УДК 004.942

Я. А. Калиновский[1], Ю. Е. Бояринова[1,2],
Я. В. Хицко[2], А. С. Сукало[3]
[1]Институт проблем регистрации информации НАН Украины
ул. Н. Шпака, 2, 03113 Киев, Украина
[2]Национальный технический университет Украины «КПИ им. Игоря Сикорского»
проспект Победы, 37, 03113 Киев, Украина
[3]Национальный университет водного хозяйства и природопользования
ул. Соборная, 11, 33028 Ровно, Украина

Структура алгоритма быстрой двухмерной свертки с помощью изоморфных гиперкомплексных числовых систем

Рассмотрены вопросы построения алгоритмов быстрой двухмерной свертки массивов различной размерности. Алгоритмы строятся на основе представления массивов данных в изоморфных гиперкомплексных числовых системах, полученных умножением размерности систем двойных чисел и ортогональных двойных чисел, что дает возможность простого по структуре перехода от одной системы к другой. Это приводит к уменьшению количества операций, необходимых для выполнения двухмерных линейных сверток массивов различной величины. Изучен эффект уменьшения количества операций. Исследования выполнены с помощью системы аналитических вычислений Maple.

***Ключевые слова**: гиперкомплексная числовая система, двухмерная свертка, изоморфизм гиперкомплексных числовых систем, двойные числа, ортогональные двойные числа, оператор изоморфизма, Maple.*

Введение

При математическом моделировании линейных систем приходится многократно выполнять линейную свертку дискретных сигналов. Радиолокационные системы, системы звуковой локации, обработка сейсмической информации, неразрушающий контроль и компьютерная томография, обработка изображений — этот, даже очень далекий от полноты перечень областей использования линейной свертки дискретных сигналов, дает представление о важности данной задачи.

Так как сложность вычисления линейной свертки быстро увеличивается с ростом длины свертываемых массивов и их размерности, то используются методы «быстрых» вычислений свертки. Одним из наиболее распространенных методов является свертка с использованием алгоритмов быстрого преобразования Фурье (БПФ), в основе которых лежит декомпозиция исходной задачи большой размерности в большое количество задач малой размерности. Отсюда весьма важным является разработка таких методов решения задач малой размерности, которые используют, возможно, меньшее число вещественных операций.

Как показано в [1], нижняя граница количества вещественных умножений при линейной свертке равна $2n-1$, где n — длина свертываемого массива. Это число может быть ориентиром при оценке качества алгоритма. В работе [2] отмечается, что «Эффективность алгоритма (свертки) определяется не только числом операций, но и такими параметрами, как число перемещений данных, стоимость вспомогательных операций, общая структурная сложность, различные возможности, представляемые используемой вычислительной системой, искусство программиста. Поэтому упорядочение алгоритмов по их действительной эффективности, выраженной временем выполнения, является весьма трудным делом, так что сравнения, основанные лишь на числе арифметических операций, должны быть «взвешены» с учетом факторов, возникающих при конкретных реализациях этих алгоритмов».

Существует большое количество методов быстрого вычисления линейной свертки. С некоторыми из них можно познакомиться в работах [1–5] и многих других. Рассмотрим кратко принципы, на которых базируются основные методы.

Алгоритм вычисления линейной свертки Кука-Тоома [1, 2, 5] основан на представлении сворачиваемых последовательностей полиномами и восстановлении результирующего полинома-произведения с помощью интерполяционной формулы Лагранжа. При этом предполагается, что одна из сворачиваемых последовательностей является постоянной. Это условие является существенным, так как в противном случае алгоритм Кука-Тоома, как, впрочем, и все остальные методы, теряет свои преимущества в количестве операций.

Алгоритм вычисления свертки по методу Винограда [1, 2, 5] основан, как и предыдущий метод Кука-Тоома, на представлении сворачиваемых последовательностей полиномами. Однако восстановление результирующего полинома-произведения производится с помощью его представления по полиномиальному модулю, который допускает разложение на полиномы низших степеней.

Для вычисления свертки преобразование Фурье интересно тем, что выполняется теорема о свертке: преобразование Фурье-свертки сворачиваемых последовательностей равно произведению преобразований Фурье сворачиваемых последовательностей.

Таким образом, чтобы найти циклическую свертку двух последовательностей, необходимо найти преобразования Фурье сворачиваемых последовательностей, покомпонентно их перемножить и подвергнуть полученную последовательность обратному преобразованию Фурье.

Как видим, основная идея применения преобразования Фурье к расчету циклической свертки заключается в переходе в другое пространство, в котором расчеты выполняются с меньшими вычислительными затратами.

