



ПРОГНОЗИРУЮЩАЯ АДАПТИВНАЯ НЕЙРОННАЯ СЕТЬ С ДИНАМИЧЕСКИМИ НЕЙРОНАМИ-ФИЛЬТРАМИ

БОДЯНСКИЙ Е.В., ПОПОВ С.В., ЧЕПЕНКО Т.Е.

Рассматривается задача прогнозирования в реальном времени нестационарных временных последовательностей в условиях структурной и параметрической неопределенности. Предлагается модифицированный алгоритм обучения прогнозирующей нейронной сети, элементами которой являются динамические нейроны – фильтры с конечной импульсной характеристикой – КИХ-нейроны.

Задача прогнозирования нестационарных временных рядов по их предыстории, заданной в виде растущей выборки наблюдений $x(1), x(2), x(k), \dots$, часто встречается в различных областях науки, техники и экономики, а для ее решения наработан достаточно мощный арсенал разнообразных методов.

В случае, если прогнозирование необходимо производить в реальном времени по мере поступления новых данных, весьма эффективным представляется адаптивный подход, возможности которого ограничены требованием априорного задания структуры математической модели прогнозируемого процесса. Если же природа временной последовательности неизвестна, наиболее адекватным в данной ситуации является применение искусственных нейронных сетей (ИНС), позволяющих по прошлым наблюдениям восстановить нелинейное отображение вида

$$x(k) = \mathcal{N}(x(k-1), x(k-2), \dots, x(k-n_A)) + e(k) = \hat{x}(k) + e(k), \quad (1)$$

где $\hat{x}(k)$ – оценка (прогноз) значения $x(k)$, полученная на выходе искусственной нейронной сети, представляющей в данном случае нелинейную авторегрессионную модель; $e(k)$ – ошибка прогнозирования.

Эффективность использования НАР-модели в задачах прогнозирования определяется теоремой Тэ-кенса о диффеоморфизме [1], устанавливающей существование порядка n_A , который обеспечивает

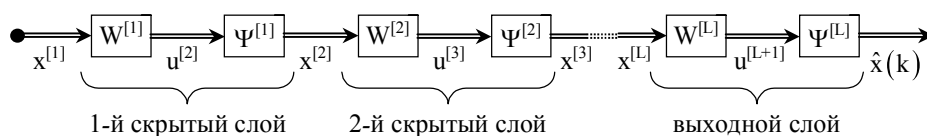


Рис. 1. L-слойная ИНС с прямой передачей информации

сколь угодно малые значения ошибки $e(k)$, и универсальными аппроксимирующими свойствами нейронных сетей [2, 3]. В качестве основы для построения НАР-моделей обычно используются многослойные ИНС с прямой передачей информации, входной (нулевой) слой которых образован линиями элементов чистой задержки с отводами. На рис. 1. приведена обобщенная архитектура

