

УДК 681.518

Е.В. Бодянский, О.С. Удовенко

Харьковский национальный университет радиоэлектроники

## НЕПОЗИЦИОННОЕ КОДИРОВАНИЕ В ЗАДАЧАХ НЕЙРОСЕТЕВОГО ПРОГНОЗИРОВАНИЯ ВРЕМЕННЫХ РЯДОВ

Рассмотрен подход к реализации модульных вычислений для задач нейросетевого прогнозирования, позволяющий получить ряд существенных преимуществ, связанных с повышением надежности работы систем на основе применения параллельной структуры нейронных сетей. Показано, что непозиционная система остаточных классов обладает способностью самокоррекции, обнаружения и исправления ошибок. Кроме того, она позволяет сократить разрядность обрабатываемых данных, представляя их в модулярном коде в виде совокупности взаимно простых модулей. В связи с этим становится возможным упростить структуру нейронной сети, используемой для прогнозирования, и избежать проблем, связанных с масштабированием входных данных.

**Ключевые слова:** модулярный код, искусственная нейронная сеть, система остаточных классов, сигнаптическая матрица.

### Введение

Для решения задач прогнозирования и создания устойчивых необслуживаемых систем управления, диагностики, и контроля и широко используются искусственные нейронные сети (ИНС). При этом к наиболее важным критериям работы прогнозирующей системы на нейронных сетях следует отнести ее надежность и устойчивость к отказам элементов нейронной сети. Исследование отказоустойчивости ИНС прямого распространения показало, что параллельная организация является эффективным механизмом, обеспечивающим надежность нейронных структур, но запас устойчивости такой сети мал и при ее реализации могут возникнуть следующие проблемы:

- аппаратная реализация ИНС затруднена при росте количества скрытых слоев или нейронов в отдельном слое;
- отсутствует возможность оперативного контроля количества неисправных нейронов (при неисправности нейрона ошибка не проявляет себя на выходе);
- отсутствует возможность оперативной замены отказавших элементов сети;
- при использовании алгоритмов самообучения ИНС в условиях возникновения неисправностей увеличивается вероятность потери устойчивости в петле обучения;
- при обучении ИНС возникает необходимость переобучения;
- использование немасштабированных данных может привести к «параличу» сети.

Анализ возможных путей повышения эффективности выполнения арифметических операций показывает, что в рамках обычной позиционной системы значительного ускорения выполнения опе-

раций и увеличения надежности добиться почти невозможно [1]. Поэтому с развитием теории нейронных сетей, в частности, при решении задачи отображения модулярных вычислений на нейронные структуры, появилась возможность использовать преимущества модулярного кодирования для решения различных задач, в том числе и задач прогнозирования.

Система остаточных классов (СОК), лежащая в основе такого кодирования, обладает рядом существенных преимуществ, так как является непозиционной системой счисления, которая позволяет сократить разрядность обрабатываемых данных, представляя их в модулярном коде в системе взаимно простых модулей, и обладает способностью к обнаружению ошибок и самокоррекции.

### Постановка задачи

Основой системы остаточных классов является теория сравнений. СОК дает нестандартное представление чисел и используется для повышения эффективности операций над кодами в остатках [2].

Число  $A$  в модулярном коде представляется в виде набора  $(\alpha_1, \alpha_2, \dots, \alpha_k)$   $k$  остатков (вычетов), вычисленных по модулю каждого из оснований непозиционной системы счисления  $p_i$ :

$$\alpha_i = |A|_{p_i} = \left[ \begin{matrix} A \\ p_i \end{matrix} \right] \cdot p_i, \text{ для } i = 1, 2, \dots, k,$$

где  $[ ]$  – целая часть числа;  $k$  – количество оснований. В отличие от позиционной системы счисления (ПСС), в которой число  $A$  представляется в виде

$$A = A_n N^n + A_{n-1} N^{n-1} + \dots + A_1 N^1 + A_0 N^0 = \sum_{i=0}^n A_i N^i,$$

где  $N$  – основание ПСС, значение числа в модулярном коде не зависит от местоположения каждого

разряда в его представлении, а зависит от значения основания соответствующего разряда, поэтому модулярный код является непозиционным.

Если основания в представлении  $p_1, p_2, \dots, p_k$  выбрать попарно взаимно простыми, то, согласно Китайской теореме об остатках (КТО), это обеспечивает однозначность представления любых целых чисел из диапазона  $[0, P]$

$$P = \prod_{i=1}^k p_i.$$

СОК обеспечивает возможность выполнения операций сложения и умножения с высокой скоростью. Образование остатков от числа в СОК производится независимо друг от друга, вследствие чего каждый разряд содержит информацию о числе в целом. При выполнении сложения, вычитания и умножения каждая цифра результата зависит от соответствующих цифр операндов.

