

МЕТОДЫ И СИСТЕМЫ АВТОМАТИЗАЦИИ.
ОБРАБОТКА ДАННЫХ ДИСТАНЦИОННОГО ЗОНДИРОВАНИЯ

УДК 681.1

С.А. Иконников

**СИНТЕЗ RNS-ПРОЦЕССОРА НЕЙРОКОМПЬЮТЕРНОЙ СЕТИ
ДЛЯ ОБРАБОТКИ ВИДОВОЙ АЭРОКОСМИЧЕСКОЙ ИНФОРМАЦИИ
В ЗАДАЧАХ ЭКОЛОГИЧЕСКОГО МОНИТОРИНГА**

Рассмотрены общие принципы функционирования нейронной сети. Описан способ определения вектора поправок матрицы весовых коэффициентов на основе линейного алгоритма восстановления. Предложен алгоритм синтеза оптимальной структуры RNS- процессора.

Введение

Весьма перспективным направлением исследований, способным привести к новым существенным изменениям в принципах создания вычислительных систем, используемых в обработке аэрокосмической видовой информации, представляются концепции использования нейронных сетей и нетрадиционной арифметики, например на основе системы счисления в остаточных классах (RNS – residue-number-system).

При решении задач экологического мониторинга в различных источниках нейронные сети рассматривают как одно из направлений развития искусственного интеллекта, альтернативу цифровым компьютерам и алгоритмическому программированию. Эти определения нельзя считать абсолютно точными, однако предварительные исследования позволяют судить о том, что нейронные сети и RNS-арифметика могут кардинально изменить ситуацию, где цифровая обработка и алгоритмическое программирование не позволяют добиться серьезных успехов. Согласно [11] некоторые университетские лаборатории (Иллинойс, Калифорния) и солидные корпорации (INTEL, J.ELECTRIC) ведут активные научно-исследовательские работы с целью внедрения методов и средств нейросетей для таких серьезных прикладных задач, как распознавание образов и чтение символов.

Такой подход позволит в перспективе существенно повысить функциональные возможности проектирования компьютеров, которые должны будут включать различные методы обработки информации.

1. Общие принципы функционирования нейрокомпьютерной сети

Поскольку многие из решаемых нейрокомпьютером задач связаны с обработкой первичной содержательной информации, такой как изображение, в качестве архитектурных принципов построения нейросети выбирают простые модели биологических систем.

Нейронная сеть – адаптивная сеть, состоящая из входов, выходов и процессорных элементов (ПЭ) и способная минимизировать функцию стоимости для получения необходимого выходного результата.

На входах и выходах может быть практически любой тип кодируемой информации в зависимости от конкретной прикладной задачи.

ПЭ производит «взвешивание» входного сигнала (r_i), т.е. определение его синоптического веса ($r_i W_i$), позволяющее сети адекватно и точно отображать входные данные на выходе, которые затем суммируются и проходят через НЕЙРОН (рис. 1, а). Нейрон выполняет функцию $F(r_i W_i)$, которая может быть линейной, шаговой или нелинейной сигмоидальной и выбираться таким образом, чтобы достигались быстрая сходимость и конечный результат R (рис. 1, б).

Практически реализованные сети имеют две ключевые архитектурные характеристики: каждый вход соединяется с каждым ПЭ, а в сети более одного уровня выходы одного уровня становятся входами следующего уровня (рис. 2).

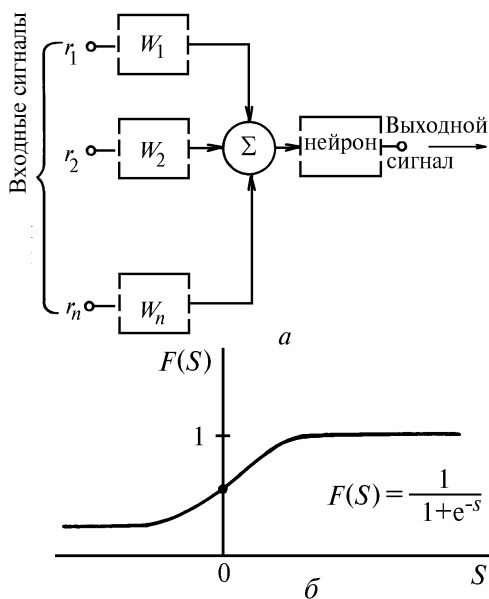


Рис. 1. Структура процессорного элемента (а) и сигмоидальная функция нейрона (б)

