предъявить сети несколько искаженных вариантов входного образа, то сеть сама может создать на выходе идеальный образ, с которым она никогда не встречалась.

К задачам, успешно решаемым НС на данном этапе их развития относятся:

- распознавание зрительных, слуховых образов;
- ассоциативный поиск информации и создание ассоциативных моделей, синтез речи, формирование естественного языка;
- формирование моделей и различных нелинейных и трудно описываемых математически систем, прогнозирование развития этих систем во времени;
- системы управления и регулирования с предсказанием;
- разнообразные конечные автоматы: системы массового обслуживания и коммутации, телекоммуникационные системы;
- принятие решений и диагностика, исключаящие логический вывод, особенно в областях, где отсутствуют четкие математические модели: в медицине, криминалистике, финансовой сфере.

На рис. 1.1 показана структура пары типичных биологических нейронов. Дендриты идут от тела нервной клетки к другим нейронам, где они принимают сигналы в точках соединения, называемых синапсами. Принятые синапсом входные сигналы подводятся к телу нейрона. Здесь они суммируются, причем одни входы стремятся возбудить нейрон, другие – воспрепятствовать его возбуждению. Когда суммарное возбуждение в теле нейрона превышает некоторый порог, нейрон возбуждается, посылая по аксону сигнал другим нейронам. У этой основной функциональной схемы много усложнений и исключений, тем не менее большинство искусственных нейронных сетей моделируют лишь эти простые свойства.

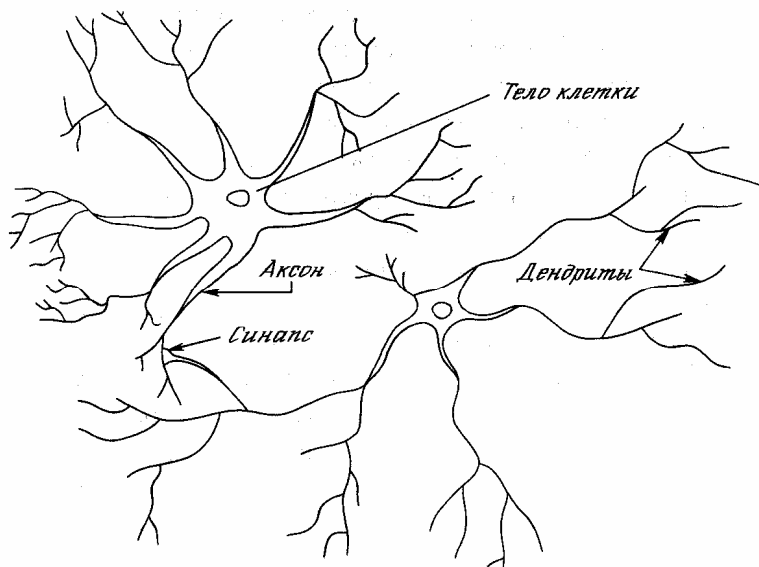


Рис. 1.1. Биологический нейрон

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

$$y = f\left(\sum_{i=1}^n w_i x_i + T\right), \quad (1.1)$$

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 1.2. Каждому i -му входу нейрона соответствует весовой коэффициент w_i (синапс), который характеризует силу синаптической связи по аналогии с биологическим нейроном.

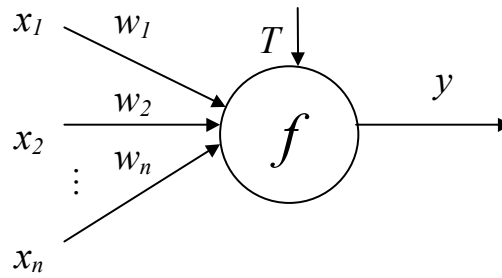


Рис. 1.2. Искусственный нейрон

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 [2] вводят дополнительный вход нейрона x_0 , всегда равный 1, и обозначают порог как его вес w_0 . Это позволяет упростить выражение (1.1) 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. 1.2)

$$y = \frac{1}{(1 + e^{-x})}; \quad (1.2)$$

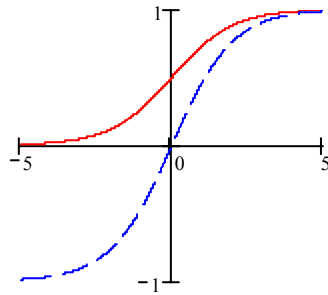


Рис. 1.2. Сигмоидные функции активации

5) биполярная сигмоидная функция (см. рис. 1.2)

$$y = \frac{2}{(1 + e^{-x})} - 1; \quad (1.3)$$

6) гиперболический тангенс (рис. 1.3)

$$y = th(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}}. \quad (1.4)$$

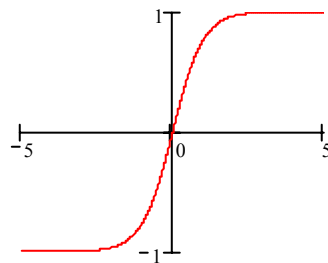


Рис. 1.3. Гиперболический тангенс

Для классификаторов чаще всего применяют функции (1.2) – (1.4), 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. 1.4).