Для того чтобы вычисление свертки по этой схеме было эффективным, необходимо, чтобы прямое и обратное преобразования Фурье не требовали большого объема вычислений. Однако преобразование Фурье само по себе является сверткой. В основе алгоритмов БПФ лежат методы декомпозиции исходной задачи в совокупность задач малой размерности. Именно такой подход лежит в основе часто применяемых алгоритмов Кули-Тьюки и Гуда-Томаса, структуру которых можно найти, например, в [1].

Линейная свертка с использованием гиперкомплексных числовых систем

Все вышеизложенное говорит о том, что необходимы алгоритмы с самыми разнообразными свойствами. В данном разделе будут рассмотрены алгоритмы выполнения свертки, основанные на переходе к гиперкомплексным пространствам. Основы такого подхода разработаны авторами и изложены в работах [3, 4].

Пусть необходимо выполнить одномерную линейную свертку двух числовых последовательностей длиной 2^n. Рассмотрим эти числовые последовательности как компоненты гиперкомплексных чисел, принадлежащих некоторой гиперкомплексной числовой системе (ГЧС) Γ_1 размерностью $\dim \Gamma_1 = 2^n$. Произведение этих чисел будет содержать парные произведения компонентов свертываемых числовых последовательностей. Однако они будут комбинироваться в суммы не в таком составе, как это нужно для организации компонентов свертки. Кроме того, число вещественных умножений при умножении гиперкомплексных чисел в общем случае равно 2^{2n}, это столько же, как и при прямом вычислении свертки, то есть, нет никакого выигрыша.

Таким образом, здесь возникают две проблемы: первая — это снижение количества вещественных операций при умножении гиперкомплексных чисел; вторая — организация выбора парных произведений компонентов свертки. Решение этих двух проблем позволяет синтезировать такие алгоритмы свертки, которые будут по количеству операций эффективнее прямых алгоритмов вычисления свертки.

Для решения первой проблемы можно перейти в такую ГЧС, изоморфную исходной, таблица умножения которой заполнена слабо. Такие пары ГЧС существуют и описаны в [7, 8]. Переход между такими ГЧС требует только выполнения операций сложения вещественных чисел. Решение второй проблемы зависит от конкретного вида используемых ГЧС и будет рассмотрено ниже.

Для решения первой из вышеуказанных проблем можно перейти в такую ГЧС, изоморфную исходной, таблица умножения которой заполнена слабо. Такие пары ГЧС существуют и описаны в [7, 8]. Переход между такими ГЧС требует только выполнения операций сложения вещественных чисел.

Решение второй из вышеуказанных проблем зависит от конкретного вида используемых ГЧС и будет рассмотрено далее.

Так как одномерная линейная свертка выполняется по формуле [1, 2, 5]

$$z_k = \sum_{m=-\infty}^{k} x_m y_{k-m}, \qquad (1)$$

то отсюда видно, что если рассматривается линейная свертка двух последовательностей длиной по 2^n элементов, то при прямом вычислении ее $2^{n+1}-1$ отсчетов необходимо произвести 2^{2n} вещественных умножений. С помощью гиперкомплексных числовых систем можно уменьшить это количество.

Для демонстрации принципа снижения количества вещественных умножений при умножении гиперкомплексных чисел рассмотрим случай $n=1$, то есть вычисляется линейная свертка двух числовых последовательностей — $\{x_0, x_1\}$ и $\{y_0, y_1\}$. В соответствии с (1) свертка будет иметь три отсчета:

$$\begin{aligned} z_{-1} &= x_0 y_0, \\ z_0 &= x_0 y_1 + x_1 y_0, \\ z_1 &= \qquad\quad x_1 y_1. \end{aligned} \tag{2}$$

Как видно из (2), для выполнения данной свертки необходимо выполнить 4 вещественных умножения.

Рассмотрим элементы последовательностей $\{x_0, x_1\}$ и $\{y_0, y_1\}$ как компоненты гиперкомплексных чисел, принадлежащие ГЧС $W(e,2)$ размерности 2 с таблицей умножения:

W	e_1	e_2
e_1	e_1	e_2
e_2	e_2	e_1

Пусть

$$X = x_0 e_0 + x_1 e_1, \tag{3}$$

$$Y = y_1 e_0 + y_0 e_1. \tag{4}$$

Второе число записано в инверсном виде.