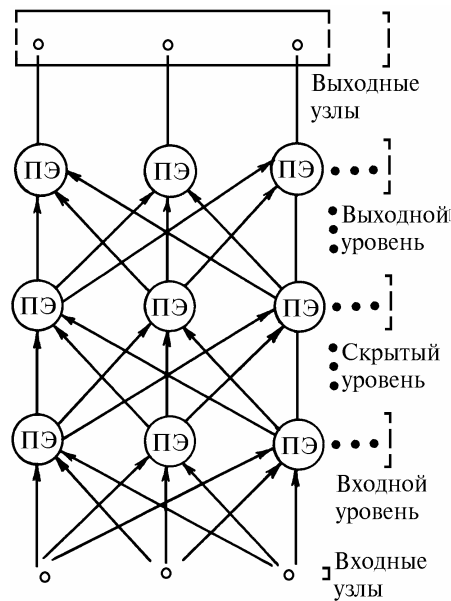


Рис. 2. Многоуровневая структура нейросети

Для нахождения правильного веса используется процесс обучения нейросети – способ проб и ошибок. Успешное обучение возможно, если данные представляются в правильном формате и охватывают требуемые числовые диапазоны, т.е. данные обучения должны включать весь диапазон исходных данных для сети. Алгоритм статистического анализа, реализованный в нейросети, будет аппроксимировать кривую данных с интерполяцией в области, не совпадающей со значениями данных, которые использовались при обучении в сети. Иначе говоря, при обучении данные «прокачиваются» через сеть, чтобы зафиксировать правильные значения синоптических весов. После обучения любые входные данные можно будет отобразить на правильное выходное значение.

С точки зрения математического моделирования нейронная сеть – это динамическая система, которую можно моделировать как некоторое множество связанных дифференциальных уравнений. Подобная система потенциально неустойчива, так как небольшие колебания весовых коэффициентов могут приводить к появлению неуправляемых возбуждений, выбросов, стоячих и бегущих волн, а в пределе – к хаотическому состоянию.

Если рассматривать сеть с ее изменениями состояний как некую энергетическую поверхность, то устойчивые состояния будут соответствовать энергетическим минимумам. Каждый хранимый образ (pattern) сети, например блок данных, соответствует некоему устойчивому состоянию, или локальному минимуму. Этот образ ассоциируется с неким входным образом, так что поступление последнего в нейросеть приведет к тому, что сеть выберет соответствующий хранимый образ – подобно ассоциативной памяти.

Однако сложность в том, что в различных вариантах модели одному и тому же входному образу могут соответствовать несколько энергетических минимумов, а это означает, что не всегда можно получить правильный выходной результат при обработке входных данных.

Чтобы снизить вероятность ошибки при работе нейрокомпьютера, для каждого класса решаемых задач разрабатывается или используется специализированный алгоритм обучения нейросети. В настоящее время наибольшее распространение получил алгоритм обучения нейросети ВРЕ (Back Propagation of Error) [10] – обратное распространение ошибки, точность 94,1%. Обучение сети по этому алгоритму начинается с присвоения синоптическим весам небольших случайных значений. Для первого множества входных векторов получаются неправильные выходные векторы. Однако поскольку правильные значения выходных векторов известны, можно вычислить разностные значения, которые называются корректирующими коэффициентами или дельта-ошибками (Q).

Полученные Q -значения для каждого синоптического веса W_i распространяются по сети в обратном направлении.