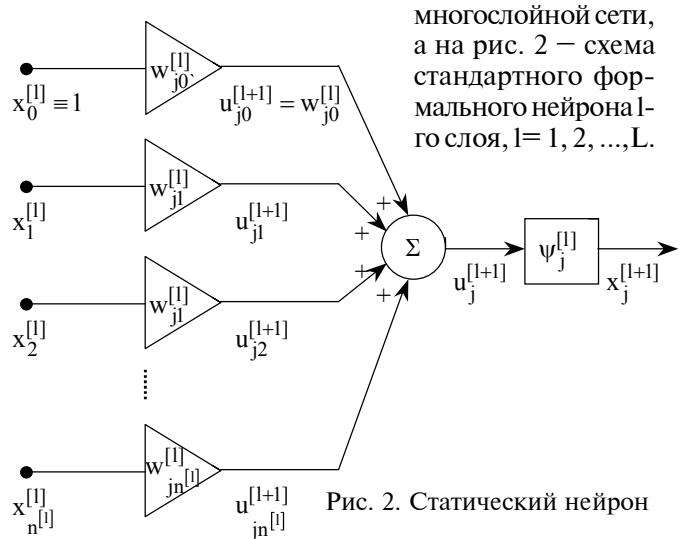


Рис. 2. Статический нейрон

многослойной сети, а на рис. 2 – схема стандартного формального нейрона l -го слоя, $l = 1, 2, \dots, L$. На первый скрытый слой сети поступает $n = n^{[1]} = n_A = n_A^{[1]}$ -мерный вектор $x^{[1]}$, сформированный в нулевом слое с помощью элементов задержки z^{-1} ($z^{-1}x(k) = x(k-1)$) и образованный прошлыми значениями прогнозируемого процесса $x(k-1), x(k-2), \dots, x(k-n_A^{[1]})$. Выходным сигналом первого скрытого слоя является $(n^{[2]} + 1)$ -вектор $x^{[2]}$, подающийся на вход второго скрытого слоя и т.д. На выходе L -го (выходного) слоя появляется прогнозный $m = n^{[L+1]}$ -мерный вектор $\hat{x}(k)$. Таким образом, каждый слой имеет $n^{[l]}$ входов и $n^{[l+1]}$ выходов и характеризуется $((n^{[l]} + 1) \times n^{[l+1]})$ -матрицей синаптических весов $W^{[l]}$ и $(n^{[l+1]} \times n^{[l+1]})$ -диагональным оператором $\Psi^{[l]}$, образованным нелинейными функциями активации $\psi_j^{[l]}, j = 1, 2, \dots, n^{[l+1]}$.

“Строительным блоком” такой сети является стандартный статический нейрон, реализующий нелинейное отображение

$$x_j^{[l+1]} = \psi_j^{[l]}(u_j^{[l+1]}) = \psi_j^{[l]} \left(\sum_{i=0}^{n^{[l]}} u_{ji}^{[l+1]} \right) = \psi_j^{[l]} \left(\sum_{i=0}^{n^{[l]}} w_{ji}^{[l]} x_i^{[l]} \right),$$

$n^{[l]} + 1$ синаптических весов $w_{ji}^{[l]}$ которого подлежат уточнению в процессе обучения нейронной сети.

Всего сеть содержит $\sum_{l=1}^L (n^{[l]} + 1) n^{[l+1]}$ неизвестных

параметров, настраиваемых с помощью процедуры обратного распространения ошибок.

Общим недостатком прогнозирующих сетей, построенных на статических нейронах, является чрезвычайно большое количество настраиваемых весов и низкая скорость обучения, что, естественно, вызывает серьезные проблемы, особенно при работе в реальном времени. В связи с этим Э. Ваном [4-7] было предложено в прогнозирующих нейронных сетях вместо стандартных статических нейронов использовать их динамические аналоги, у которых синаптические веса образованы цифровыми фильтрами с конечной импульсной характеристикой (КИХ-фильтры) так, как это показано на рис. 3.

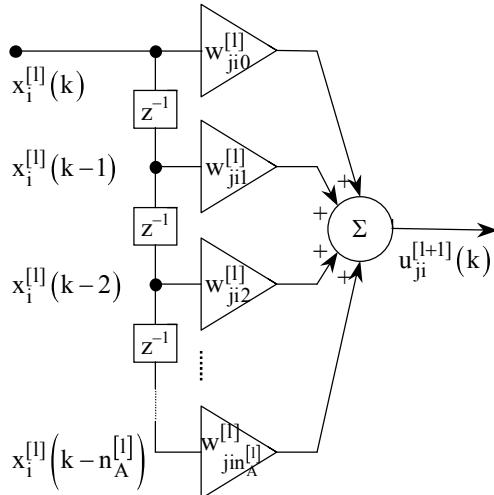


Рис. 3. Синапс – КИХ-фильтр

Несложно видеть, что

$$u_{ji}^{[l+1]}(k) = w_{ji0}^{[l]}x_i^{[l]}(k) + w_{ji1}^{[l]}x_i^{[l]}(k-1) + \dots + w_{jin_A}^{[l]}x_i^{[l]}(k-n_A^{[l]})$$

или $u_{ji}^{[l+1]}(k) = W_{ji}^{[l]T} X_i^{[l]}(k)$, где

$$W_{ji}^{[l]} = \left(w_{ji0}^{[l]}, w_{ji1}^{[l]}, \dots, w_{jin_A}^{[l]} \right)^T,$$

$$X_i^{[l]}(k) = \left(x_i^{[l]}(k), x_i^{[l]}(k-1), \dots, x_i^{[l]}(k-n_A^{[l]}) \right)^T -$$

$\left((n_A^{[l]}+1) \times 1 \right)$ -векторы.