В соответствии с таблицей умножения ГЧС W [7, 8]:

$$XY = (x_0 y_1 + x_1 y_0)e_0 + (x_0 y_0 + x_1 y_1)e_1. \tag{5}$$

Правая часть выражения (5) содержит все парные произведения, необходимые для формирования свертки (2). Более того, компонента при e_0 полностью совпадает с отсчетом z_0, а компонента при e_1 равна сумме остальных отсчетов. Но количество вещественных умножений также равно 4, как и в (2). Таким образом, возникают две проблемы: во-первых, уменьшить число умножений в (5), во-вторых, разделить отсчеты z_{-1} и z_1. Для уменьшения количества умножений целесообразно перейти из системы W в систему W_1 с таблицей умножения:

W_1	f_1	f_2
f_1	f_1	0
f_2	0	f_2

Для перевода чисел $X, Y \in W$ в числа $X_1, Y_1 \in W_1$ необходимо воспользоваться оператором изоморфизма этих систем [3, 7, 8]. Тогда:

$$X_1 = (x_0 + x_1) f_0 + (x_0 - x_1) f_1, \tag{6}$$

$$Y_1 = (y_0 + y_1) f_0 + (y_0 - y_1) f_1. \tag{7}$$

Такой переход требует дополнительно 4 сложения. Однако, учитывая, что обычно числовая последовательность сворачивается с постоянным ядром, переход элементов которого в W_1 сделан заранее, то можно считать, что переход из системы W в систему W_1 требует только 2 сложения.

Умножение чисел (6) и (7) в соответствии с таблицей умножения W_1 дает:

$$X_1 Y_1 = \alpha_0 f_0 + \alpha_1 f_1 = (x_0 + x_1)(y_0 + y_1) f_0 + (x_0 - x_1)(y_0 - y_1) f_1, \tag{8}$$

что требует 2 вещественных умножения.

Обратный переход из W_1 в систему в соответствии с [3, 7, 8] будет иметь вид:

$$XY = \frac{\alpha_0 + \alpha_1}{2} e_0 + \frac{\alpha_0 - \alpha_1}{2} e_1, \tag{9}$$

что также требует 2 сложения. Деление на 2 — это короткая операция, требующая только сдвига регистра. Ее можно выполнить заранее над ядром свертки. Поэтому ее можно не учитывать.

Приравнивая правые части (5) и (9), получим:

$$x_0 y_1 + x_1 y_0 = \frac{\alpha_0 + \alpha_1}{2}, \tag{10}$$

$$x_0 y_0 + x_1 y_1 = \frac{\alpha_0 - \alpha_1}{2}. \tag{11}$$

Если вычислить заранее одно из слагаемых в левой части (11), например, $x_0 y_0$, то компоненты свертки (2) будут вычисляться по следующим формулам:

$$\begin{aligned} z_{-1} &= x_0 y_0, \\ z_0 &= \frac{\alpha_0 + \alpha_1}{2}, \\ z_1 &= \frac{\alpha_0 - \alpha_1}{2} - x_0 y_0, \end{aligned} \tag{12}$$

что требует 3 умножения и 5 сложений. При расчетах по (2, 3) соответственно 4 умножения и 1 сложение. В данном случае эффект скорее отрицательный. Этот пример, как было отмечено выше, носит чисто иллюстративный характер для демонстрации идеи. Однако, как будет показано далее, в некоторых условиях эффект становится положительным.

Рассмотрим необходимый для дальнейшего изложения алгоритм свертки двух числовых последовательностей длиной 4 — $\{x_0, x_1, x_2, x_3\}$ и $\{y_0, y_1, y_2, y_3\}$, следуя работам [3, 4]. В соответствии с (1) свертка будет иметь 7 отсчетов:

$$
\begin{aligned}
z_{-3} &= x_0y_0, \\
z_{-2} &= x_0y_1 + x_1y_0, \\
z_{-1} &= x_0y_2 + x_1y_1 + x_2y_0, \\
z_0 &= x_0y_3 + x_1y_2 + x_2y_1 + x_3y_0, \\
z_1 &= \qquad\quad x_1y_3 + x_2y_2 + x_3y_1, \\
z_2 &= \qquad\qquad\qquad x_2y_3 + x_3y_2, \\
z_3 &= \qquad\qquad\qquad\qquad\quad x_3y_3.
\end{aligned} \tag{13}
$$

Для выполнения данной свертки необходимо выполнить 16 вещественных умножений и 9 сложений.