По мере их прохождения по уровням нейросети происходит изменение значений, соответствующих каждому синоптическому весу W . В конечном итоге матрица синоптических весов

$$W_{ij} = \begin{pmatrix} W_{11}, & W_{12}, & \dots, & W_{1n} \\ W_{21}, & W_{22}, & \dots, & W_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ W_{m1}, & W_{m2}, & \dots, & W_{mn} \end{pmatrix}, \quad (1.1)$$

где i – n -й синоптический вес j -го уровня, j – m -й уровень нейросети, настраивается таким образом, что сеть оказывается способной вычислить правильный выходной вектор для первого входного вектора. Эти значения весовых коэффициентов используются при обработке второго множества входных векторов.

Процесс повторяется, и матрица синоптических весов W_{ij} модифицируется снова и снова, пока не будет получена оптимальная конфигурация для всего поля известных данных.

Для более точных вычислений матрицы W_{ij} в зависимости от класса решаемых задач прибегают к более сложным алгоритмам обучения нейросетей [10]. В числе известных:

LVQ – (Learning Vector Quantization) – линейная квантизация векторов, 95,7 %;

STLVQ – (Shift-Tolerant Learning Vector Quantization) – линейная квантизация векторов с толерантностью к сдвигам, 98 %;

RBF – (Radial Basic Functions) – радиальные базисные функции, 99%;

RCE – (Restricted Coulomb Energy) – ограниченная энергия в кулонах, 99 %;

BDT – (Bias Data Training) – обучение на данных с коррекцией, 99%.

Две базовые концепции построения нейросети, а именно установление множества соединений и выполнение множества итераций для достижения адекватного отображения входных параметров на выходные, обуславливают большую интенсивность вычислений, необходимую для решения конкретных прикладных задач. Вследствие этого стараются по возможности уменьшить размеры нейросети после определения ее начальной конфигурации. Однако выгоды, достигаемые благодаря настройке и сокращению сети, бывают обычно весьма ограниченными. Гораздо чаще в связи с этим требуются аппаратные средства ускорения вычислений.

2. Определение вектора поправок Q на основе линейного алгоритма восстановления

Жесткие ограничения на допустимое время обучения нейросети и обработки информации, большие объемы при ее передаче и относительно малые значения шумового коэффициента (Q -ошибки) показывают целесообразность применения линейных алгоритмов восстановления. Центральной и наиболее трудоемкой операцией линейных алгоритмов является свертка наблюдаемых отсчетов с некоторой весовой функцией корректирующего элемента [2].

Для синтеза быстрых алгоритмов свертки важное значение имеет учет свойств алгебраических структур, над которыми они вычисляются. В данном случае предлагается рассмотреть вычисление свертки над кольцами классов вычетов Z_m . Эти алгебраические структуры представляются наиболее оптимальными для обучения и функционирования нейросети.

Линейная свертка L -точечной последовательности $\{r_i\}$ с M -точечной последовательностью $\{W_i\}$ дает растянутую $K = L + M + 1$ -точечную последовательность $\{S_i\}$.

В компактном виде одномерная линейная свертка $S = r \otimes W$ двух целочисленных последовательностей $\{r_i\}$ и $\{W_i\}$ выглядит как

$$S(x) = r(x) W(x), \quad (2.1)$$

где степени полиномов $\deg r(x) = L - 1$, $\deg W(x) = M - 1$, $\deg S(x) = L + M - 2$.

В работе [2] показано, что коэффициенты S полинома $S(x)$ находятся из выражения

$$\sum_{i=0}^{N-1} S_i x^i = \sum_{i=0}^{L-1} r_i x^i \sum_{m=0}^{M-1} W_m x^m, \quad (2.2)$$

откуда

$$S_i = \sum_{n=0}^{N-1} r_n W_{i-n}, \quad (2.3)$$

где r_i – квантованные по уровню дискретные отсчеты первой целочисленной последовательностью, например, отсчеты входного сигнала; W_m – дискреты второй последовательности, например передаточной характеристикой синоптического веса. $i, n = 0, 1, \dots, N$; $l = 0, 1, \dots, L - 1$; $m = 0, 1, \dots, M - 1$.