Так, если операнды  $A$  и  $B$  представлены соответственно остатками  $\alpha$  и  $\beta$  по основаниям  $p_i$  при  $i = 1, 2, \dots, k$ , то

$$\begin{aligned} A &= (\alpha_1, \alpha_2, \dots, \alpha_k); \quad B = (\beta_1, \beta_2, \dots, \beta_k), \\ |A+B|_p &= (|\alpha_1 + \beta_1|_{p_1}, |\alpha_2 + \beta_2|_{p_2}, \dots, |\alpha_n + \beta_n|_{p_k}), \\ |A-B|_p &= (|\alpha_1 - \beta_1|_{p_1}, |\alpha_2 - \beta_2|_{p_2}, \dots, |\alpha_n - \beta_n|_{p_k}), \\ |AB|_p &= (|\alpha_1 \beta_1|_{p_1}, |\alpha_2 \beta_2|_{p_2}, \dots, |\alpha_n \beta_n|_{p_k}). \end{aligned}$$

Таким образом, выполнение всех арифметических операций в модулярном коде производится независимо по каждому из модулей, что обеспечивает параллелизм данной системы. Это обстоятельство определяет возможность независимой обработки операндов, то есть поразрядного выполнения операций, что избавляет от необходимости переноса единиц из старшего разряда и приводит к образованию кодов с параллельной структурой, которые позволяют ускорить выполнение основных арифметических операций (сложение и умножение) в нейронной сети.

Кроме того, в СОК достаточно эффективно реализуются операции основного кольца. Пусть операнды  $A$  и  $B$  представлены соответственно остатками  $\alpha_i$  и  $\beta_i$  по основаниям  $p_i$  при  $i = 1, 2, \dots, n$ . Тогда результаты операции сложения и умножения  $A+B$  и  $AB$  представляются соответственно остатками  $\gamma_i$  и  $\sigma_i$  по тем же основаниям:

$$\begin{aligned} A &= (\alpha_1, \alpha_2, \dots, \alpha_n); \quad B = (\beta_1, \beta_2, \dots, \beta_n); \\ A+B &= (\gamma_1, \gamma_2, \dots, \gamma_n); \quad AB = (\sigma_1, \sigma_2, \dots, \sigma_n) \end{aligned}$$

и при этом имеют место следующие соотношения:

$$\begin{aligned} \gamma_i &= \alpha_i + \beta_i - \left[ \frac{\alpha_i + \beta_i}{p_i} \right] p_i; \\ \sigma_i &= \alpha_i \beta_i - \left[ \frac{\alpha_i \beta_i}{p_i} \right] p_i. \end{aligned}$$

Одной из проблем, возникающих при практической реализации СОК, является ограниченность ее применения только для целых положительных чисел. Эту проблему предлагается решить, используя алгоритм представления дробных чисел и работы с ними.

### Решение задачи

Рассмотрим пример реализации в СОК метода наименьших квадратов (МНК) для выделения тренда и определения коэффициентов модели регрессии по исходным данным, представленным в табл. 1 [3].

Таблица 1

Исходные данные

X	Y	x*y	X <sup>2</sup>
1(1,1,1,1)	1(1,1,1,1)	1(1,1,1,1)	1(1,1,1,1)
2(2,2,2,2)	4(1,4,4,4)	8(2,3,1,8)	4(1,4,4,4)
3(0,3,3,3)	3(0,3,3,3)	9(0,4,2,9)	9(0,4,2,9)
4(1,4,4,4)	5(2,0,5,5)	20(2,0,6,9)	16(1,1,2,5)
5(2,0,5,5)	4(1,4,4,4)	20(2,0,6,9)	25(1,0,4,3)
$\sum 15(0,0,1,4)$	$\sum 17(2,2,3,6)$	$\sum 58(1,3,2,3)$	$\sum 55(1,0,6,0)$

Зададим значения оснований:

$$p_1 = 3; \quad p_2 = 5; \quad p_3 = 7; \quad p_4 = 11.$$

Построим в соответствии с этими данными линию регрессии  $y = a_0 + a_1 * x$ .