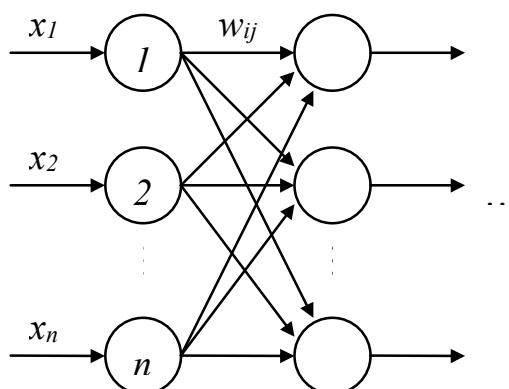


Рис. 1.4. Распределительный слой

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.

Для лучшего понимания данного вопроса приведем одну из возможных классификаций НС в зависимости от различных характеристик [2]:

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 входное 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, двунаправленная ассоциативная память, рекуррентные НС);
 - гибридные НС (НС встречного распространения, НС с радиально-базисной функцией активации).

Лабораторная работа №1

Нейронная сеть Хопфилда

1. Цель работы

Изучение топологии, алгоритма функционирования сети Хопфилда.

2. 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. 2.1. 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.

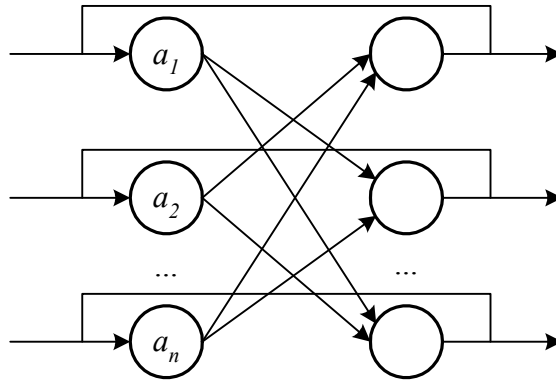


Рис. 2.1. Топология сети Хопфилда

Обучение. Обучение сети осуществляется в соответствии с соотношением

$$w_{ij} = \begin{cases} \sum_{k=1}^m a_i^k a_j^k, & i \neq j, \text{ для } i, j = \overline{1, n}, \\ 0, & i = j \end{cases} \quad (2.1)$$

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

$$W = \begin{bmatrix} w_{11} & w_{12} & \dots & w_{1n} \\ w_{21} & w_{22} & \dots & w_{2n} \\ \dots & \dots & \dots & \dots \\ w_{n1} & w_{n2} & \dots & w_{nn} \end{bmatrix}. \quad (2.2)$$

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 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

$$a_i(t+1) = f\left(\sum_{j=1}^n w_{ij} a_j(t)\right), \quad (2.3)$$

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

$$f(x) = \begin{cases} 1 & x > 0, \\ -1 & x \leq 0. \end{cases} \quad (2.4)$$

При работе в синхронном режиме на один такт работы сети все нейроны одновременно меняют состояние по формуле (2.4). 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 (2.4) 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

$$m = \frac{n}{2 \ln n + \ln \ln n}, \quad (2.5)$$

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

3. Задание

1. Ознакомьтесь с теоретической частью.
2. На языке C, C++ напишите программу, реализующую нейронную сеть Хопфилда.
3. Произведите обучение сети Хопфилда на заданный тип образов. Для запоминания в соответствии с вариантом задано 3 образа (бинарные изображения размером 10×10).
4. Подайте на вход сети ряд тестовых образов, в которые внесено зашумление (процент зашумления образа – 10%, 20%, 30%, 35%, 40%, 45%, 50%, 60%, 70%, 80%, 90%, 100%). Тестовых образов должно быть не менее 10 для каждого из классов с одним и тем же процентом зашумления.

5. Проанализируйте результаты, при каком проценте зашумления тестовые образы распознаются верно.

6. Напишите отчет.

