



УДК 681.5.13

П.И. Кравец, канд. техн. наук, **В.Н. Шимкович**, аспирант
Национальный технический университет Украины
«Киевский политехнический институт»
(Украина, 03056, Киев, пр-т Победы, 37,
тел. 4068346, e-mail: kravets@ntu-kpi.kpi.ua)

Метод оптимизации весовых коэффициентов нейронных сетей с помощью генетического алгоритма при реализации на программируемых логических интегральных схемах

Предложен метод оптимизации весовых коэффициентов нейронной сети с помощью генетического алгоритма при реализации на программируемых логических интегральных схемах (ПЛИС). Приведены примеры аппаратно-программной реализации средствами ПЛИС настройки весовых коэффициентов нейронной сети с помощью генетического алгоритма. Показано, что распараллеливанием вычислений обеспечивается значительное ускорение процедуры настройки весовых коэффициентов нейронной сети.

Запропоновано метод оптимізації вагових коефіцієнтів нейронної мережі за допомогою генетичного алгоритму при їх реалізації на програмованих логічних інтегральних схемах (ПЛІС). Наведено приклади апаратно-програмної реалізації засобами ПЛІС налаштування вагових коефіцієнтів нейронної мережі за допомогою генетичного алгоритму. Показано, що розпаралелюванням обчислень забезпечується значне прискорення процедури налаштування вагових коефіцієнтів нейронної мережі.

К л ю ч е в ы е с л о в а: нейронные сети, генетический алгоритм, ПЛИС.

Нейронные сети широко используются для выполнения «мягких» вычислений, решения задач классификации и распознавания, аппроксимации и оптимизации функций, реализации систем управления [1, 2]. В частности высокое быстродействие нейронных сетей в сочетании со способностью к обучению делает их весьма привлекательными для создания самонастраиваемых и адаптивных систем управления сложными нелинейными динамическими объектами, функционирующими в условиях неопределенности [1, 2].

В настоящее время основным методом реализации искусственных нейронных сетей является программный метод с использованием компьютерной техники или специализированных контроллеров, построенных на

© П.И. Кравец, В.Н. Шимкович, 2013

ее основе. Однако высокая стоимость систем управления значительно ограничивает их практическую реализацию и делает их использование нецелесообразным. Кроме того, компьютерные нейросетевые системы управления имеют ограниченную производительность и требуются значительные затраты времени на их обучение. Это в большинстве случаев не позволяет согласовывать быстрдействие процедуры перенастройки нейросетевых систем управления с динамикой объекта управления. Единственной альтернативой этому является реализация искусственных нейронных сетей на нейрочипах или программируемых логических интегральных схемах (ПЛИС), которые позволяют распараллелить процедуры функционирования и настройки внутренних элементов нейронных сетей и тем самым снять проблему быстрого их обучения [3, 4].

Обучением искусственной нейронной сети является процесс оптимизации весовых коэффициентов ее базовых процессорных элементов. Многообразие алгоритмов обучения определяется функциональным назначением сети, ее архитектурой и выбранной стратегией обучения. Известные методы, используемые при обучении нейронных сетей, аналогичны методам определения экстремума функции многих переменных. В частности это градиентные методы, квазиньютоновы алгоритмы (алгоритм метода секущих плоскостей OSS, алгоритм LM Левенберга—Марквардта) [5].

Однако в некоторых случаях (многоэкстремальность целевой функции, недифференцируемость функций активации нейронов и др.), когда применение итеративных градиентных методов локального поиска невозможно, целесообразно использовать методы эволюционного поиска, не требующие вычисления значений производных целевых функций.

Методы эволюционной оптимизации используются для решения различных задач, связанных с синтезом нейросетей: отбор признаков, настройка весов, выбор оптимальной архитектуры сети, адаптация учебного правила, инициализация значений весовых коэффициентов, извлечение правил построенной сети, поиск оптимальных значений параметров исследуемой системы по имеющейся нейромодели и др. [6].

Известные способы реализации эволюционных методов настройки нейронных сетей, ориентированные на использование компьютерной программы, недостаточно «быстрые» для настройки нейросетевых систем управления динамическими объектами [6, 7].

Предлагается разработанный метод аппаратно-программной реализации на ПЛИС эволюционных методов обучения нейронных сетей, предназначенных для построения нейросетевых систем управления, которые могут обучаться и функционировать в одном темпе с процессом управления динамическими объектами.