Система уравнений для определения коэффициентов модели по МНК имеет вид.

$$\begin{cases} na_0 + a_1 \sum_{i=1}^n x_i = \sum_{j=1}^n y_j; \\ a_0 \sum_{i=1}^n x_i + a_1 \sum_{i=1}^n x_i^2 = \sum_{i=1}^n x_i y_i. \end{cases}$$

Решая эту систему в десятичной системе счисления, получаем модель

$$y = 1,3 + 0,7x.$$

Решим эту систему в СОК:

$$\begin{aligned} (2,0,5,5)_{(5)} * a_0 + a_1 * (0,0,1,4)_{(15)} &= (2,2,3,6)_{(17)}; \\ a_1 * (0,0,1,4)_{(15)} + a_1 * (1,0,6,0)_{(55)} &= (1,3,2,3)_{(58)}. \end{aligned}$$

В нижнем индексе, в скобках, указаны значения чисел СОК в десятичной системе счисления. Выразим  $a_0$ :

$$a_0 = \frac{(2,2,3,6)_{(17)} - a_1 (0,0,1,4)_{(15)}}{(2,0,5,5)_{(5)}}.$$

Подставим это значение во второе уравнение:

$$(0, 0, 1, 4)_{(15)} * \frac{(2, 2, 3, 6)_{(17)}}{(2, 0, 5, 5)_{(5)}} - a_1 * (0, 0, 1, 4)_{(15)} * \frac{(0, 0, 1, 4)_{(15)}}{(2, 0, 5, 5)_{(5)}} + a_1 * (1, 0, 6, 0)_{(55)} = (1, 3, 2, 3)_{(58)};$$

$$(0, 1, 2, 7)_{(51)} - a_1 * (0, 0, 3, 1)_{(45)} + a_1 * (1, 0, 6, 0)_{(55)} = (1, 3, 2, 3)_{(58)};$$

$$a_1 * ((1, 0, 6, 0)_{(55)} - (0, 0, 3, 1)_{(45)}) = (1, 3, 2, 3)_{(58)} - (0, 1, 2, 7)_{(51)};$$

$$a_1 * (1, 0, 3, 10)_{(10)} = (1, 2, 0, 7)_{(7)}.$$

Тогда

$$a_1 = \frac{(1, 2, 0, 7)_{(7)}}{(1, 0, 3, 10)_{(10)}} = 0,7.$$

Далее числа СОК можно перевести в двоичную или десятичную системы счисления и там осуществить деление, либо применить алгоритм приближенного деления, реализуемого в СОК, либо использовать при делении теорию индексов, либо воспользоваться алгоритмом вычитания, реализуемого в СОК. При этом соответствующее значение  $a_0$  будет равно:

$$a_0 = \frac{(2, 2, 3, 6)_{(17)} - \frac{(1, 2, 0, 7)_{(7)}}{(1, 0, 3, 10)_{(10)}} * (0, 0, 3, 1)_{(45)}}{(2, 0, 5, 5)_{(5)}} =$$

$$\frac{(2, 2, 3, 6)_{(17)} - \frac{(1, 2, 0, 7)_{(7)}}{(2, 2, 2, 2)_{(2)}} * (0, 3, 3, 3)_{(3)}}{(2, 0, 5, 5)_{(5)}} =$$

$$\frac{\frac{(1, 4, 6, 1)_{(34)}}{(2, 2, 2, 2)_{(2)}} - \frac{(0, 1, 0, 10)_{(10)}}{(2, 2, 2, 2)_{(2)}}}{(2, 0, 5, 5)_{(5)}} =$$

$$= \frac{(1, 3, 6, 2)_{(13)}}{(1, 0, 3, 10)_{(10)}} = 1,3.$$

Используя указанный подход, можно реализовывать любые математические модели в СОК без представления операндов в формате с плавающей точкой.

Покажем возможность реализации в СОК алгоритма обратного распространения для обучения нейронных сетей.

Рассмотрим пример, демонстрирующий процесс обучения сети методом обратного распространения, реализованный в СОК. Допустим, обучающее множество состоит из одного набора  $((1,1,1,1)_{(1)}; (2,2,2,2)_{(2)})$ . Сеть должна научиться формировать на выходе число  $(0,3,3,3)_{(3)}$ .

Пусть сеть состоит из двух слоев нейронов: два нейрона в скрытом слое и один нейрон в выходном слое, как это показано на рис. 1.