Содержание отчета:

- топология сети Хопфилда;
- описание алгоритма работы сети;
- тестируемые образы (3 образа);
- искаженные образы (процент зашумления образа – 10%, 20%, 30%, 35%, 40%, 45%, 50%, 60%, 70%, 80%, 90%, 100%);
- результаты распознавания, статистика;
- выводы.

Таблица 2.1

Варианты задания

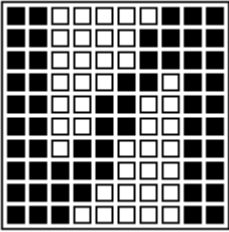
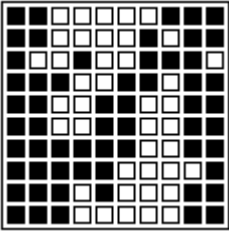
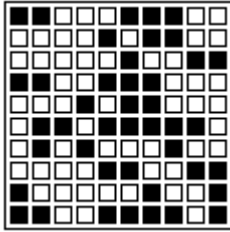
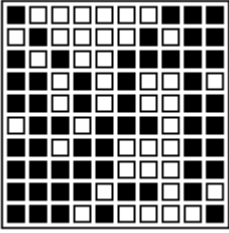
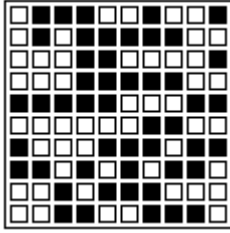
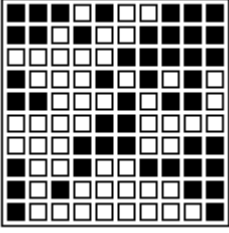
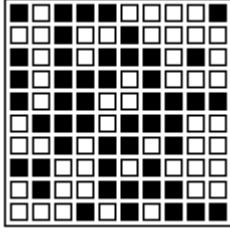
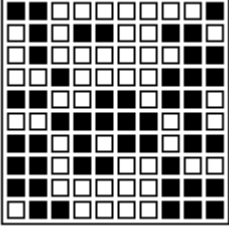
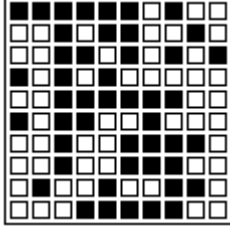
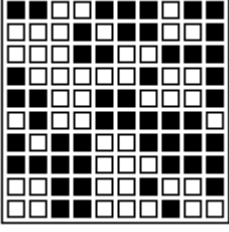
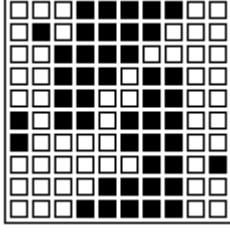
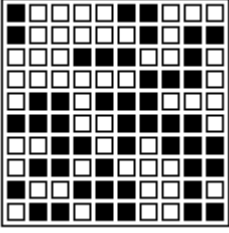
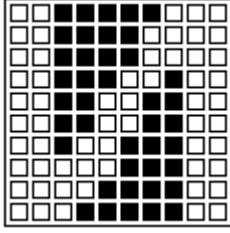
№ варианта	1-ый тестируемый образ	2-ой тестируемый образ	3-ий тестируемый образ
1	«А»	«И»	«Р»
2	«Б»	«К»	«С»
3	«В»	«Л»	«Т»
4	«Г»	«М»	«У»
5	«Д»	«Н»	«Х»
6	«Е»	«О»	«Ш»
7	«З»	«П»	«Ь»

4. 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.

Таблица 2.2

Пример задания тестируемого и искаженных образов

Тестируемый образ	Процент зашумления образа	Вид искаженного образа	Процент зашумления образа	Вид искаженного образа
	10%		50%	
	20%		60%	
	30%		70%	
	35%		80%	
	40%		90%	
	45%		100%	

Лабораторная работа №2

Многослойный персептрон

1. Цель работы

Изучение топологии, алгоритма функционирования многослойного персептрона.

2. 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. Для этого достаточно одного скрытого слоя нейронов с сигмоидной функцией активации (1.2) – (1.4), т.е. многослойный персептрон обычно состоит из 3 слоев: первого распределительного, второго скрытого и третьего выходного (рис. 3.1).