Рассмотрим элементы последовательностей $\{x_0, x_1, x_2, x_3\}$ и $\{y_0, y_1, y_2, y_3\}$ как компоненты гиперкомплексных чисел ГЧС $W^{(2)}(e,4)$ [3, 8]:

$$X = x_0e_0 + x_1e_1 + x_2e_2 + x_3e_3, \tag{14}$$

$$Y = y_0e_0 + y_1e_1 + y_2e_2 + y_3e_3. \tag{15}$$

Тогда в соответствии с таблицей умножения ГЧС $W^{(2)}(e,4)$:

$$XY = \sum_{i=0}^{3} \alpha_i e_i, \tag{16}$$

где компоненты α_i имеют вид:

$$
\begin{aligned}
\alpha_0 &= x_0y_3 + x_1y_2 + x_2y_1 + x_3y_0, \\
\alpha_1 &= x_0y_2 + x_1y_3 + x_2y_0 + x_3y_1, \\
\alpha_2 &= x_0y_1 + x_1y_0 + x_2y_3 + x_3y_2, \\
\alpha_3 &= x_0y_0 + x_1y_1 + x_2y_2 + x_3y_3.
\end{aligned} \tag{17}
$$

Для уменьшения количества умножений целесообразно перейти из системы $W^{(2)}(e,4)$ в систему $W_1^{(2)}(f,4)$ [3, 4]. Тогда для перевода гиперкомплексных чисел, принадлежащих $W^{(2)}(e,4)$, в гиперкомплексные числа, принадлежащие $W_1^{(2)}(f,4)$, необходимо воспользоваться оператором изоморфизма между этими системами [3, 4]. Тогда:

$$
\begin{aligned}
X_1 = \sum_{i=0}^{3} \beta_i f_i &= (x_0 + x_1 + x_2 + x_3) f_0 + (x_0 - x_1 + x_2 - x_3) f_1 + {} \\
&+ (x_0 + x_1 - x_2 - x_3) f_2 + (x_0 - x_1 + -x_2 + x_3) f_3
\end{aligned}. \tag{18}
$$

Аналогично и Y_1:

$$Y_1 = \sum_{i=0}^{3} \gamma_i f_i. \tag{19}$$

Такой переход требует дополнительно 16 сложений. Однако, учитывая, что обычно числовая последовательность сворачивается с постоянным ядром, переход элементов которого в $W_1^{(2)}(f,4)$ сделан заранее, то можно считать, что переход из системы $W^{(2)}(e,4)$ в систему $W_1^{(2)}(f,4)$ требует только 8 сложений.

Умножение X_1 и Y_1 в соответствии с таблицей умножения $W_1^{(2)}(f,4)$ будет

$$X_1Y_1 = \sum_{i=0}^{3} \beta_i \gamma_i f_i \qquad (20)$$

и требует всего лишь 4 умножения.

Обратный переход из $W_1^{(2)}(f,4)$ в систему $W^{(2)}(e,4)$ в соответствии с [3, 4] требует 8 сложений. Деление на 4 — это короткая операция, требующая только сдвига регистра. Поэтому ее можно не учитывать.

Далее целесообразно сначала сделать свертку последовательностей $\{x_0,x_1\}$ и $\{y_0,y_1\}$. На это уйдет 3 умножения, и будут получены значения x_1y_1 и $x_0y_1+x_1y_0$. При этом полученное при преобразовании значение x_0+y_1 может быть использовано в дальнейшем, сокращая количество сложений. Далее выполняется свертка последовательностей $\{x_0,x_2\}$ и $\{y_0,y_2\}$. На это уже необходимо только 2 умножения, и будут получены значения x_2y_2 и $x_0y_2+x_2y_0$, которые используются в дальнейшем.

Тогда вычисления отсчетов свертки с указанием количества умножений сведутся к следующим вычислениям:

$$\begin{array}{ll}
x_0y_0, x_1y_1, x_0y_1+x_1y_0 & -\ 3\\
x_2y_2, x_1y_1, x_0y_2+x_2y_0 & -\ 2\\
z_{-3} = x_0y_0 & -\\
z_{-2} = x_0y_1+x_1y_0 & -\\
z_{-1} = x_0y_2+x_2y_0+x_1y_1 & -\\
z_0 = \alpha_0 & -\ 4\\
z_1 = \alpha_1 - (x_0y_2+x_2y_0)\ +\ x_2y_2 & -\\
z_2 =\ \alpha_2 - x_0y_1 + x_1y_0 & -\\
z_3 =\ \alpha_3 - (x_0y_0+x_1y_1+x_2y_2) & -
\end{array} \qquad (21)$$

Как можно рассчитать, всего необходимо выполнить 9 умножений и 30 сложений (здесь учтены также 5 умножений и 10 сложений для сверток 2×2, а также 14 сложений, необходимых для перехода из $W^{(2)}(e,4)$ в систему $W_1^{(2)}(f,4)$ и обратно).

Двухмерная линейная свертка

Компоненты двухмерной линейной свертки рассчитываются по формуле [9]:

$$z(n_1,n_2) = \sum_{l_1=-\infty}^{\infty} \sum_{l_2=-\infty}^{\infty} x(k_1,k_2)h(n_1-k_1,n_2-k_2)\,. \qquad (22)$$

Здесь и сигнал $x(k_1,k_2)$, и ядро $h(n_1,n_2)$ являются двухмерными массивами. Реальные дискретные сигналы имеют конечные размеры.