Тогда нелинейное отображение, реализуемое динамическим КИХ-нейроном, условное изображение которого приведено на рис. 4, можно записать в виде

$$x_j^{[l+1]}(k) = \psi_j^{[l]} \left(u_j^{[l+1]}(k) \right) = \psi_j^{[l]} \left(\sum_{i=0}^{n_A^{[l]}} u_{ji}^{[l+1]}(k) \right) = \psi_j^{[l]} \left(\sum_{i=0}^{n_A^{[l]}} W_{ji}^{[l]T} X_i^{[l]}(k) \right).$$

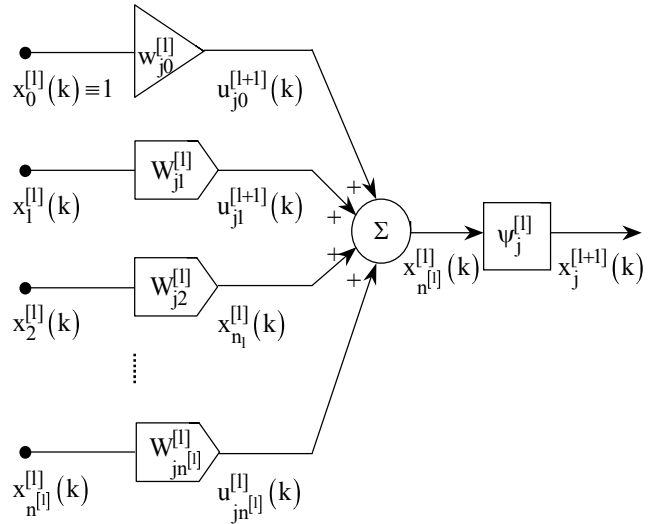


Рис. 4. Динамический КИХ-нейрон

И хотя динамический нейрон содержит $n^{[l]} \left(n_A^{[l]} + 1 \right) + 1$ параметров, что превышает количество синаптических весов стандартного нейрона, сеть, построенная из таких узлов, содержит много меньше параметров, чем стандартная архитектура на статических нейронах с линиями задержки на входе, реализующая те же функции. В [5] было доказано, что в сети на стандартных нейронах количество параметров растет геометрически в зависимости от n_A , в то время как в ИНС с динамическими нейронами число настраиваемых синаптических весов есть линейная функция от n_A и L .

Для обучения ИНС на динамических нейронах в [4, 5] была введена градиентная процедура, получившая название обратного распространения ошибок во времени (Temporal error backpropagation – ТЕВР). Вводя в рассмотрение стандартный одношаговый критерий обучения

$$J(k) = \frac{1}{2} \|e(k)\|^2 = \frac{1}{2} \|d(k) - \mathcal{N}(X^{[1]}, W^{[1]}, \dots, W^{[L]})\|^2 = \frac{1}{2} \|d(k) - \hat{x}(k)\|^2$$

(здесь $d(k)$ – обучающий сигнал, в качестве которого в задачах прогнозирования принимается текущее значение $x(k)$), можно записать в общем виде алгоритм его минимизации:

$$W_{ji}^{[l]}(k+1) = W_{ji}^{[l]}(k) - \eta^{[l]}(k) \nabla_{W_{ji}^{[l]}} J(k),$$

где $\eta^{[l]}(k)$ – параметр, определяющий скорость сходимости процесса обучения и полагаемый в ТЕВР-процедуре постоянным [4-7].

Применяя далее цепное правило в виде

$$\nabla_{W_{ji}^{[l]}} J(k) = \frac{\partial J(k)}{\partial u_j^{[l+1]}(k)} \nabla_{W_{ji}^{[l]}} u_j^{[l+1]}(k),$$

получаем

$$W_{ji}^{[l]}(k+1) = W_{ji}^{[l]}(k) - \eta^{[l]}(k) \frac{\partial J(k)}{\partial u_j^{[l+1]}(k)} \nabla_{W_{ji}^{[l]}} u_j^{[l+1]}(k). \quad (2)$$

С учетом того, что

$$u_j^{[l+1]}(k) = \sum_i u_{ji}^{[l+1]}(k) = \sum_i W_{ji}^{[l]T} X_i^{[l]}(k)$$

и $\nabla_{W_{ji}^{[l]}} u_j^{[l+1]}(k) = X_i^{[l]}(k)$, можно ввести в рассмотрение локальную ошибку $\frac{\partial J(k)}{\partial u_j^{[l+1]}(k)} = \delta_j^{[l]}(k)$ и окончательно представить алгоритм обучения (2) в форме $W_{ji}^{[l]}(k+1) = W_{ji}^{[l]}(k) - \eta^{[l]}(k) \delta_j^{[l+1]}(k) X_i^{[l]}(k)$.