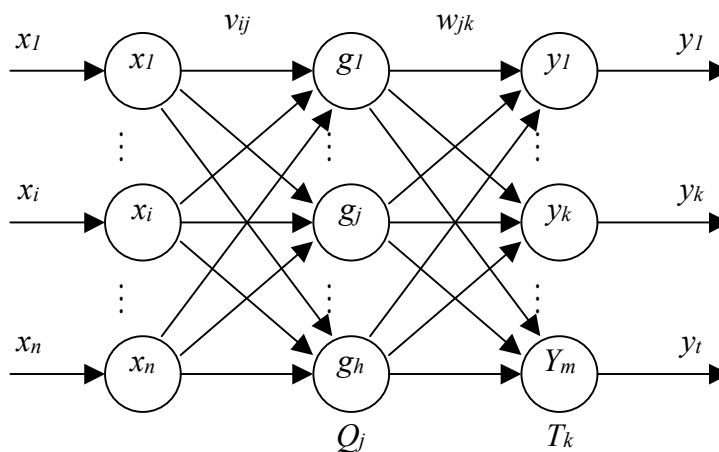


Рис. 3.1. Многослойный персептрон

Такая сеть имеет n входов и n нейронов распределительного слоя, h нейронов скрытого слоя и t выходных нейронов. Используются две матрицы весов: скрытого слоя v размером $n \times h$ и выходного слоя w размером $h \times t$. 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:

$$g_j = f\left(\sum_{i=1}^n v_{ij}x_i + Q_j\right), \quad (3.1)$$

$$y_k = f\left(\sum_{j=1}^h w_{jk}g_j + T_k\right). \quad (3.2)$$

В качестве функции активации используется одна из функций (1.2) – (1.4). Вид функции определяет диапазон чисел, в котором работает сеть. В дальнейшем будет использоваться сигмоидная функция (1.2), 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 $x^r = (x_1^r, x_2^r, \dots, x_i^r, \dots, x_n^r)$ 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 персептрона 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. 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 (3.1), (3.2).

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:

$$w_{jk}(t+1) = w_{jk}(t) + \alpha \frac{\partial E}{\partial w_{jk}(t)}, \quad (3.3)$$

$$T_k(t+1) = T_k(t) + \alpha \frac{\partial E}{\partial T_k(t)}, \quad (3.4)$$

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

$$E = \frac{1}{2} \sum_{k=1}^m (y_k^r - y_k)^2. \quad (3.5)$$

Ошибка k -го нейрона выходного слоя определяется как

$$d_k = \frac{\partial E}{\partial y_k} = y_k^r - y_k. \quad (3.6)$$

Выразим производные из формул (3.3), (3.4) через легко вычисляемые величины. Определим взвешенную сумму, аргумент функции активации как

$$S_k = \sum_{j=1}^h w_{jk} g_j + T_k. \quad (3.7)$$

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

$$\frac{\partial E}{\partial w_{jk}} = \frac{\partial E}{\partial y_k} \frac{\partial y_k}{\partial S_k} \frac{\partial S_k}{\partial w_{jk}}, \quad (3.8)$$

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

$$\frac{\partial E}{\partial w_{jk}} = d_k f'(S_k) g_j. \quad (3.9)$$

Аналогично (3.8), с учетом того, что $\frac{\partial S_k}{\partial T_k} = 1$, получаем

$$\frac{\partial E}{\partial T_k} = \frac{\partial E}{\partial y_k} \frac{\partial y_k}{\partial S_k} \frac{\partial S_k}{\partial T_k} = d_k f'(S_k). \quad (3.10)$$

Веса и пороги скрытого слоя также корректируются по формулам, аналогичным (3.3), (3.4), с учетом (3.9) и (3.10). При этом главной трудностью является определение ошибки нейрона скрытого слоя. Эту ошибку явно определить по формуле, аналогичной (3.6), нельзя, однако существует возможность рассчитать ее через ошибки нейронов выходного слоя (отсюда произошло название алгоритма обратного распространения ошибки):

$$e_j = \frac{\partial E}{\partial g_j} = \sum_{k=1}^m \frac{\partial E}{\partial y_k} \frac{\partial y_k}{\partial S_k} \frac{\partial S_k}{\partial g_j} = \sum_{k=1}^m d_k f'(S_k) w_{jk}. \quad (3.11)$$

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

$$f'(S_k) = \left(\frac{1}{(1 + e^{-S_k})} \right)' = f(S_k)(1 - f(S_k)) = y_k(1 - y_k). \quad (3.12)$$

Аналогично:

$$\begin{aligned} (th(S_k))' &= 1 - y_k^2; \\ \left(\frac{2}{(1 + e^{-x})} - 1 \right)' &= \frac{(1 - y_k^2)}{2}. \end{aligned}$$