Существует большое количество методов быстрого вычисления линейной двухмерной свертки. С некоторыми из них можно познакомиться в работах [1–5, 9] и многих других. В данной работе будут рассмотрены алгоритмы выполнения линейной двухмерной свертки, основанные на переходе к гиперкомплексным пространствам, принцип которого описан в предыдущем разделе.

Структура алгоритма быстрой линейной двухмерной свертки массивов 2×2

Сигнал x и ядро y двухмерной свертки представляются 2-мерными матрицами 2×2. Двухмерная свертка массивов 2×2 вычисляется по формуле:

$$con_{kl} = \sum_{n=1}^{2}\sum_{m=1}^{2} x_{nm} y_{n-k,m-l}\,,\ k,l = -1,0,1\,. \qquad (23)$$

Пусть сигнал и ядро свертки имеют вид:

x_{11}	x_{12}
x_{21}	x_{22}

y_{11}	y_{12}
y_{21}	y_{22}

Тогда для корректного вычисления по формуле (23) матрицу сигнала нужно окаймить со всех сторон строками и столбцами, заполненными нулями:

$y_{00}=0$	$y_{01}=0$	$y_{02}=0$	$y_{03}=0$
$y_{10}=0$	y_{11}	y_{12}	$y_{13}=0$
$y_{20}=0$	y_{21}	y_{22}	$y_{23}=0$
$y_{30}=0$	$y_{31}=0$	$y_{32}=0$	$y_{33}=0$

Таким образом, компоненты свертки будут иметь следующие значения:

$$\begin{aligned}
&z_{-1,-1} := x_{21}y_{12},\\
&z_{-1,0} := x_{11}y_{12} + x_{21}y_{22},\\
&z_{-1,1} := x_{11}y_{22},\\
&z_{0,-1} := x_{21}y_{11} + x_{22}y_{12},\\
&z_{0,0} := x_{11}y_{11} + x_{12}y_{12} + x_{21}y_{21} + x_{22}y_{22},\\
&z_{0,1} := x_{11}y_{21} + x_{12}y_{22},\\
&z_{1,-1} := x_{22}y_{11},\\
&z_{1,0} := x_{12}y_{11} + x_{22}y_{21},\\
&z_{1,1} := x_{12}y_{21}.
\end{aligned} \qquad (24)$$

Ниже представлен алгоритм быстрой линейной двухмерной свертки массивов 2×2, оформленный в виде процедуры *Conv22*, написанной в системе символьных вычислений Maple. Процедура *Conv22* входит в пакет гиперкомплексных вычислений, разработанный в отделе специализированных средств моделирования ИПРИ НАН Украины и описанный в работах [10–12]. В самой процедуре *Conv22* используются и другие процедуры этого пакета.

Процедура *Conv22* имеет следующие формальные параметры:

1) *Aj* — ядро двухмерной свертки в виде списка;

2) *Ajm* — модифицированное с помощью процедуры *Modi2* ядро двухмерной линейной свертки в виде списка;

3) *B* — двухмерный сигнал 2×2 в виде матрицы;

4) *W*4 — таблица Кэли четверной гиперкомплексной числовой системы $W^{(2)}(e,4)$ в виде списка;

5) *W*41 — таблица Кэли орточетверной гиперкомплексной числовой системы $W_1^{(2)}(f,4)$ в виде списка.

На выходе процедуры компоненты свертки в виде списка длиной 9 элементов.

Листинг процедуры *Conv22* приведен ниже:

```
Conv22 := proc(Aj, Ajm, B, W4, W41)
    local Y1, i, j, Ys, C1, con, C2, B1, Bs;
    B1 := MatrList(B);
    Bs := W4W41(B1);
    C1 := inMulti(Bs[1], Ajm, W41);
    con := array( − 1 ..1, − 1 ..1);
    C2 := W4W41(C1)[1];
    for i to 4 do C2[i] := simplify(C2[i]) end do;
    con[ − 1, − 1] := Aj[3]*B[1, 2];
    con[ − 1, 0] := Aj[1]*Y[1, 2] + Aj[3]*Y[2, 2];
    con[ − 1, 1] := Aj[1]*Y[2, 2];
    con[0, − 1] := Aj[3]*Y[1, 1] + Aj[4]*Y[1, 2];
    con[0, 0] := C2[1];
    con[0, 1] := C2[3] − con[0, − 1];
    con[1, − 1] := Aj[4]*Y[1, 1];
    con[1, 0] := C2[2] − con[ − 1, 0];
    con[1, 1] := C2[4] − con[ − 1, − 1] − con[1, − 1] − con[
     − 1, 1];
    RETURN(con)
end proc
```

В начале процедуры сигнал *B* с помощью процедуры *MatrList* преобразуется из матричной формы в списковую. При этом подразумевается, что это гиперкомплексное четверное число. Далее это число с помощью процедуры *W*4*W*41 преобразуется в орточетверное гиперкомплексное число. В процедуре *inMulti* оно умножается на модифицированное ядро *Ajm* в орточетверной ГЧС *W*41, для чего требуется 4 вещественных умножения.