Результаты исследований. Наиболее известным из эволюционных методов в настоящее время является генетический алгоритм для нахождения глобального экстремума многоэкстремальной функции [8]. Он заключается в параллельной обработке множества альтернативных решений, при этом поиск концентрируется на наиболее перспективных из них. Это свидетельствует о возможности использования генетических алгоритмов при решении любых задач оптимизации и принятия решений.

Настройка синаптических весов нейросети с использованием генетического алгоритма, как правило, выполняется в такой последовательности [9].

Шаг 1. Выполнить инициализацию начальной популяции хромосомами, содержащими информацию о значениях весовых коэффициентов сети заданной структуры.

Шаг 2. Оценить хромосомы текущей популяции.

Шаг 2.1. Декодировать каждую хромосому популяции в набор весовых коэффициентов нейронной сети.

Шаг 2.2. Построить нейросети, соответствующие оцениваемым хромосомам.

Шаг 2.3. Вычислить значение фитнес-функции (ошибки) оцениваемых хромосом, учитывающее ошибку и сложность сети.

Шаг 3. Проверить критерии окончания поиска (достижение приемлемого значения ошибки синтезированной нейросетевой модели, превышение максимально допустимого числа итераций, превышение допустимого времени функционирования метода). Если критерии окончания поиска выполнены, перейти к шагу 7.

Шаг 4. Исходя из значения фитнес-функции выбрать особи для генерации новых решений.

Шаг 5. Применить операторы скрещивания и мутации для хромосом, отобранных на предыдущем шаге.

Шаг 6. Сформировать новое поколение из элитных хромосом и хромосом-потомков, полученных путем применения скрещивания и мутации. Перейти к выполнению шага 2.

Шаг 7. Окончание поиска.

Хромосома состоит из K генов, содержащих значения весов и смещений всех нейронов сети (рис. 1). При этом для представления значений весовых коэффициентов в хромосомах применяется действительное кодирование. Размер хромосомы определяется по формуле

$$K = N_1(L+1) + \sum_{\mu=1}^M N_{\mu} (N_{\mu-1} + 1),$$

где N_{μ} — число нейронов на μ -м слое; L — число признаков в обучающей выборке; M — число слоев нейронной сети.

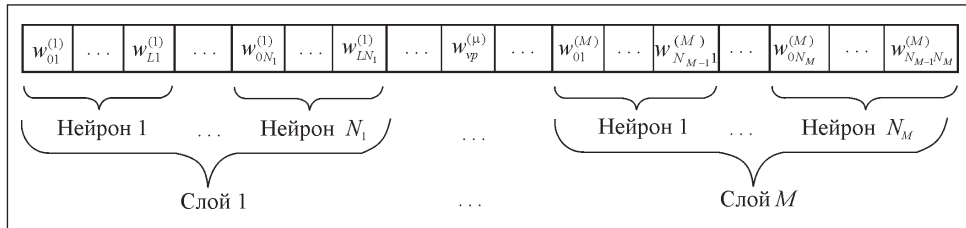


Рис. 1. Схематическое представление хромосомы

Для аппаратно-программной реализации на ПЛИС генетического алгоритма требуется распараллелить и приспособить для реализации все упомянутые выше операции. Подготовка аппаратно-программной реализации процедуры настройки нейронной сети с помощью генетического алгоритма осуществляется на основе графа (рис. 2) $GDF = (A, D)$, где A — множество вершин, соответствующих операциям; D — множество дуг, соответствующих потокам данных.

Для аппаратно-программной реализации на ПЛИС генетического алгоритма настройки синаптических весов нейросети необходимо разработать генератор случайных чисел. Генерация случайных чисел выполняется на основе мультипликативного конгруэнтного метода, в соответствии с которым каждое последующее число формируется на базе предыдущего по формуле $r_{i+1} = \text{mod}(k r_i, M)$, где M — модуль, $0 < M$; k — множитель, $0 \leq k < M$.

Число M должно быть достаточно большим, поскольку период генерации чисел не может иметь более M элементов. Логично выбрать $M = 2^N$, поскольку в этом случае операции деления, требуемые в процедурах вычислений, для ПЛИС не являются стандартными и могут быть заменены операциями сдвига.