Процесс вычисления локальных ошибок, следуя обычной методике обратного распространения, начнем с выходного слоя, выходной сигнал которого является и выходом сети в целом, т.е. $\hat{x}(k) = x^{[L+1]}(k)$.

Поскольку

$$\begin{aligned} \delta_j^{[L+1]}(k) &= \frac{\partial J(k)}{\partial u_j^{[L+1]}(k)} = \frac{1}{2} \frac{\partial \|e(k)\|^2}{\partial u_j^{[L+1]}(k)} = \\ &= -e_j(k) \psi_j^{[L]} \left(u_j^{[L+1]}(k) \right), \end{aligned}$$

алгоритм настройки нейронов выходного слоя сразу же можно записать в виде

$$\begin{aligned} W_{ji}^{[L]}(k+1) &= \\ &= W_{ji}^{[L]}(k) + \eta^{[L]}(k) e_j(k) \psi_j^{[L]} \left(u_j^{[L+1]}(k) \right) X_i^{[L]}(k) = \\ &= W_{ji}^{[L]}(k) + \eta^{[L]}(k) e_j(k) J_i^{[L]}(k). \end{aligned} \quad (3)$$

Скорость настройки сети существенным образом зависит от значения скалярного параметра $\eta^{[L]}(k)$ и может быть увеличена путем использования процедур второго порядка, среди которых наиболее распространен алгоритм Левенберга-Марквардта [8], принимающий в данном случае вид

$$\begin{aligned} W_{ji}^{[L]}(k+1) &= W_{ji}^{[L]}(k) + \\ &+ e_j(k) \left(J_i^{[L]}(k) J_i^{[L]T}(k) + \alpha I \right)^{-1} J_i^{[L]}(k), \end{aligned} \quad (4)$$

где $\alpha > 0$ – регуляризирующий параметр; I – единичная матрица соответствующего размера.

Хотя данный алгоритм обладает высокой скоростью сходимости, его использование в реальном времени ограничивается необходимостью обращения на каждом такте $(n^{[L]} \times n^{[L]})$ -матрицы. Используя для обращения формулу Шермана-Моррисона и проводя цепочку очевидных преобразований, можно получить простое выражение

$$\left(J_i^{[L]}(k) J_i^{[L]T}(k) + \alpha I \right)^{-1} J_i^{[L]}(k) = \frac{J_i^{[L]}(k)}{\alpha + \left\| J_i^{[L]}(k) \right\|^2},$$

после чего переписать алгоритм (4) в виде

$$W_{ji}^{[L]}(k+1) = W_{ji}^{[L]}(k) + \frac{e_j(k) J_i^{[L]}(k)}{\alpha + \left\| J_i^{[L]}(k) \right\|^2},$$

который является обобщением процедуры обучения многослойных ИНС, введенной в [9], а при $\alpha \rightarrow 0$ представляет нелинейное расширение алгоритма Уидроу-Хоффа.

Для скрытых слоев сети локальная ошибка может быть записана в форме

$$\begin{aligned} \delta_j^{[l]}(k) &= \frac{\partial J(k)}{\partial u_j^{[l]}(k)} = \sum_{q=1}^{n^{[l+1]}} \sum_{t=k}^{n_A^{[l]}+k} \frac{\partial J(k)}{\partial u_q^{[l+1]}(t)} \frac{\partial u_q^{[l+1]}(t)}{\partial u_j^{[l]}(k)} = \\ &= \sum_{q=1}^{n^{[l+1]}} \sum_{t=k}^{n_A^{[l]}+k} \delta_q^{[l+1]}(t) \frac{\partial u_q^{[l+1]}(t)}{\partial u_j^{[l]}(k)} = \\ &= \psi_j^{[l-1]} \left(u_j^{[l]}(k) \right) \sum_{q=1}^{n^{[l+1]}} \sum_{t=k}^{n_A^{[l]}+k} \delta_q^{[l+1]}(t) \frac{\partial u_{jq}^{[l+1]}(t)}{\partial x_j^{[l]}(k)}. \end{aligned}$$

С учетом соотношения

$$u_{jq}^{[l+1]}(k) = \sum_{k'=0}^{n_A^{[l]}} w_{jqk'}^{[l]} x_j^{[l]}(t-k')$$

получаем выражение

$$\frac{\partial u_{jq}^{[l+1]}(t)}{\partial x_j^{[l]}(k)} = \begin{cases} w_{jqt-k}^{[l]} & \text{при } 0 \leq t-k \leq n_A^{[l]}, \\ 0 & \text{в противном случае,} \end{cases}$$