Модифицированное ядро — это также орточетверное гиперкомплексное число в виде списка, все компоненты которого умножены на коэффициент модификации, равный в данном случае 1/4. Эта операция выполняется вне процедуры *Conv22*, что снижает количество вычислений в ней. Далее произведение переводится с помощью процедуры $W4W41$ в четверную ГЧС $W4$. Здесь используется именно процедура $W4W41$, а не $W41W4$, потому что они отличаются только коэффициентом модификации, который уже учтен в модифицированном ядре свертки.

После этого вычисляются компоненты свертки. Первые четыре компонента вычисляются непосредственно по компонентам сигнала и немодифицированного ядра. Для этого требуется 6 умножений. Компонент свертки $con(0,0)$ совпадает с первым компонентом произведения сигнала и ядра. Остальные компоненты вычисляются с помощью 2-, 3- и 4-го компонентов произведения сигнала и ядра в четверной форме и уже вычисленных компонентов свертки. Здесь операция умножения не используется.

Таким образом, как было отмечено выше, всего требуется выполнить 9 вещественных умножений вместо 16, если выполнять свертку непосредственно по формуле (23).

Вычисление этой же свертки можно построить путем сведения двухмерной свертки к 4 построчным одномерным сверткам массивов сигнала и ядра. При этом необходимо выполнить $4 \cdot 3 = 12$ вещественных операций, что больше, чем в предыдущем варианте.

Структура алгоритма быстрой линейной двухмерной свертки массивов 4×4

Структура этого алгоритма подобна структуре алгоритма свертки 2×2, однако есть существенные различия, обусловленные большей размерностью массивов сигнала и ядра, на что будет обращено внимание далее.

Сигнал x и ядро y двухмерной свертки представляются 2-мерными матрицами 4×4.

Двухмерная свертка массивов 4×4 вычисляется по формуле:

$$con_{kl} = \sum_{n=1}^{4} \sum_{m=1}^{4} x_{nm} y_{n-k,m-l}, \; k,l = -3,-2,..,0,..,3. \tag{25}$$

Пусть сигнал и ядро свертки имеют вид:

x_{11}	x_{12}	x_{13}	x_{14}
x_{21}	x_{22}	x_{23}	x_{24}
x_{31}	x_{32}	x_{33}	x_{34}
x_{41}	x_{42}	x_{43}	x_{44}

y_{11}	y_{12}	y_{13}	y_{14}
y_{21}	y_{22}	y_{23}	y_{24}
y_{31}	y_{32}	y_{33}	y_{34}
y_{41}	y_{42}	y_{43}	y_{44}

Как и в предыдущем разделе для корректного вычисления по формуле (25) матрицу сигнала нужно окаймить со всех сторон строками и столбцами, заполненными нулями, как показано ниже:

0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	y_{11}	y_{12}	y_{13}	y_{14}	0	0	0
0	0	0	y_{21}	y_{22}	y_{23}	y_{24}	0	0	0
0	0	0	y_{31}	y_{32}	y_{33}	y_{34}	0	0	0
0	0	0	y_{41}	y_{42}	y_{43}	y_{44}	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0

При этом получится матрица размером 10×10, а нумерация строк и столбцов будет от –2 до 7. Компоненты свертки будут иметь индексы от [–3, –3] до [3, 3], всего 49 компонентов. Они будут содержать 1, 2, 3, 4, 6, 8, 9, 12 или 16 парных произведений компонентов сигнала и ядра свертки. Вид этих компонентов полностью здесь приводиться не будет ввиду их громоздкости и большого количества — 49. Приведем отдельные примеры:

$$z_{-3,\,-3} := y_{1,\,4}\,x_{4,\,1}$$

$$\begin{aligned} z_{0,\,0} := {} & y_{1,\,1}\,x_{1,\,1} + y_{1,\,2}\,x_{1,\,2} + y_{1,\,3}\,x_{1,\,3} + y_{1,\,4}\,x_{1,\,4} + y_{2,\,1}\,x_{2,\,1} \\ & + y_{2,\,2}\,x_{2,\,2} + y_{2,\,3}\,x_{2,\,3} + y_{2,\,4}\,x_{2,\,4} + y_{3,\,1}\,x_{3,\,1} + y_{3,\,2}\,x_{3,\,2} \\ & + y_{3,\,3}\,x_{3,\,3} + y_{3,\,4}\,x_{3,\,4} + y_{4,\,1}\,x_{4,\,1} + y_{4,\,2}\,x_{4,\,2} + y_{4,\,3}\,x_{4,\,3} \\ & + y_{4,\,4}\,x_{4,\,4} \end{aligned}$$