Для генетического алгоритма необходимо генерировать случайные числа в диапазоне от нуля до семи. Их можно получить по формуле $r_i^* = \text{mod}(r_i, 8)$, т.е. используя три младших разряда сгенерированного случайного числа. Это возможно согласно теореме о том, что при использовании конгруэнтного метода генерации случайных чисел младшие разряды полученного случайного числа ведут себя так же случайно, как и старшие [10, 11]. Число восемь определяет число бит в хромосоме. Если хромосома будет состоять из другого числа бит, то возникнет необходимость генерации случайных чисел в другом диапазоне. Например, для диапазона $1 \dots 10$ $r_i^* = \text{mod}(r_i, 8) + \text{mod}(r_i, 4)$.

На основании изложенного аппаратно-программная реализация настройки весовых коэффициентов нейронной сети с помощью генетического алгоритма заключается в следующем.

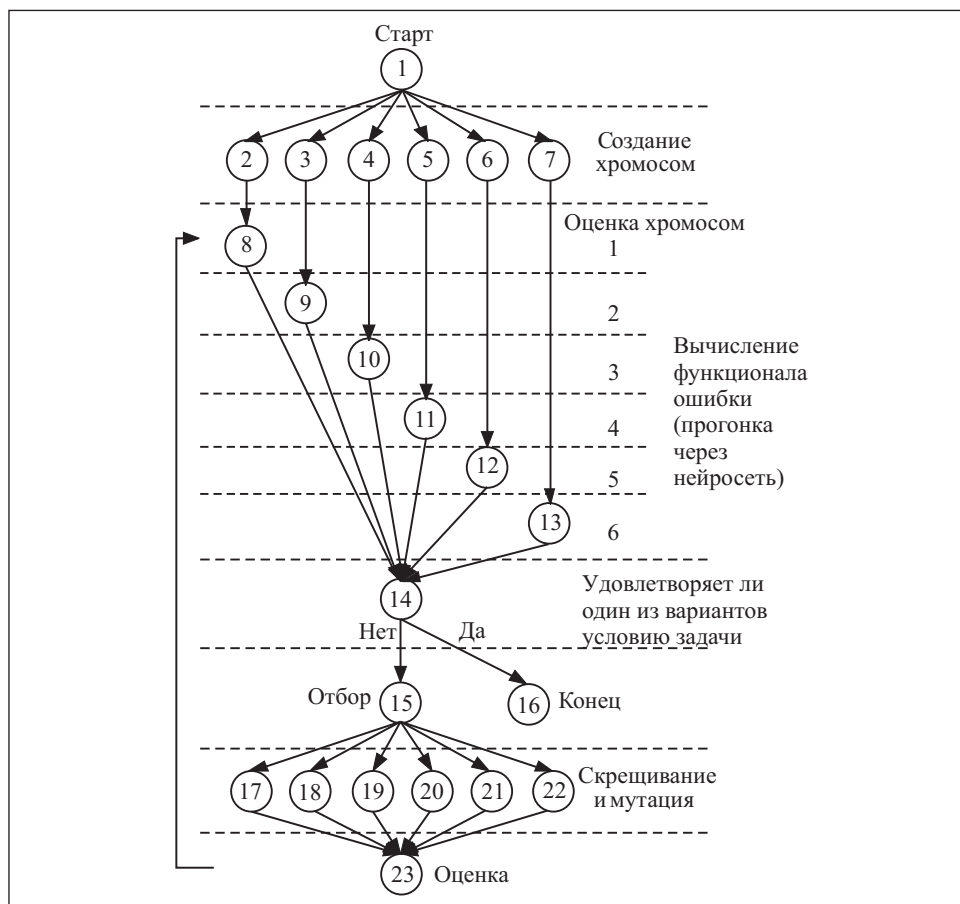


Рис. 2. Граф вычислений

Шаг 1. Строим нейросеть согласно методике, описанной в работе [12], задаем начальные параметры нейросети и кодируем их в хромосому. Создаем еще пять хромосом, аналогичных по размеру. Информация может быть представлена двумерным массивом, каждая строчка которого соответствует хромосоме и содержит информацию обо всем наборе весовых коэффициентов сети.

Шаг 2. Декодирование хромосомы в весовые коэффициенты выполняется с использованием стандартных функций языка VHDL, а именно `To_integer` и `Signed`. Значение фитнес-функции рассчитываем, как разницу между необходимым выходом сети и полученным, и определяем, насколько полученный выход сети отличается от требуемого. Подставляем поочередно значения весовых коэффициентов каждой хромосомы и вычисляем фитнес-функцию.

Шаг 3. Если одно из значений удовлетворяет условию задачи, переходим к шагу 7. Если критерий окончания поиска не удовлетворяет условию задачи, переходим к шагу 4.