Таким образом, можно записать окончательные выражения (3.3), (3.4) для двух слоев, использующих сигмоидную функцию:

$$w_{jk} = w_{jk} + \alpha y_k (1 - y_k) d_k g_j, \quad (3.13)$$

$$T_k = T_k + \alpha y_k (1 - y_k) d_k, \quad (3.14)$$

$$v_{ij} = v_{ij} + \beta g_j (1 - g_j) e_j x_i, \quad (3.15)$$

$$Q_j = Q_j + \beta g_j (1 - g_j) e_j. \quad (3.16)$$

В эти формулы вводится дополнительный параметр – β – скорость обучения скрытого слоя, который может отличаться от аналогичного параметра для выходного слоя. Рекомендуются изменять скорости обучения обратно пропорционально количеству шагов 2 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

$$\max |d_k| < D, k = \overline{1, m}, r = \overline{1, p}, \quad (3.17)$$

где D – достаточно маленькая константа – величина максимальной ошибки, которую требуется достичь в процессе обучения. Если условие (3.17) 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. Персептрон 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

$$h \sim \sqrt{\frac{p}{n}}. \quad (3.18)$$

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 (3.5), а гарантируется определение только локального минимума. Проблемы, возникающие в процессе градиентного спуска, можно проанализировать на примере функции ошибки, схематически изображенной на рис. 3.2.

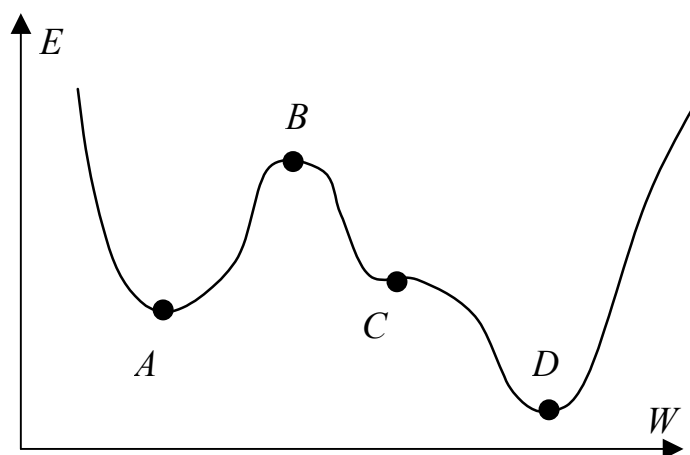


Рис. 3.2. Схематическая функция ошибки сети

На рис. 3.2 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 (3.17). 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 (3.3) в момент времени t будет вычисляться как

$$\Delta w(t) = \alpha \frac{\partial E}{\partial w} + \mu \Delta w(t-1), \quad (3.19)$$

где μ - параметр, определяющий величину влияния момента. С использованием (3.19) 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.

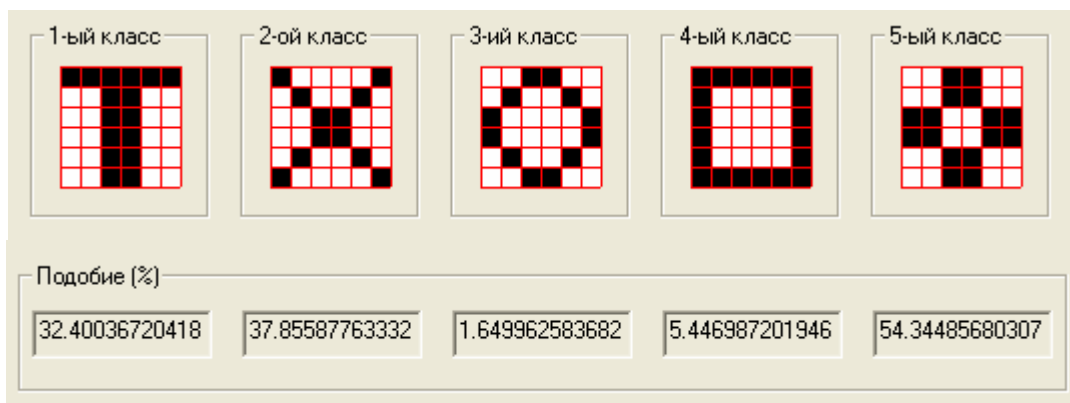
Этот и другие алгоритмы оптимизации обучения персептрона позволяют улучшить работу сети в условиях плохой сходимости. Однако они усложняют процесс обучения, не гарантируя в то же время полного успеха во всех случаях. Успех обучения классификатора зависит от самого алгоритма обучения и качества обучающей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

3. Задание

1. Ознакомьтесь с теоретической частью.
2. На языке C, C++ напишите программу, реализующую многослойный персептрон.
3. Произведите обучение многослойного персептрона. Исходные данные - 5 классов образов, размер идеального образа 6×6 (в соответствии с вариантом).
4. Подайте на вход сети ряд тестовых образов, по 3 зашумленных образа каждого из 5 классов.
5. Проанализируйте результаты работы программы, которые должны иметь следующий вид:
 - вывести распознаваемый зашумленный образ.