$$\begin{aligned} z_{2,\,0} := {} & y_{1,\,3}\,x_{1,\,1} + y_{1,\,4}\,x_{1,\,2} + y_{2,\,3}\,x_{2,\,1} + y_{2,\,4}\,x_{2,\,2} + y_{3,\,3}\,x_{3,\,1} \\ & + y_{3,\,4}\,x_{3,\,2} + y_{4,\,3}\,x_{4,\,1} + y_{4,\,4}\,x_{4,\,2} \end{aligned}$$

и т.д.

Для построения алгоритма необходимо синтезировать две изоморфных ГЧС 16-й размерности — сильно- и слабозаполненные.

Первая, сильнозаполненная, получается путем автоумножения ГЧС W размерности 4 применением процедуры *MultiDim*(W,W,e,1,W16) из пакета гиперкомплексных вычислений. Назовем ее W16. Так как таблица умножения ГЧС W16 очень громоздкая, то приведем ее схему:

$$W16 = \begin{array}{|c|c|} \hline G_1 & G_2 \\ \hline G_2 & G_1 \\ \hline \end{array} \quad ,$$

где:

$$G_1 = \begin{matrix} e_1 & e_2 & e_3 & e_4 & e_5 & e_6 & e_7 & e_8 \\ e_2 & e_1 & e_4 & e_3 & e_6 & e_5 & e_8 & e_7 \\ e_3 & e_4 & e_1 & e_2 & e_7 & e_8 & e_5 & e_6 \\ e_4 & e_3 & e_2 & e_1 & e_8 & e_7 & e_6 & e_5 \\ e_5 & e_6 & e_7 & e_8 & e_1 & e_2 & e_3 & e_4 \\ e_6 & e_5 & e_8 & e_7 & e_2 & e_1 & e_4 & e_3 \\ e_7 & e_8 & e_5 & e_6 & e_3 & e_4 & e_1 & e_2 \\ e_8 & e_7 & e_6 & e_5 & e_4 & e_3 & e_2 & e_1 \end{matrix}, \quad G_2 = \begin{matrix} e_9 & e_{10} & e_{11} & e_{12} & e_{13} & e_{14} & e_{15} & e_{16} \\ e_{10} & e_9 & e_{12} & e_{11} & e_{14} & e_{13} & e_{16} & e_{15} \\ e_{11} & e_{12} & e_9 & e_{10} & e_{15} & e_{16} & e_{13} & e_{14} \\ e_{12} & e_{11} & e_{10} & e_9 & e_{16} & e_{15} & e_{14} & e_{13} \\ e_{13} & e_{14} & e_{15} & e_{16} & e_9 & e_{10} & e_{11} & e_{12} \\ e_{14} & e_{13} & e_{16} & e_{15} & e_{10} & e_9 & e_{12} & e_{11} \\ e_{15} & e_{16} & e_{13} & e_{14} & e_{11} & e_{12} & e_9 & e_{10} \\ e_{16} & e_{15} & e_{14} & e_{13} & e_{12} & e_{11} & e_{10} & e_9 \end{matrix}.$$

Вторая ГЧС, слабозаполненная, получается путем автоумножения ГЧС $W1$ размерности 4 применением процедуры $MultiDim(W1,W1,e,1,W16o)$ из пакета гиперкомплексных вычислений. Назовем ее $W16o$. Таблица умножения ГЧС $W16o$ тривиальна:

$$f_i \cdot f_j = \begin{cases} 0 : i \neq j \\ f_i : i = j \end{cases}, \text{ где } f_i \text{ — базисные элементы ГЧС } W16o.$$

Так как ГЧС W и $W1$ изоморфны, то, на основании теоремы 2 из [8] ГЧС $W16$ и $W16o$ также изоморфны. Теорема 3 из [8] определяет метод построения оператора изоморфизма между $W16$ и $W16o$, то есть зависимости между базисными элементами обоих систем. Из оператора изоморфизма видно, что коэффициент модификации ядра свертки в данном случае равен 1/16.

Матрицы числа и ядра можно преобразовать в гиперкомплексные числа 16-й размерности. Предполагаем, что они принадлежат ГЧС $W16$. При этом компонент $x_{i,j}$ матрицы преобразуется в компонент гиперкомплексного числа $x_{4(i-1)+j}$.