Шаг 4. Решаем следующую задачу: если N — число хромосом в генетическом алгоритме, какие из них выбрать для скрещивания, чтобы в результате получить новых N хромосом, используя при этом хромосомы с лучшими генетическими качествами. Первая хромосома должна иметь лучшие генетические качества, последняя — худшие. Это достигается сортировкой по значению фитнес-функции.

До начала процесса сортировки имеем массив хромосом chr и массив значений фитнес-функций rel . Номера элементов в этих массивах соответствуют друг другу, т.е. первой хромосоме соответствует первый элемент массива rel , второй хромосоме — второй элемент массива rel и так далее. Сортировку хромосом необходимо выполнять по значению фитнес-функции. В этом случае сортируются оба массива, массив rel будет отсортирован по возрастанию значений, а массив chr сортируется для того, чтобы элементы массивов продолжали соответствовать друг другу и каждому значению фитнес-функции ставилась в соответствие хромосома, с использованием которой было рассчитано это значение. Таким образом, при завершении процесса сортировки хромосомы будут отсортированы по убыванию их уровня приспособленности: первая хромосома в массиве chr будет иметь наилучшее значение фитнес-функции, последняя — самое худшее.

Для выполнения шагов 5 и 6 следует применить операторы скрещивания и мутации хромосом, отобранных на предыдущем шаге. Для реализации операторов скрещивания и мутации рассчитаем необходимые параметры n_1 и n_2 ,

$$n_2 = \frac{1 + \sqrt{1 + 8N}}{2}, \quad n_1 = n_2 - 0,5 - \sqrt{(n_2 - 0,5)^2 - 2N},$$

по формуле

$$H' = \sum_{i=1}^{n_1} \sum_{j=i+1}^{n_2} \left(\sum_{k=1}^{b-1} H_{ik} 2^k + H_{jb} 2^b + \sum_{k=b+1}^B H_{ik} 2^k \right), \quad n_{1,2} < N, \quad (1)$$

где H' и H — новый и предыдущий наборы хромосом.

Все хромосомы рассматриваются как набор битов. Тогда первая сумма определяет первую хромосому для скрещивания (ее номер i), вторая сумма — вторую хромосому (номер j). Выражение в скобках определяет операции с битами новой хромосомы, где b и k — номера битов, 2^b , 2^k — обозначают позицию бита в двоичном числе, H_{ik} — k -й бит i -й хромосомы, H_{jb} — бит с

номером b в j -й хромосоме. Таким образом, все биты новой хромосомы копируются с i -й хромосомы предыдущего поколения, кроме бита с номером b , который копируется из j -й хромосомы.

Формула (1) является упрощенной, так как в ней хромосома рассматривается как набор битов, а от второй хромосомы наследуется только один бит. Увеличение числа бит в таком алгоритме — достаточно трудная задача. Поэтому предлагается формула, в которой учтен тот факт, что хромосома состоит из весовых коэффициентов, а каждый из них является набором битов:

$$H' = \sum_{i=1}^{n_1} \sum_{j=i+1}^{n_2} \sum_{m=1}^{m \max} \left(\sum_{k=1}^{b-1} H_{imk} 2^k + H_{jmb} 2^b + \sum_{k=b+1}^B H_{imk} 2^k \right), n_{1,2} < N, \quad (2)$$

где m — номер весового коэффициента. Данная операция реализуется с помощью циклов `for`, после чего формируется новое поколение, и переходим к шагу 2.

Шаг 7. Останавливаем работу генетического алгоритма и подставляем в нейросеть полученные значения.

Рассмотрим пример настройки параметров одного нейрона с тремя входами и соответственно тремя весовыми коэффициентами при аппаратно-программной реализации генетического алгоритма по предложенному методу. Аппаратно-программная реализация выполнена на чипе семейства Spartan 3 — XC3S200 в среде Xilinx ISE Design Suite 13.2 и промоделирована в ISE Simulator (ISim) (рис. 3).

На временной диаграмме сигналы `in1`, `in2`, `in3`, `zout` подаются на входы `in1`, `in2`, `in3` нейрона, `zout` — выход, который должен быть достигнут в результате обучения.

Сигнал `clk` формирует задержки в алгоритме обучения, а сигнал `cout` — выход нейрона на каждом шаге обучения («...» соответствует 100, так как трехзначным числам не хватает места для отображения, а 100 — единственное трехзначное число, которое можно получить на выходе).

