

**ИНТЕЛЕКТУАЛЬНА ЕВОЛЮЦІЙНА СИСТЕМА ПІДТРИМКИ  
ПРИЙНЯТТЯ РІШЕНЬ НА ОСНОВІ НЕЙРОМЕРЕЖЕВОГО  
І ІМУННОГО ПІДХОДІВ**

**ИНТЕЛЛЕКТУАЛЬНАЯ ЭВОЛЮЦИОННАЯ СИСТЕМА  
ПОДДЕРЖКИ ПРИНЯТИЯ РЕШЕНИЙ  
НА ОСНОВЕ НЕЙРОСЕТЕВОГО И ИММУННОГО ПОДХОДОВ**

**INTELLECTUAL EVOLUTION SYSTEM OF SUPPORT  
OF DECISION-MAKING BASED ON NEURAL NETWORK  
AND IMMUNE APPROACHES**

**Н.М. КОРАБЛЕВ**, докт.техн.наук,  
**Д.Н. СОЛОВЬЕВ, Р.Р. МАЛЮКОВ**

*Харківський національний університет радіоелектроніки, Україна*

*У статті розглядається модель інтелектуальної системи підтримки прийняття рішень на основі нейронної мережі, навчання та еволюція якої здійснюються з використанням імунного підходу. Еволюція системи розглядається як задача її адаптації до умов зміни зовнішнього середовища і властивостей об'єкта прийняття рішень, що складається з процедур корекції числа нейронів в прихованих шарах і параметрів моделі системи, які використовують імунні моделі клонального відбору та імунної мережі.*

**Ключові слова:** прийняття рішень, нейронна мережа, штучна імунна система, афінність, клональний відбір, імунна мережа.

*В статье рассматривается модель интеллектуальной системы поддержки принятия решений на основе нейронной сети, обучение и эволюция которой осуществляются с использованием иммунного подхода. Эволюция системы рассматривается как задача ее адаптации к условиям изменения внешней среды и свойств объекта принятия решений, состоящая из процедур коррекции числа нейронов в скрытых слоях и параметров модели системы, использующих иммунные модели клонального отбора и иммунной сети.*

**Ключевые слова:** принятие решений, нейронная сеть, искусственная иммунная система, аффинность, клональный отбор, иммунная сеть.

*The article considers the model of the intellectual decision support system based on the neural network, the training and evolution of which are carried out using the immune approach.*

© Кораблев Н.М., Соловьев Д.Н., Малюков Р.Р., 2017

*The evolution of the system is considered as the task of adapting it to the conditions for changing the external environment and the properties of the decision-making object, consisting of procedures for correcting the number of neurons in the hidden layers and the parameters of the system model using immune models for clonal selection and the immune network.*

**Keywords:** *decision-making, neural network, artificial immune system, affinity, clonal selection, immune network.*

**Введение.** В настоящее время стремительно увеличиваются потоки перерабатываемой информации, что требует не только автоматизации процессов обработки и анализа данных, но и интеллектуализации информационных и организационных процессов, а также использования эффективных методов и технологий принятия решений и управления. Актуальной задачей является разработка информационных технологий принятия решений, учитывающих современные требования, которые предъявляются или могут возникнуть при управлении сложными системами. Появление компьютерных систем и сетей высокой производительности с большой емкостью памяти, а также появление многоядерных процессоров, допускающих реализацию параллельных вычислений, с одной стороны, и необходимость обработки значительных массивов информации для принятия решений – с другой, привели к созданию интеллектуальных систем поддержки принятия решений ИСППР [1; 2]. Реальная ситуация требует других подходов к созданию ИСППР в условиях изменения окружающей среды и характеристик состояния системы, учесть которые априори практически невозможно.

Перспективным направлением исследований в этой области является создание гибридных ИСППР, использующих преимущества каждой из технологий искусственного интеллекта.

Существуют различные подходы к созданию ИСППР, использующих аппарат теории нечетких множеств и нечеткой логики, искусственные нейронные сети (ИНС), генетические алгоритмы, эволюционное программирование, искусственные иммунные системы (ИИС) и др.

Эти системы предназначены для помощи лицу, принимающему решение (ЛПР), при управлении сложными объектами и процессами различной природы в условиях жестких временных ограничений и наличия различного рода неопределенностей.

Они относятся к классу интегрированных интеллектуальных систем, сочетающих строгие математические модели и методы поиска решения с нестрогими методами и моделями, базирующимися на знаниях экспертов и накопленном опыте.

Особое место среди ИСППР занимают системы на основе ИНС, которые широко применяются там, где классические системы неприменимы, а именно, в случае нелинейности системы, в условиях неопределенности и др.

Ключевым концептуальным положением создания современных ИСППР является придание им способности эволюционировать в условиях изменения внешней среды, свойств объекта принятия решений и управления и др. [1; 3].