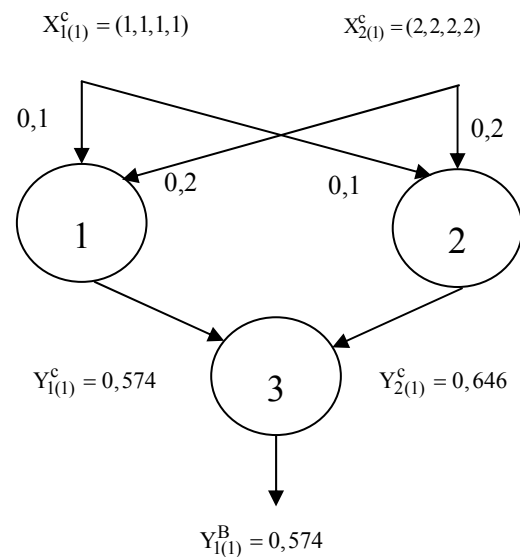


Рис. 1. Структура ИНС, используемой для тестирования

Значения весовых коэффициентов в СОК примут вид:

$$0,1 = \frac{(1,1,1,1)_{(1)}}{(1,0,3,10)_{(10)}};$$

$$0,2 = \frac{(2,2,2,2)_{(2)}}{(1,0,3,10)_{(10)}};$$

$$3 = (0,3,3,3)_{(3)}.$$

Будем использовать следующие обозначения:  $X$  – вход какого-либо нейрона, верхний идентификатор определяет слой, в котором находится нейрон ( $X^C$  – вход скрытого слоя,  $X^B$  – вход выходного слоя); индекс в скобках указывает номер итерации обучения ( $X_{(1)}^B$  – вход скрытого слоя в процессе первой итерации,  $X_{(2)}^B$  – вход выходного слоя в процессе второй итерации и т.д.);  $Y$  – выход нейрона, при этом идентификаторы и индексы здесь те же, что и у  $X$ ;  $W_i$  – синаптическая матрица, используемая на  $i$ -й итерации обучения; идентификаторы «с» и «в» указывают на синаптические веса соответствующего слоя (скрытого или выходного).

Ошибка сети или ошибка, полученная на выходе  $j$ -го нейрона, определяется следующим образом:

$$E_j = \frac{1}{2} (Y_j - D_j)^2,$$

где  $D_j$  – желаемый выход  $j$ -го нейрона.

Все нейроны будут возбуждаться с помощью одной сигмоидной функции вида:

$$F(X) = \frac{1}{1 + e^{-x}}.$$

Для реализации функции в СОК представим  $e^{-x}$  разложением в ряд Тейлора:

$$e^{-x} = 1 - x + \frac{x^2}{2} - \frac{x^3}{6} + \dots$$

Рассмотрим две итерации обучения и один обратный проход.

На первой итерации при прямом проходе на вход скрытого слоя сети подается набор обучающих данных:

$$X_{(1)}^c = [X_{1(1)}^c; X_{2(1)}^c] = [1, 2];$$

$$X_{(1)}^c = [X_{1(1)}^c; X_{2(1)}^c] = [(1, 1, 1, 1); (2, 2, 2, 2)].$$

Здесь  $X_{j(1)}^c$  – значение  $j$ -го входа скрытого слоя во время первой итерации.

Синаптическая матрица для скрытого слоя имеет вид:

$$W_{(1)}^c = \begin{bmatrix} 0,1 & 0,2 \\ 0,1 & 0,2 \end{bmatrix}.$$

Представим эту матрицу СОК:

$$W_{(1)}^c = \begin{bmatrix} \frac{(1,1,1,1)_{(1)}}{(1,0,3,10)_{(10)}} & \frac{(2,2,2,2)_{(2)}}{(1,0,3,10)_{(10)}} \\ \frac{(1,1,1,1)_{(1)}}{(1,0,3,10)_{(10)}} & \frac{(2,2,2,2)_{(2)}}{(1,0,3,10)_{(10)}} \end{bmatrix}.$$

Для выходного слоя синаптическая матрица в десятичной системе и в системе остаточных классов соответственно равна:

$$W_{(1)}^B = \begin{bmatrix} 0,2 \\ 0,1 \end{bmatrix} \dots, \quad W_{(1)}^B = \begin{bmatrix} \frac{(2,2,2,2)_{(2)}}{(1,0,3,10)_{(10)}} \\ \frac{(1,1,1,1)_{(1)}}{(1,0,3,10)_{(10)}} \end{bmatrix} \dots$$

Индекс (1) указывает на значения матрицы во время первой итерации.