Для того чтобы коэффициенты W_{i-n} всегда имели смысл, необходимо периодически продолжать их значения с периодом N . Это означает, что при значении индекса $(i-n)$, не лежащем в интервале $0 < i-n < N-1$, необходимо добавить или вычесть некоторое число, кратное N , чтобы выполнялось неравенство $0 < i-n + mN < N-1$.

Тогда для вычисления свертки можно использовать значения

$$W_{i-n \pm mN} = W_{(i-n) \pmod{N}}.$$

Полученная таким образом свертка $S = r \otimes h$ называется **циклической сверткой** и записывается в полиномиальном виде как вычет линейной свертки по модулю полинома $X - 1$:

$$S(x) = r(x) W(x) \pmod{x - 1} \quad (2.4)$$

или

$$\sum_{i=0}^N S_i x^i = \sum_{i=0}^N r_i x^i W_i x^i \pmod{x - 1}, \quad (2.5)$$

откуда

$$S_i = \sum_{n=0}^N r_n W_{(i-n) \pmod{N}}. \quad (2.6)$$

Для различных значений S_i при $n = 0, 1, \dots, N - 1$ выражению $W_{(i-n) \pmod{N}}$ будут соответствовать последовательности, отличающиеся одна от другой лишь циклическим сдвигом. Следовательно, вычисление разности $(i-n) \pmod{N}$ приводит к периодизации (циклическости) перестановок элементов последовательности отсчетов W_i . Эта цикличность наиболее отчетливо видна при матричной форме записи свертки:

$$\begin{bmatrix} S_0 \\ S_1 \\ \cdot \\ \cdot \\ \cdot \\ S_{N-1} \end{bmatrix} = \begin{bmatrix} W_0 & W_{N-1} & \dots & W_1 \\ W_1 & W_0 & \dots & W_2 \\ \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot \\ W_{N-1} & W_{N-2} & \dots & W_0 \end{bmatrix} \times \begin{bmatrix} r_0 \\ r_1 \\ \cdot \\ \cdot \\ \cdot \\ r_{N-1} \end{bmatrix}, \quad (2.7)$$

где $\begin{bmatrix} S_0 \\ S_1 \\ \cdot \\ \cdot \\ \cdot \\ S_{N-1} \end{bmatrix} = S$ – вектор-столбец результатов вычисления свертки;

$$\begin{bmatrix} W_0 & \dots & W_1 \\ \cdot & \cdot & \\ \cdot & \cdot & \\ \cdot & \cdot & \\ W_{N-1} & \dots & W_0 \end{bmatrix} = W - \text{матрица отсчетов передаточной характеристики синоптических весов};$$

$[r_0 \ r_2 \ \dots \ r_{N-1}]^T = R$ – вектор-столбец отсчетов текущей последовательности данных.

Если циклическая свертка используется для вычисления линейной свертки, то влияние периодичности циклической свертки следует компенсировать. Для вычисления всей последовательности $\{S_i\}$ с помощью циклической свертки длина последней должна быть равной N точкам. Кроме того, последовательности $\{r_i\}$ и $\{W_i\}$ должны быть продолжены до N -точечных последовательностей. Для этого достаточно к $\{r_i\}$ и $\{W_i\}$ добавить соответствующее число нулей и построить последовательности

$$r_i = \begin{cases} r_i, & 0 \leq i \leq L-1, \\ 0, & L \leq i \leq N-1; \end{cases} \quad (2.8)$$

$$W_i = \begin{cases} W_i, & 0 \leq i \leq M-1, \\ 0, & M \leq i \leq N-1. \end{cases} \quad (2.9)$$

Тогда на интервале $0 < i < N-1$ значения циклической S'_i и линейной свертки S_i совпадают, т.е. в полиномиальном виде

$$S'(x) = r'(x) W'(x) \pmod{x-1} = r(x) W(x) = S(x). \quad (2.10)$$

Для вычисления коэффициентов свертки над кольцом классов вычетов Z_m для линейной свертки используют выражение

$$S_i = \left(\sum_{n=0}^{N-1} r_n W_{i-n} \right) \pmod{m}, \quad (2.11)$$

а циклической свертки соответственно

$$S'_i = \left(\sum_{n=0}^{N-1} r_n W_{(i-n) \pmod{N}} \right) \pmod{m}. \quad (2.12)$$

Пока модуль m больше любого целого числа, используемого в вычислениях, операция по модулю m излишня, и все вычисления выполняются по обычным законам целочисленной арифметики.