Сигнал `finishteching` принимает значения FALSE или TRUE, значение TRUE он принимает, когда выполнен один из критериев окончания настройки нейронной сети.

Сигнал `start` формирует выход сети для каждого нового набора весовых коэффициентов и меняет свое значение при запуске этого процесса; i — номер хромосомы, на основе которой сформированы весовые коэффициенты.

Сигнал `noutsforming` находится в значении 1 в течение всего времени вычисления фитнес-функции и переходит из единицы в 0, когда все шесть хромосом обработаны и можно переходить к анализу результатов: завер-

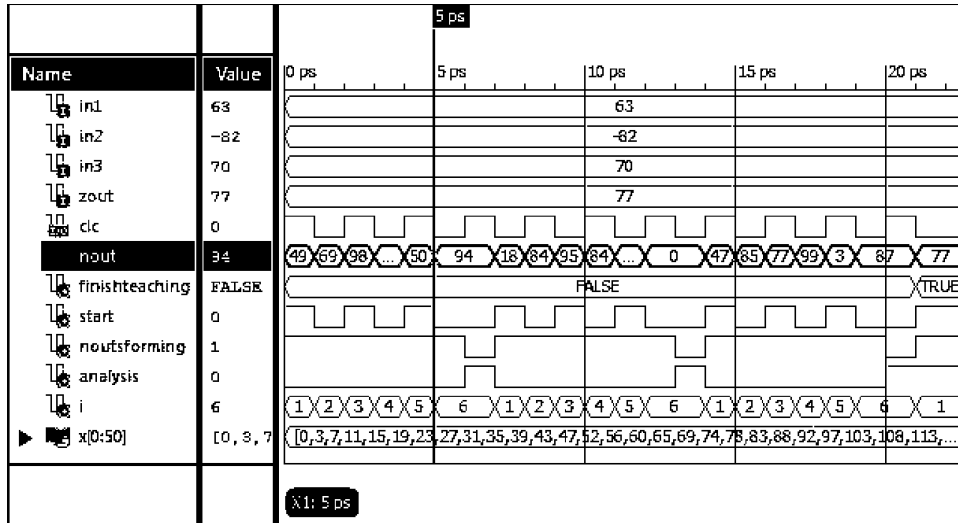


Рис. 3. Экранная форма временной диаграммы процесса настройки параметров одного нейрона

шить алгоритм или посредством скрещивания и мутации сформировать новые хромосомы.

Переход сигнала из нуля в единицу означает начало новой итерации генетического алгоритма. Решения о переходе к анализу данных, полученных в нейронной сети, и о начале новой итерации в алгоритме обучения принимаются в двух различных процессах. Поскольку язык VHDL не допускает изменения значения одного сигнала в двух различных процессах, введен дополнительный сигнал — *analysis*. После вычисления фитнес-функции шести хромосом в нейронной сети этот сигнал переходит из нуля в единицу, а в начале новой итерации генетического алгоритма — из единицы в 0. При переходе сигнала *analysis* из единицы в 0 изменяется и сигнал *noutsforming*, но в другом процессе. Сигнал *analysis* находится в единице в течение всего времени выполнения анализа информации, полученной во время пребывания хромосом в нейронной сети.

В течение первых шести пикосекунд происходит вычисление фитнес-функции хромосом в нейронной сети (*noutsforming* = 1). Затем значение сигнала *i* изменяется от единицы до шести каждую пикосекунду, в начале каждой пикосекунды изменяется значение сигнала *start* и формируется выход нейронной сети для *i*-й хромосомы (изменяется значение сигнала *nout*). На седьмой пикосекунде происходит анализ информации, полученной при вычислении фитнес-функции каждой хромосомы (*analysis* = 1).

Поскольку на этой итерации не достигнуто необходимое значение выхода (т.е. в течение шести пикосекунд (1—6) $pout \neq zout$), формируются новые хромосомы и запускается новая итерация генетического алгоритма. В течение семи пикосекунд (8—14) сигналы i , $start$, $noutsforming$, $analysis$ изменяются так же, как и в течение предыдущих семи пикосекунд, однако сигнал $pout$ уже имеет другое значение, поскольку алгоритм выполняется для других хромосом.