Определим, чему равен взвешенный вход для каждого из нейронов скрытого слоя. Выход скрытого слоя сети, преобразованный функцией возбуждения, рассчитывается следующим образом:

$$\bar{X}_{(1)}^c = [\bar{X}_{(1)}^c; \bar{X}_{(2)}^c] = X_{(1)}^c W_{(1)}^c =$$

$$= [1; 2] \cdot \begin{bmatrix} 0,1 & 0,2 \\ 0,1 & 0,2 \end{bmatrix} = [0,3; 0,6].$$

Соответственно, в СОК:

$$\bar{X}_{(1)}^c = [(1,1,1,1); (2,2,2,2)] \times$$

$$\times \begin{bmatrix} \frac{(1,1,1,1)_{(1)}}{(1,0,3,10)_{(10)}} & \frac{(2,2,2,2)_{(2)}}{(1,0,3,10)_{(10)}} \\ \frac{(1,1,1,1)_{(1)}}{(1,0,3,10)_{(10)}} & \frac{(2,2,2,2)_{(2)}}{(1,0,3,10)_{(10)}} \end{bmatrix} =$$

$$= \left[ \frac{(0,3,3,3)_{(3)}}{(1,0,3,10)_{(10)}}, \frac{(0,1,6,6)_{(6)}}{(1,0,3,10)_{(10)}} \right];$$

$$Y_{(1)}^c = [Y_{(1)}^c; Y_{(2)}^c] =$$

$$= F(\bar{X}_{(1)}^c) = [F(\bar{X}_{(1)}^c); F(\bar{X}_{(2)}^c)] =$$

$$= [F(0,3); F(0,6)] =$$

$$= \left[ \frac{1}{1+e^{-0,3}}; \frac{1}{1+e^{-0,6}} \right] = [0,574; 0,646].$$

Соответственно, в СОК:

$$Y_{(1)}^c = \left[ \frac{1}{1+e^{\frac{(0,3,3,3)_{(3)}}{(1,0,3,10)_{(10)}}}}; \frac{1}{1+e^{\frac{(0,1,6,6)_{(6)}}{(1,0,3,10)_{(10)}}}} \right] =$$

$$= \left[ \frac{(0,2,1,2)_{(57)}}{(1,0,2,1)_{(100)}}; \frac{(2,0,2,10)_{(65)}}{(1,0,2,1)_{(100)}} \right],$$

$$\text{где } e^{-x} = 1 - x + \frac{x^2}{2} - \frac{x^3}{6} + \dots,$$

а в СОК

$$e^{\frac{(0,3,3,3)_{(3)}}{(1,0,3,10)_{(10)}}} = (1,1,1,1) - \frac{(0,3,3,3)_{(3)}}{(1,0,3,10)_{(10)}} +$$

$$+ \frac{\left( \frac{(0,3,3,3)_{(3)}}{(1,0,3,10)_{(10)}} \right)^2}{(2,2,2,2)_{(2)}} - \frac{\left( \frac{(0,3,3,3)_{(3)}}{(1,0,3,10)_{(10)}} \right)^3}{(0,1,6,6)_{(6)}}.$$

Для второй итерации применим ту же методику. Выход скрытого слоя является входом для выходного слоя, т.е.

$$X_{(1)}^B = Y_{(1)}^c.$$

Как и для скрытого слоя, рассчитаем взвешенный вход для выходного слоя

$$X_{(1)}^B = [\bar{X}_{(1)}^B; \bar{X}_{(2)}^B] = X_{(1)}^B W_{(1)}^B =$$

$$= [0,574; 0,646] \cdot \begin{bmatrix} 0,2 \\ 0,1 \end{bmatrix} = [0,179].$$

Соответственно, в СОК:

$$X_{(1)}^B = [\bar{X}_{(1)}^B; \bar{X}_{(2)}^B] = Y_{(1)}^B W_{(1)}^B =$$

$$= \left[ \frac{(0,2,1,2)_{(57)}}{(1,0,2,1)_{(100)}}; \frac{(2,0,2,10)_{(65)}}{(1,0,2,1)_{(100)}} \right] \times$$

$$\times \begin{bmatrix} \frac{(2,2,2,2)_{(2)}}{(1,0,3,10)_{(10)}} \\ \frac{(1,1,1,1)_{(1)}}{(1,0,3,10)_{(10)}} \end{bmatrix} =$$

$$= \left[ \frac{(0,3,4,7)_{(18)}}{(1,0,2,1)_{(100)}} \right].$$

Здесь  $W_{(1)}^B$  – синаптическая матрица выходного слоя сети, значения которой соответствуют первой итерации.