откуда локальная ошибка окончательно приобретает вид

$$\begin{aligned} \delta_j^{[l]}(k) &= \psi_j^{[l-1]} \left(u_j^{[l]}(k) \right) \sum_{q=1}^{n^{[l+1]}} \sum_{t=k}^{n_A^{[l]}+k} \delta_q^{[l+1]}(t) w_{jqt-k}^{[l]} = \\ &= \psi_j^{[l-1]} \left(u_j^{[l]}(k) \right) \sum_{q=1}^{n^{[l+1]}} \sum_{r=0}^{n_A^{[l]}} \delta_q^{[l+1]}(k+r) w_{jqr}^{[l]} = \\ &= \psi_j^{[l-1]} \left(u_j^{[l]}(k) \right) \sum_{q=1}^{n^{[l+1]}} \Delta_q^{[l+1]T}(k) W_{jq}^{[l]}, \end{aligned} \quad (5)$$

где $\Delta_q^{[l]}(k) = \left(\delta_q^{[l]}(k), \delta_q^{[l]}(k+1), \dots, \delta_q^{[l]}(k+n_A^{[l-1]}) \right)^T$.

С учетом (5) алгоритм обратного распространения ошибок во времени может быть записан в виде

$$W_{ji}^{[l]}(k+1) = W_{ji}^{[l]}(k) - \eta^{[l]}(k) \delta_j^{[l+1]}(k) X_i^{[l]}(k), \quad (6)$$

здесь

$$\delta_j^{[l]}(k) = \begin{cases} -e_j(k) \psi_j^{[l-1]} \left(u_j^{[l]}(k) \right), & l=L, \\ \psi_j^{[l-1]} \left(u_j^{[l]}(k) \right) \sum_{q=1}^{n^{[l+1]}} \Delta_q^{[l+1]T}(k) w_{jq}^{[l]}, & 1 \leq l \leq L-1. \end{cases}$$

В целях оптимизации по скорости процесса обучения в скрытых слоях перепишем алгоритм их настройки в виде

$$\begin{aligned}
W_{ji}^{[l]}(k+1) &= W_{ji}^{[l]}(k) - \\
&- \eta^{[l]}(k) \left(\sum_{q=1}^{n^{[l+1]}} \Delta_q^{[l+1]T}(k) W_{jq}^{[l]} \right) \psi_j^{[l-1]}(u_j^{[l]}(k)) X_i^{[l]}(k) = \\
&= W_{ji}^{[l]}(k) + \eta^{[l]}(k) e_j^{[l]}(k) \psi_j^{[l-1]}(u_j^{[l]}(k)) X_i^{[l]}(k) = \\
&= W_{ji}^{[l]}(k) + \eta^{[l]}(k) e_j^{[l]}(k) J_i^{[l]}(k),
\end{aligned}$$

после чего, применяя алгоритм Левенберга-Марквардта и формулу Шермана-Моррисона, окончательно получаем простую адаптивную процедуру:

$$W_{ji}^{[l]}(k+1) = W_{ji}^{[l]}(k) + \frac{e_j^{[l]}(k) J_i^{[l]}(k)}{\alpha + \|J_i^{[l]}(k)\|^2}, \quad 1 \leq l \leq L. \quad (7)$$

Чтобы придать алгоритму (7) сглаживающие свойства, которые необходимы при обработке “зашумленных” сигналов, процедуру обучения можно модифицировать следующим образом [10]:

$$\begin{cases}
W_{ji}^{[l]}(k+1) = W_{ji}^{[l]}(k) + \frac{e_j^{[l]}(k) J_i^{[l]}(k)}{\alpha_i^{[l]}(k)}, \\
\alpha_i^{[l]}(k+1) = \beta \alpha_i^{[l]}(k) + \|J_i^{[l]}(k)\|^2, \quad 0 \leq \beta \leq 1,
\end{cases} \quad (8)$$

где β – параметр сглаживания.

Несложно видеть, что при $\beta=0$ приходим к нелинейной форме алгоритма Уидроу-Хоффа, при $\beta=1$ получаем процедуру стохастической аппроксимации.

Сеть на динамических нейронах может работать в двух режимах: обучения и собственно прогнозирования, при этом благодаря “внутренней памяти” КИХ-нейронов на вход ИНС достаточно подавать лишь одно значение прогнозируемой последовательности так, как это показано на рис. 5.