В противном случае вступает в силу Z_m -арифметика, т.е. арифметика по модулю m . Полученные таким образом величины $S(S)$ позволяют на основе формул (1.1) и (2.7) достаточно просто определить вектор поправок $[Q_{ij}]$ к матрице весовых коэффициентов $[W_{ij}]$, что позволит построить достаточно эффективные специализированные численные алгоритмы обучения нейросети.

3. Алгоритм синтеза оптимального RNS-процессора

Для вычисления циклической свертки модуль m может быть выбран таким образом, чтобы в кольце Z_m имелись достаточно удобные процедуры для вычисления свертки.

В этих целях желательно использовать преимущества системы счисления в остаточных классах – С СОК (RNS – residue-number-system). Система RNS – это система счисления, в которой числа представляются в виде совокупности неотрицательных вычетов a_1, a_2, \dots, a_n по взаимно простым модулям (основаниям) m_1, m_2, \dots, m_n :

$$A = \{a_i \pmod{m} = 1, n\}. \quad (3.1)$$

Условие попарной взаимной простоты выбранных модулей $\{m_i\} = 1, i = \overline{1, n}$ обеспечивает однозначное представление числа A в диапазоне D , который равен произведению этих модулей:

$$D = \prod_{i=1}^n m_i. \quad (3.2)$$

В связи с тем что основания $\{m_i\}$ по определению независимы друг от друга, для выполнения арифметических операций $\{+, -, \times\}$ существует следующий изоморфизм прямой суммы:

$$x \cdot y = \sum \oplus (x \cdot y). \quad (3.3)$$

Данный изоморфизм определяет взаимнооднозначное соответствие между целым положительным числом $a < m$ и вектором вычетов (a_1, a_2, \dots, a_n) . Изоморфизм позволяет вести обработку квантованных значений видеоданных параллельно и независимо в каждом из колец Z_{m_j} по законам RNS-арифметики. В каждом кольце Z_{m_j} будет вычисляться вычет

$$S_{ij} = S_i \pmod{m_j}, \quad (3.4)$$

а целочисленный результат свертки, т.е. результат в кольце, определится по формуле

$$S_i = \left(\sum_{j=1}^n S_{ij} M_j N_j \right) \pmod{m}, \quad (3.5)$$

где $M_i = m_i/m_j, M_j N_j = 1 \pmod{m}$.

Таким образом, вычисление свертки можно осуществлять как над кольцом Z_m , так и над прямой суммой колец $Z_{m_1} + Z_{m_2} + \dots + Z_{m_n}$. На рис. 3 изображена схема вычисления свертки последовательностей $\{r_i\}$ и $\{W_i\}$ над кольцом Z_m и над прямой суммой $Z_{m_1} + \dots + Z_{m_n}$ колец вычетов попарно простым модулям m_1, m_2, \dots, m_n .

Тем самым предполагается, что арифметические операции могут выполняться над несколькими конечными кольцами, а результат прямой суммы отображаться в большее кольцо по модулю $\prod_{i=1}^n m_i$. Все вычисления не зависят от соседних, поэтому их можно выполнять параллельно, что и происходит в RNS-процессоре, т.е. реализуется принцип параллелизма до самой глубокой степени.