- вывести процент подобия распознаваемого зашумленного образа по отношению к каждому из 5 классов.



- вывести количество шагов, затраченных на обучение сети на заданное количество классов.

6. Напишите отчет.

Содержание отчета:

- топология многослойного персептрона;
- основные формулы обучения и воспроизведения;
- идеальные образы для обучения многослойного персептрона;
- тестовые зашумленные образы;
- результаты воспроизведения: процент подобия по отношению к каждому из классов, количество шагов, затраченных на обучение;
- результаты сравнения многослойного персептрона с нейронной сетью Хопфилда;
- выводы: преимущества и недостатки многослойного персептрона.

Таблица 3.1

Варианты задания

№ варианта	1-ый класс	2-ой класс	3-ий класс	4-ый класс	5 класс
1	2	3	4	5	7
2	N	F	I	P	D
3	\wedge	\vee	\exists	\subset	\supset
4	\otimes	\oplus	\times	\div	\pm
5	\leq	\geq	\neq	\equiv	\approx
6	L	U	T	O	K
7	\rightarrow	\leftrightarrow	\leftarrow	\downarrow	\curvearrowright

4. 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 № 3

Сеть РБФ

1. Цель работы

Изучение топологии, алгоритма функционирования сети РБФ.

2. Теоретические сведения

Сеть РБФ (радиальная базисная функция) является аналогом многослойного персептрона (рис. 4.1). 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.

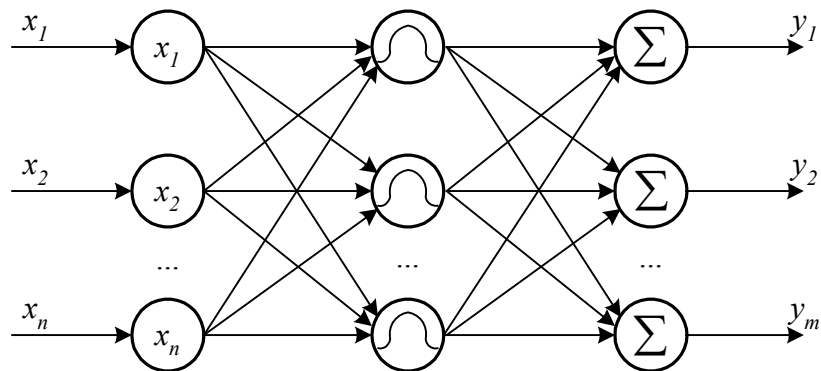


Рис. 4.1. Топология сети РБФ

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:

$$g_j = \exp\left(\frac{-|x - t^j|^2}{\sigma_j^2}\right), \quad (4.1)$$

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. 4.1). 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.

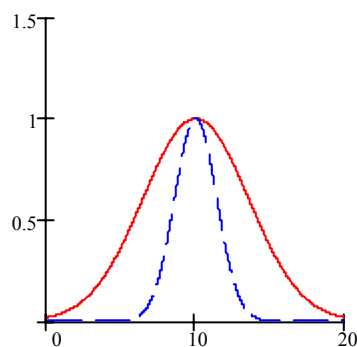


Рис. 4.2. Примеры функций РБФ ячеек с одинаковым центром и разным разбросом

Выходной слой РБФ сети обычно состоит из суммирующих ячеек

$$y_k = \sum_{j=1}^h w_{jk} g_j. \quad (4.2)$$

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 2.2 значительно упрощаются. Формулы (3.1) и (3.2) заменяются соответственно на (4.1) и (4.2). Формула (3.13) упрощается до

$$w_{jk} = w_{jk} + \alpha d_k g_j, \quad (4.3)$$

поскольку функция активации в выходном слое сети РБФ линейная и ее производная равна 1. Для коррекции весовых коэффициентов выходных суммирующих ячеек используется только (4.3).

Контроль завершения алгоритма обучения производится аналогично.

Воспроизведение. Сеть функционирует по формулам (4.1) и (4.2).

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.