Функция возбуждения для 3-го нейрона остается прежней, поэтому для него получим следующий выход:

$$Y_{(1)}^B - F(\bar{X}_{(1)}^B) = F(0, 179) = \frac{1}{1 + e^{-0,179}} = 0,545.$$

Соответственно, в СОК:

$$Y_{(1)}^B - F(\bar{X}_{(1)}^B) = F\left(\frac{(0,3,4,7)_{(18)}}{(1,0,2,1)_{(100)}}\right) = \frac{1}{1 + e^{\frac{(0,3,4,7)_{(18)}}{(1,0,2,1)_{(100)}}}} = \frac{(1,0,6,0)_{(55)}}{(1,0,2,1)_{(100)}}.$$

В то же время желаемое значение выхода сети, преобразованное функцией возбуждения, равно  $D = F(3) = 0,952$ , а в СОК:

$$D = F(0,3,3,3) = \frac{(2,0,4,7)_{(95)}}{(1,0,2,1)_{(100)}}.$$

При этом среднеквадратичная ошибка в десятичной системе и в СОК соответственно равна:

$$E = \frac{1}{2}(0,545 - 0,952)^2 = 0,0828;$$

$$E = \frac{(2,3,1,8)_{(8)}}{(1,0,2,1)_{(100)}}.$$

Значения фактического и желаемого значений выхода совпадают не полностью. Улучшить результат можно изменением синаптических весов сети. Для этого необходимо выяснить, каким образом повлияют эти изменения на общий результат, т.е. на величину ошибки. Такой анализ в соответствии со схемой алгоритма обратного распространения выполняются, начиная с выходного слоя сети.

## Выводы

Рассмотренный подход реализации модульных вычислений для задач прогнозирования на нейронных сетях, позволяет получить ряд существенных преимуществ:

- параллельная структура нейронных сетей повышает общую надежность прогнозирующей модели;

- используемая в вычислениях система остаточных классов обладает способностью самокоррекции, обнаружения и исправления ошибок;

- система остаточных классов, являясь непозиционной, позволяет сократить разрядность обрабатываемых данных, представляя их в модулярном коде совокупностью взаимно простых модулей. В связи с этим становится возможным упростить структуру нейронной сети и избежать вычислительных проблем, связанных с необходимостью масштабирования входных данных.

## Список литературы

1. Червяков Н.И. Применение нейронных сетей для задач прогнозирования и проблемы идентификации моделей прогнозирования / Н.И. Червяков, Э.Е. Тихонов // *Нейрокомпьютеры: разработка, применение*. – М.: Радиотехника, 2003. – № 10. – С. 25-31.
2. Барский А.Б. Нейронные сети: распознавание, обучение, принятие решений / А.Б. Барский. – М.: Финансы и статистика, 2004. – 320 с.
3. Суровцев И.С. Нейронные сети / И.С. Суровцев, В.И. Клюкни, Р.П. Пивоварова. – Воронеж: ВГУ, 2004. – 224 с.

Поступила в редколлегию 12.11.2008

**Рецензент:** д-р техн. наук, проф. М.В. Алипов, Харьковский национальный университет радиоэлектроники, Харьков.

## НЕПОЗИЦІЙНЕ КОДУВАННЯ У ЗАДАЧАХ НЕЙРОМЕРЕЖЕВОГО ПРОГНОЗУВАННЯ ЧАСОВИХ РЯДІВ

Є.В. Бодянський, О.С. Удовенко

*Розглядається метод модульного кодування інформації у прогнозуючих моделях, що реалізуються за допомогою штучних нейромереж. Переваги запропонованого підходу полягають у спрощенні структури нейромережі, скороченні розрядності даних та можливості використання паралельних обчислювальних процедур, що дозволяє підвищити загальну ефективність системи прогнозування.*

**Ключові слова:** модулярний код, штучна нейронна мережа, система залишкових класів, синаптична матриця.

## NOPOSITION ENCODING IN TASKS OF NEURAL NETWORKS PREDICTION OF TIME SERIES

E.V. Bodyansky, O.S. Udovenko

*Is considered one of the algorithms of evaluation and determinations of acceptability of risk level when ensuring information safety. The method of the module information encoding is examined in forecasting models which will be realized on artificial neural networks. Advantages of offered approach consist of simplification of structure of neural networks, diminishing of data and possibility of the use of parallel computable procedur, that allows to promote general efficiency of the system of prediction.*

**Keywords:** modular code, artificial neuron network, system of remaining classes, synaptic matrix.