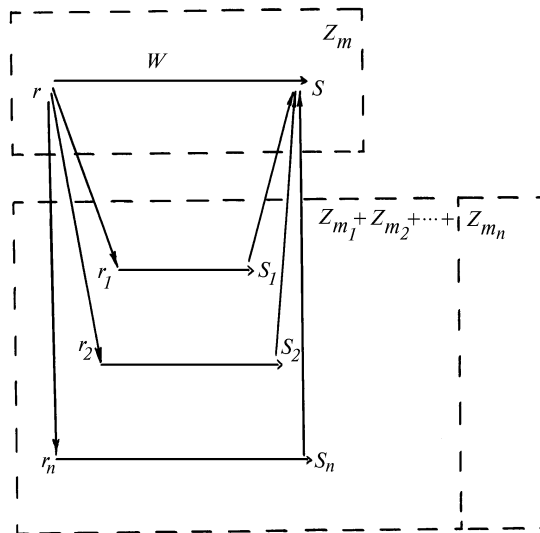


Рис. 3. Схема вычисления свертки $S = r \cdot W$ над прямой суммой $Z_{m_1} + Z_{m_2} + \dots + Z_{m_n}$

Структура RNS-процесса предельно проста – это регулярная матрица, где вычисления сводятся к простому выбору результата из j -го узла просматриваемой таблицы (матрицы).

Под j -м узлом понимается часть матрицы, реализующая отношение

$$C_j = \Phi(a; b),$$

где a и b принадлежат одному диапазону, т.е. $D \leq a^2$.

Очевидно, что в общем случае число узлов K матрицы для модуля m будет определяться $K_i = m_i^2 + m_i$, а для всего набора модулей $\{m_i\}$

$$K = \sum_{i=1}^n (m_i^2 + m_i). \quad (3.6)$$

При этом реализуется каждая возможная комбинация входных величин, а выходы узлов объединяются по правому Φ .

Основным недостатком табличного варианта реализации RNS-процессора является резкий рост объема оборудования (количества узлов K) при возрастании величин применяемых модулей $\{m_i\}$.

В этой связи возникает проблема оптимального подхода к выбору $\{m_i\}$ в смысле минимизации аппаратных затрат.

Проведенный автором анализ существующих критериев оценки наборов модулей $\{m_i\}$ в [3] позволил выявить возможные подходы к выбору наборов модулей с заранее заданными свойствами и разработать практическую методику выбора оптимального набора $\{m_i\}$ с точки зрения минимальных затрат.

Известно, что аппаратные затраты на построение RNS-процессора зависят от количества оснований и их значений. Кроме того, если члены K и D в (3.6) в некоторой степени коррелированы, следовательно, оптимальные аппаратные затраты $K_{\text{опт}}$ выражаются некоторой зависимостью

$$K_{\text{опт}} = f(n_{\text{max}}, V),$$

где V – относительный коэффициент перекрытия диапазона.

Если обратиться к формуле (3.2), то она выразится в виде

$$D = \left(\prod_{i=1}^n m_i \right) V, \quad (3.7)$$

следовательно, если один из модулей m первоначального набора $\{m_i\}$ разделить на V , то набор $\{m_i\}$ будет перекрывать диапазон D в $V \approx 1$.

В качестве уменьшаемого модуля выберем m_n , так как это основание в (3.2) имеет самый большой весовой коэффициент.

Ввиду того, что m_n может быть только целым числом, вновь полученное основание m'_n должно быть округлено до большего целого, т.е.

$$m'_n = \lceil m_n/V \rceil. \quad (3.8)$$

Кроме того, m'_n должно удовлетворять условию попарной простоты

$$(m_i, m_j) = 1, \text{ где } i, j \in Z. \quad (*)$$

Если m'_n не удовлетворяет условию (*), то m'_n необходимо увеличивать на 1 до тех пор, пока данное условие не выполнится. Таким образом, полученный набор $\{m_1, m_2, \dots, m'_n\}$ будет оптимальным для D в смысле аппаратных затрат.

Алгоритм выбора оптимальных наборов модулей $\{m_i\}$ RNS-процессора будет выглядеть следующим образом:

1. задается диапазон функционирования RNS-процессора D ;

2. Из ряда ППЧ $\{m_i\}$, начиная с 3, выбрать такие, которые выполняют условие

$$\prod_{i=1}^n m_i > D.$$

3. Определить коэффициент относительного перекрытия из условия

$$V = \left(\prod_{i=1}^n m_i \right) / D.$$

4. Самое большое основание m_n из набора $\{m_i\}$ разделить на V и округлить до целого, большего

$$m'_n = \lfloor m_n / V \rfloor.$$

5. Проверить m'_n на выполнение условия попарной простоты

$$(m_n, m'_n) = 1.$$

6. Условие $(m_n, m'_n) = 1$: а) выполняется – оптимальный набор найден; б) не удовлетворяется – увеличивать P_n на 1 до тех пор, пока условие не выполнится.