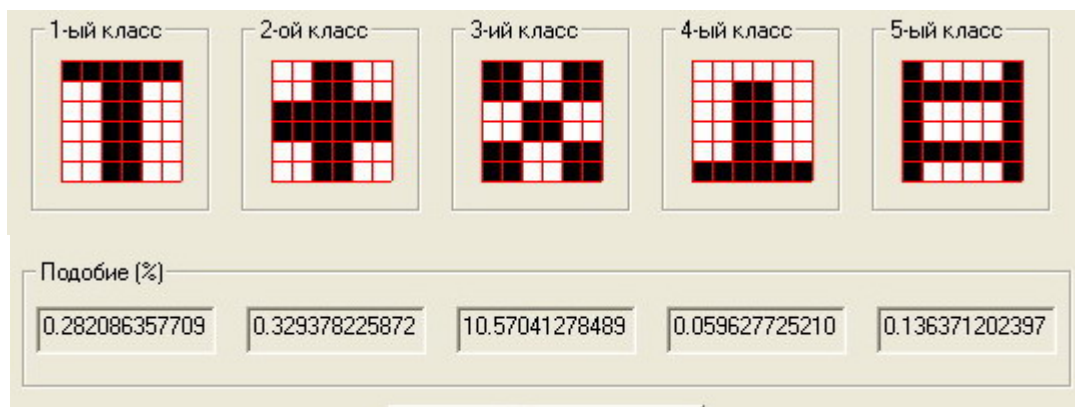
3. Задание

1. Ознакомьтесь с теоретической частью.
2. Написать программу на С, С++, реализующую сеть РБФ.
3. Произведите обучение сети РБФ. Исходные данные - 5 классов образов, размер идеального образа 6×6 (в соответствии с вариантом).
4. Подайте на вход сети ряд тестовых образов, по 3 зашумленных образа каждого из 5 классов.
5. Проанализируйте результаты работы программы, которые должны иметь следующий вид:

- вывести распознаваемый зашумленный образ;



- вывести процент подобия распознаваемого зашумленного образа по отношению к каждому из 5 классов;



- вывести количество шагов, затраченных на обучение сети на заданное количество классов.

6. Напишите отчет.

Содержание отчета:

- топология сети РБФ;
- основные формулы обучения и воспроизведения;
- идеальные образы для обучения сети РБФ;
- тестовые зашумленные образы;
- результаты воспроизведения: процент подобия по отношению к каждому из классов, количество шагов, затраченных на обучение;
- результаты сравнения сети РБФ с многослойным персептроном;
- выводы: преимущества и недостатки сети РБФ.

Таблица 4.1

Варианты задания

№ варианта	1-ый класс	2-ой класс	3-ий класс	4-ый класс	5 класс
1	2	3	4	5	7
2	N	F	I	P	D
3	\wedge	\vee	\exists	\subset	\supset
4	\otimes	\oplus	\times	\div	\pm
5	\leq	\geq	\neq	\equiv	\approx
6	L	U	T	O	K
7	\rightarrow	\leftrightarrow	\leftarrow	\downarrow	\lrcorner

4. 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 № 4

Конкурентная нейронная сеть

1. Цель работы

Изучение топологии, алгоритма функционирования конкурентной нейронной сети.

2. 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. 5.1).

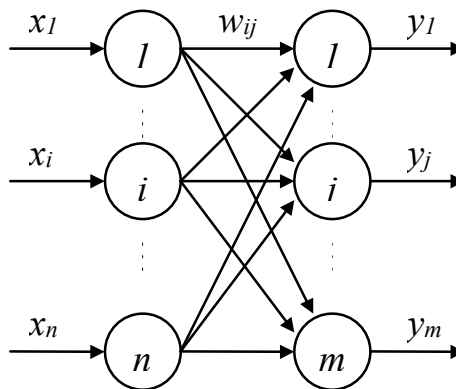


Рис. 5.1. Конкурентная нейронная сеть

Первый слой является распределительным. Нейроны второго слоя функционируют по формуле

$$y_j = \sum_{i=1}^n w_{ij} x_i = |w^j| |x| \cos \alpha, \quad (5.1)$$

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 (5.1) максимально. Для этого нейрона синаптические связи усиливаются по формуле

$$w_{ij}(t+1) = w_{ij}(t) + \beta(x_i - w_{ij}(t)), \quad (5.2)$$

где β – скорость обучения.