На третьей итерации третья хромосома обеспечивает необходимое значение выхода сети — 77. Поэтому после анализа на 21-й пикосекунде сигнал $finishteaching$ принимает значение $true$, алгоритм обучения завершается, перестают изменяться значения сигналов i , $start$, $noutsforming$, $analysis$, весовые коэффициенты сети принимают значения, соответствующие третьей хромосоме на последней итерации алгоритма, а выход сети устанавливается в необходимое значение 77.

Сигналам $in1$, $in2$, $in3$, $zout$ были специально заданы такие значения, чтобы весь процесс настройки весовых коэффициентов нейросети завершился за три итерации. В среде моделирования ISE Simulator (ISim) шаг, имеющий одну пикосекунду на каждую операцию генетического алгоритма, задан программно для отладки и наглядной демонстрации работы. Время обучения составило 36,7 наносекунды. Было выполнено моделирование процесса настройки весовых коэффициентов нейросетей с двумя и тремя нейронами, время обучения составило соответственно 37 и 38,6 наносекунд.

Выводы

При аппаратно-программной реализации генетического алгоритма распараллеливание вычислений приводит к ускорению его работы. Это позволяет строить нейросетевые системы управления, которые могут перенастраиваться и функционировать в одном темпе с процессом управления нелинейными динамическими объектами.

The authors propose a method for optimizing the weighting factors of the neural network by genetic algorithm with their implementation on Field-programmable gate array (FPGA). The examples of the hardware and software implementation of setting the weighting factors of neural network using the genetic algorithm by PLIC means are presented. It is shown that the parallelizing of calculations provides a significant acceleration of setting the weighting factors of the neural network.

СПИСОК ЛИТЕРАТУРЫ

1. Егунов Н.Д. Методы робастного, нейро-нечеткого и адаптивного управления. — М. : изд-во МГТУ им. Н.Э. Баумана, 2001. — 744 с.
2. Терехов В.А. Нейросетевые системы управления: Учеб. пособие для вузов. — М : Высш. шк., 2002. — 183 с.
3. Соловьев В. Проектирование цифровых систем на основе ПЛИС. — М. : Радио и связь, 2003. — 376 с.
4. Гильгурт С.Я. Анализ применения реконфигурируемых вычислителей на базе ПЛИС для реализации нейронных сетей // Моделювання та інформаційні технології. Зб. наук. пр. ІПМЕ НАН України. — Київ : 2006. — Вип. 37. — С. 168—174.
5. Fletcher R., Reeves C.M. Function minimization by conjugate gradients // Computer Journal. — 1964. — Vol. 7. — P. 149—157.
6. Головка В.А. Нейронные сети: обучение, организация и применение. — М. : ИПРЖР, 2001.
7. Горбань А.Н. Обучение нейронных сетей. — М. : СП «ParaGraph», 1990. — 160 с.
8. *The practical handbook of genetic algorithms. Applications* / Ed. L.D. Chambers. — Florida: CRC Press, 2000. — Vol. I. — 520 p.
9. Субботін С.О., Олійник А.О., Олійник О.О. Неітеративні, еволюційні та мультиагентні методи синтезу нечіткологічних і нейромережних моделей / Під заг. ред. С.О. Субботіна. — Запоріжжя: ЗНТУ, 2009. — 375 с.
10. Кнут Д.Э. Искусство программирования. Т. 2. Получисленные алгоритмы. — М.: Вильямс, 2000. — 832 с.
11. Бараиш Л. Алгоритм AKS проверки чисел на простоту и поиск констант генераторов псевдослучайных чисел // Безопасность информационных технологий. — 2005. — № 2. — С. 27—38.
12. Кравец П.І., Шимкович В.М., Зубенко Г.А. Технологія апаратно-програмної реалізації штучного нейрона та штучних нейронних мереж засобами FPGA / Вісн. НТУУ «КПІ». Інформатика, управління та обчислювальна техніка: Зб. наук. пр. — Київ, 2012. — № 55. — С. 174—180.

Поступила 02.04.13;
после доработки 15.05.13

КРАВЕЦ Петр Иванович, канд. техн. наук, доцент кафедры автоматики и управления в технических системах Национального технического университета «Киевский политехнический ин-т». В 1972 г. окончил Киевский политехнический ин-т. Область научных исследований — автоматизированные системы управления технологическими процессами.

ШИМКОВИЧ Владимир Николаевич, аспирант, ассистент кафедры автоматики и управления в технических системах Национального технического университета «Киевский политехнический ин-т», который окончил в 2010 г. Область научных исследований — автоматизированные системы управления технологическими процессами.