Предлагаемый алгоритм выбора оптимального набора модулей $\{m_i\}$ изложен с точки зрения синтеза стандартного RNS-процессора. Для количественной оценки степени оптимизации его аппаратных затрат при выборе оптимального $\{m_i\}$ в [4] математически определен коэффициент H , рассматриваемый как отношение количества узлов (процессорных элементов матрицы) синтезированного RNS-процессора K_c и условно идеального RNS-процессора K_n :

$$H = \frac{K_c}{K_n} = \frac{\sum_{i=1}^n (m_i^2 + m_i)}{n \cdot D^{1/n} (1 + D^{1/n})}. \quad (3.9)$$

Заключение

Проведенные численные эксперименты в [4] показывают, что предлагаемый алгоритм синтеза RNS-процессора позволяет проектировать обоснованно оптимальную нейронную сеть с точки зрения минимизации аппаратных затрат и проводить их количественную оценку.

Однако автору представляется, что оптимизация структуры RNS-процессора может быть углублена, если применить многоступенчатую систему счисления остаточных классов (RNS-арифметику).

1. Блейхут Р. Э. Быстрые алгоритмы цифровой обработки сигналов. М.: Мир, 1989. 448 с.
2. Бродович А. И., Шабakov Е. И. Целочисленная обработка видеосигналов и изображений. Выпуск 2. СПб.: ВИКА им. А.Ф. Можайского, 1993. 152 с.
3. Иконников С. А. Алгоритм выбора оптимального набора модулей для синтеза стандартного RNS-процессора // Материалы международной конференции «Информатика-94». СПб.: СПИИ РАН, 1994. Т. 3. С. 30–33.
4. Иконников С. А. Оценка степени оптимизации структуры RNS-процессора // Материалы научно-практической конференции. ВИКА им. А.Ф. Можайского, 1995. С. 220.
5. Лабунец В. Г. Алгебраическая теория сигналов и систем. Цифровая обработка сигналов. Красноярск: Изд-во Краснояр. ун-та, 1984. С. 244.
6. Макклелан Дж. Х., Рейдер Ч. М. Применение теории чисел в цифровой обработке сигналов. М.: Радио и связь, 1983. С. 8–59, 186–202.
7. Титков Б. В., Шабakov Е. И. Китайская теорема об остатках в цифровой обработке телевизионных сигналов // Техника средств связи. Сер. Техника телевидения. 1985. Вып. 4. С. 26–33.
8. Кнут Д. Искусство программирования для ЭВМ. Т. 2. Получисленные алгоритмы. М.: Мир, 1978. С. 74–76, 411–419, 476.
9. Vatrkov A. S., Brodovitch A. I., Shabakov E. I. Number-theoretic coding for iconic systems. SPIE. Vol. 1961. Visual Information Processing II (1993). P. 456–466.
10. Shandle Y. Neural networks are ready for prime time. 1 ED. 1993. N 4. P. 51–58.

11. T i l l Y . Computer system architecture. ED. 1989. N 1. P. 49–63.
12. V o l l u m C . The future of parallel processing software. ED. 1989. N 2. P. 38–51.

ВИКА им. А.Ф. Можайского,
С.-Петербург

Поступила в редакцию
30 мая 1995 г.

S . A . I k o n n i k o v . Synthesis of Neurocomputer Network RNS-processor for Aerospace Image Information Processing in Ecology Monitoring Problems.

General principles of a neuronetwork functioning are examined in the paper. A way for determination of the weight factors matrice correction vector is described based on reconstructing linear algorithm. An algorithm for RNS-processor optimal structure synthesis is proposed.