Смысл этой формулы в том, что вектор весовых коэффициентов нейрона – победителя “поворачивается” в сторону входного вектора, тем самым активность 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 (5.1) для нормированных векторов будет выглядеть как

$$y_j = \sum_{i=1}^n w_{ij} x_i = \cos \alpha, \quad (5.3)$$

а выражение (5.2):

$$w_{ij}(t+1) = \frac{w_{ij}(t) + \beta(x_i - w_{ij}(t))}{|w^j(t) + \beta(x - w^j(t))|}. \quad (5.4)$$

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 евклидового расстояния между входным и весовым вектором и количеством побед данного нейрона f_j :

$$d_v = \min_j (|x - w^j| f_j). \quad (5.5)$$

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

Конкурентное обучение продолжается до тех пор, пока максимум евклидового 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 (5.2) 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 ростом 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:

$$w_{ij}(t+1) = w_{ij}(t) + \beta h(t, k, j)(x_i - w_{ij}(t)). \quad (5.6)$$

В качестве функции $h(t, k, j)$ может использоваться гауссовый колокол (4.1) с параметром σ , зависящим от времени или функция вида “мексиканская шляпа”.

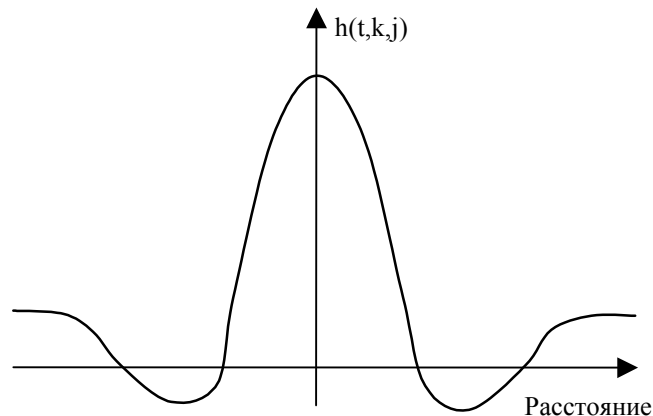


Рис. 5.2. Пример функции расстояния в сетях Кохонена

В результате модификации конкурентного обучения сеть Кохонена не только кластеризует входные примеры, но и упорядочивает их в виде одномерной или двухмерной 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.

3. Задание

1. Ознакомьтесь с теоретической частью.
2. Написать программу на C, C++, реализующую конкурентную нейронную сеть.
3. Обучить конкурентную сеть с использованием правила (5.5) на количество образов, превышающих количество нейронов сети. Рекомендуется использовать нормированные векторы. Исходные данные - 5 классов образов, размер идеального образа 6×6 (в соответствии с вариантом).
4. Убедиться, что похожие образы были спроецированы сетью в один кластер (подача их на вход активизирует один и тот же нейрон).
5. Подать на вход тестовые образы, отличные от образов из обучающей выборки. Сделать выводы.
6. Напишите отчет.

Содержание отчета:

- топология конкурентной нейронной сети;
- основные формулы обучения и воспроизведения;
- идеальные образы для обучения сети;
- тестовые зашумленные образы;
- результаты воспроизведения;
- результаты сравнения конкурентной нейронной сети с сетью РБФ и многослойным персептроном;
- выводы: преимущества и недостатки конкурентной нейронной сети.

Таблица 5.1

Варианты задания

№ варианта	1-ый класс	2-ой класс	3-ий класс	4-ый класс	5 класс
1	2	3	4	5	7
2	N	F	I	P	D
3	\wedge	\vee	\exists	\subset	\supset
4	\otimes	\oplus	\times	\div	\pm
5	\leq	\geq	\neq	\equiv	\approx
6	L	U	T	O	K
7	\rightarrow	\leftrightarrow	\leftarrow	\downarrow	\lceil

4. 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. Aleksander I., Morton H. An Introduction to Neural Computing. — London: Chapman&Hall, 1990.
2. Головкин В.А. Нейронные сети: обучение, организация и применение. Учеб. пособие для вузов.— М.:ИПРЖР, 2001. — 256 с.
3. Bishop C.M. Neural Networks for Pattern Recognition.— Oxford: Clarendon press, 1995.— 482 p.
4. Hopfield J.J. Neural networks and physical systems with emergent collective computational abilities // Proc. Natl. Acad. Sci. USA.— 1982.— Vol. 79.— P. 2554.
5. Kohonen T. Self-organization and associative memory. — Springer-Verlag, 1989.— 312 p.
6. Kohonen T. Self-organized formation of topologically correct feature maps // Biol. Cybernetics.— 1982.— Vol. 43.— P. 56-69.
7. Kohonen T. Self-organizing maps. —Springer-Verlag, 1995.— 362 p.
8. 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.
9. Хайкин С. Нейронные сети: полный курс, 2-е изд.: Пер. с англ. — М. : Издательский дом «Вильямс», 2006. — 1104 с.
- 10.Ежов А.А., Шумский С.А. Нейрокомпьютинг и его применение в экономике и бизнесе. —М.: Мир, 1998.— 222 с.