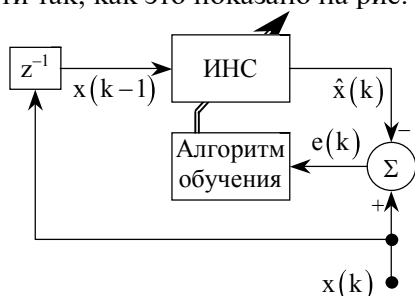


Рис. 5. Нейронная сеть в режиме обучения

Более глубокая предыстория сигнала формируется в динамических нейронах скрытых слоев. Таким образом, при прогнозировании одномерных временных рядов ИНС имеет только один вход, в то время как использование статических нейронов приводит к тому, что сеть должна иметь как минимум n_d входов.

Режим прогнозирования, иллюстрируемый рис. 6, реализуется еще проще, при этом выходной сигнал (прогноз) сети по обратной связи через элемент задержки z^{-1} подается на ее вход.

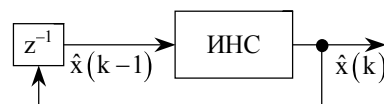


Рис. 6. Нейронная сеть в режиме прогнозирования

Таким образом, прогнозирующая адаптивная нейронная сеть с динамическими нейронами – КИХ-фильтрами, обладая стандартной архитектурой многослойной ИНС с прямой передачей информации, имеет меньшее количество настраиваемых синаптических весов и характеризуется высокой скоростью обучения.

Литература: 1. Packard N., Crutchfield J., Farmer J., Shaw R. Geometry from a time series // Phys. Rev. Lett. 1980. 45. P. 712-716. 2. Hornik K., Stinchcombe M., White H. Multilayer feedforward networks are universal approximators // Neural Networks. 1989. 2. P. 359-366. 3. Cybenko G. Approximation by superpositions of a sigmoidal function // Mathematics of Control, Signals and Systems. 1989. 2. N4. P. 303-314. 4. Wan E. Temporal backpropagation for FIR neural networks // Int. Joint Conf. on Neural Networks. V. 1. San Diego, 1990. P. 575-580. 5. Gershenfeld S.F. Studies in the Sciences of Complexity. Vol. XVII. Reading: Addison-Wesley, 1994. P. 195-218. 6. Back A.D., Wan E.A., Lawrence S., Tsoi A.C. A unifying view of some training algorithms for multilayer perceptrons with FIR filter synapses / Ed. by J. Vlontzes, J. Hwang, E. Wilson “Neural Networks for Signal Processing”. N.J.: IEEE Press, 1994. P. 146-154. 7. Yu H.-Y., Bang S.-Y. An improved time series prediction by applying the layer-by-layer learning method to FIR neural networks // Neural Networks. 1997. 10. P. 1717-1729. 8. Nelles O. Nonlinear System Identification. Berlin: Springer, 2001. 785 p. 9. Бодянский Е.В., Запорожец О.В. Адаптивный нейрорегулятор для нелинейного динамического объекта // Изв. РАН. Теория и системы управления. 2002. N 2. С. 92-96. 10. Bodyanskiy Ye., Kolodyazhnyi V., Stephan A. An adaptive learning algorithm for a neuro-fuzzy network // Ed. by B. Reusch “Computational Intelligence. Theory and Applications”. Berlin-Heidelberg-New York: Springer, 2001. P. 68-75.

Поступила в редколлегию 24.03.2003.

Рецензент: д-р техн. наук, проф. Алексеев О.П.

Бодянский Евгений Владимирович, д-р техн. наук, профессор кафедры искусственного интеллекта, научный руководитель ПНИЛ АСУ ХНУРЭ, член IEEE. Научные интересы: адаптивные системы, искусственные нейронные сети. Адрес: Украина, 61166, Харьков, пр. Ленина, 14, тел. 702-18-90. E-mail: Bodyanskiy@ieee.org

Попов Сергей Витальевич, канд. техн. наук, старший научный сотрудник ПНИЛ АСУ ХНУРЭ, член IEEE. Научные интересы: обработка и прогнозирование стохастических процессов и полей на основе адаптивных и нейросетевых технологий. Адрес: Украина, 61166, Харьков, пр. Ленина, 14, тел. 702-18-90. E-mail: Serge.Popov@ieee.org

Чепенко Татьяна Евгеньевна, аспирантка кафедры искусственного интеллекта ХНУРЭ. Научные интересы: прогнозирование временных рядов различной физической природы. Адрес: Украина, 61166, Харьков, пр. Ленина, 14.