С помощью оператора изоморфизма эти числа можно перевести в ГЧС $W16o$. При этом переходе используются только операции сложения. Произведение этих чисел в общем виде в системе $W16$ дает 16-мерное гиперкомплексное число, каждый компонент которого есть сумма 16 парных произведений компонентов ядра и сигнала. Например, компонент при e_2 имеет вид:

$$x_1y_2 + x_2y_1 + x_3y_4 + x_4y_3 + x_5y_6 + x_6y_5 + x_7y_8 + x_8y_7 +$$
$$+x_9y_{10} + x_{10}y_9 + x_{11}y_{12} + x_{12}y_{11} + x_{13}y_{14} + x_{14}y_{13} + x_{15}y_{16} + x_{16}y_{15}.$$

Анализ компонентов гиперкомплексного произведения ядра свертки и сигнала показывает, что в нем есть фрагменты компонентов свертки: 1 компонент длиной 16 парных произведений, 8 по 8, 24 по 4, 32 по 2 и 16 одиночных парных произведений.

С другой стороны, произведение сигнала и ядра в ГЧС $W16o$ переведенное в ГЧС $W16$, дает 16 значений компонентов произведения сигнала и ядра в ГЧС $W16$. Эта информация позволяет построить схему вычисления линейной двухмерной свертки массивов 4×4, число умножений в которой равно 175 умножений, что на 81 операцию меньше, чем при прямом вычислении, требующем 256 вещественных умножений.

Линейную двухмерную свертку можно выполнить путем сведения двухмерной свертки к последовательности одномерных сверток путем сворачивания каждой строки матрицы сигнала с каждой строкой матрицы ядра. При этом необходимо выполнить 16 одномерных сверток 4×4, то есть найти те же 256 парных произведений. Выполнение линейной свертки 4×4 по алгоритму (21) требует 9 умножений. Так что выполнение двухмерной свертки 4×4 по построчной схеме требует $16 \cdot 9 = 144$ умножения, что гораздо меньше предыдущего алгоритма.

Выводы

Проведенные исследования показали, что использование методов гиперкомплексных числовых систем позволяет значительно уменьшать количество вещественных операций в алгоритмах выполнения двухмерной линейной свертки. При этом выбор рациональной структуры алгоритма зависит от размерности сворачиваемых массивов.

1. Блейхут Р. Быстрые алгоритмы цифровой обработки сигналов. Москва: Мир, 1989. 449 с.

2. Нуссбаумер Г. Быстрое преобразование Фурье и алгоритмы вычисления сверток. Москва: Радио и связь, 1985. 248 с.

3. Калиновский Я.А., Ландэ Д.В., Бояринова Ю.Е., Хицко Я.В. Гиперкомплексные числовые системы и быстрые алгоритмы цифровой обработки информации. Киев: ИПРИ НАНУ, 2014. 130 с.

4. Калиновский Я.А. Структура гиперкомплексного метода быстрого вычисления линейной свертки дискретных сигналов. *Реєстрація, зберігання і оброб. даних*. 2013. Т. 15. № 1. С. 31–44.

5. Сергиенко А.Б. Цифровая обработка сигналов. Санкт-Петербург: Питер, 2003. 604 с.

6. Гольденберг Л.М., Матюшкин Б.Д., Поляк М.Н. Цифровая обработка сигналов. Москва: Радио и связь, 1985. 312 с.

7. Синьков М.В., Бояринова Ю.Е., Калиновский Я.А. Конечномерные гиперкомплексные числовые системы. Основы теории. Применения. Киев: Инфодрук, 2010. 388 с.

8. Калиновский Я.А., Бояринова Ю.Е. Высокоразмерные изоморфные гиперкомплексные числовые системы и их использование для повышения эффективности вычислений. Киев: Инфодрук, 2012. 183 с.

9. Даджион Д., Мерсеро Р. Цифровая обработка многомерных сигналов. Москва: Мир, 1988. 488 с.

10. Калиновский Я.А., Бояринова Ю.Е., Хицко Я.В., Сукало А.С. Программный комплекс для гиперкомплексных вычислений. *Электронное моделирование*. 2017. Т. 39. № 5. С. 81–96.

11. Калиновский Я.А., Бояринова Ю.Е., Хицко Я.В., Сукало А.С. Система гиперкомплексных операций в Maple. *Реєстрація, зберігання і оброб. даних*. 2017. Т. 19. № 2. С. 11–23.

12. Kalinovsky Y.A., Boyarinova Y.E., Sukalo A.S., Khitsko Y.V. The basic principles and the structure and algorithmically software of computing by hypercomplex number. arXiv preprint arXiv:1708.04021, 2017.

Поступила в редакцию 19.02.2018