Поэтому для решения проблемы повышения качества интеллектуального анализа и обработки информации при наличии априорной и текущей неопределенности и повышения эффективности использования ИСППР при решении сложных интеллектуальных задач необходимо использовать модели системы, которые будут эволюционировать во времени для адаптации к среде функционирования и свойств объекта принятия решений (ОПР), повышая тем самым уровень своего «интеллекта».

В [3] рассмотрены эволюционные методы формирования нечетких баз правил и их редукции на основе генетического алгоритма, а в [4] исследованы нейросетевые и гибридные алгоритмы обучения для повышения эффективности ИСППР в сложных средах.

В [5] предлагается модель ИСППР на основе нечеткой логики, эволюция которой осуществляется с использованием иммунного подхода, что позволяет автоматически определять и корректировать как структуру (число правил вывода), так и параметры модели системы в зависимости от изменения свойств окружающей среды и исследуемого объекта.

В данной работе предлагается модель ИСППР на основе ИНС, эволюция которой осуществляется с использованием иммунного подхода.

Это позволяет автоматически определять и корректировать как структуру (число нейронов в скрытых слоях), так и параметры модели системы в зависимости от изменения характеристик окружающей среды и свойств исследуемого объекта.

**Постановка задачи.** Пусть  $X$  – множество выборок признаков (факторов) описания объекта,  $Y$  – множество состояний объекта (классов), а функционирование ИСППР описывается как отображение пространства факторов на пространство состояний в виде соотношения

$$F: X \rightarrow Y, X \subset R^m, Y \subset R^n. \quad (1)$$

Для принятия решения система должна иметь аналитическую зависимость соотношения (1), согласно которой по известным значениям вектора входных факторов (словарь признаков)  $X^m = \{x_1, x_2, \dots, x_m\} \subset X$  можно оценить значения выходного вектора состояний (классов)  $Y^n = \{y_1, y_2, \dots, y_n\} \subset Y$ .

Для описания связи между множествами входных и выходных переменных ОПР используется модель СППР в виде многослойной нейронной сети (НС) с прямым распространением сигнала.

В работе использован трехслойный персептрон, имеющий входной слой, содержащий число нейронов, равное числу входных признаков, один промежуточный слой и выходной слой, содержащий число нейро-

нов, равное числу классов принимаемых решений. В промежуточном слое нейронов в качестве функций активации используются сигмоидальные функции

$$z_m = f(u_m) = \frac{1}{1 + e^{-\lambda_m u_m}}, \quad u_m = \sum_{n=1}^N w_{n,m} x_n + w_{o,m}, \quad (2)$$

где  $z_m$ ,  $m = \overline{1, M}$  – выходной сигнал  $m$ -го нейрона промежуточного слоя, состоящего из  $M$  нейронов, имеющих  $N$  входов;

$x_n$ ,  $n = \overline{1, N}$  –  $n$ -я компонента входного вектора признаков, подаваемого на входной слой НС;

$w_{n,m}$  – весовой коэффициент  $n$ -го входного признака  $x_n$ , поступающего на вход  $m$ -го нейрона промежуточного слоя;  $w_{o,m}$  – значение смещения;

$\lambda_m$  – коэффициент, определяющий крутизну функции активации  $f(u_m)$ .

Нейроны выходного слоя имеют пороговую функцию активации  $\varphi$  и используется для вынесения вердикта о принадлежности анализируемой системы к определенному классу решений или нет

$$y_k = \varphi\left(\sum_{m=1}^M v_{m,k} \cdot z_m + v_{o,k}\right) = \begin{cases} 1, & \text{если } y_k > 0, \\ 0, & \text{если } y_k \leq 0, \end{cases} \quad k = \overline{1, K}, \quad (3)$$

где  $v_{m,k}$  – весовые коэффициенты;

$v_{o,k}$  – смещение;

$K$  – число выходов НС.

Необходимо решить задачи обучения НС и эволюции модели СППР в условиях изменения характеристик внешней среды и свойств ОПР.

**Изложение основного материала.** Основой предлагаемой ИС-ППР, структурная схема которой приведена на рис. 1, является база знаний, в которой могут храниться общие знания, необходимые для решения всех задач принятия решений, знания о всех внутренних связях самой системы, все прикладные знания, например, описание предметных областей, правила и ограничения на процесс принятия решений, комп-лексы алгоритмов и др.

База знаний формируется на основе информации из обучающей выборки.

Система должна поддерживать непрерывную и двустороннюю связь с ОПР. Оперативные данные об ОПР, а также данные, характеризующие необходимое воздействие на него со стороны системы, образуют рабочую память системы или ее базу данных.

Важным звеном ИСППР является блок вычисления, который в диалоге с ЛПР, обеспечивает идентификацию ситуации, логический вывод решений, объяснение и обоснование этих решений, взаимодействие с базами данных и знаний, а также использует традиционные средства моделирования, алгоритмизации и программирования.

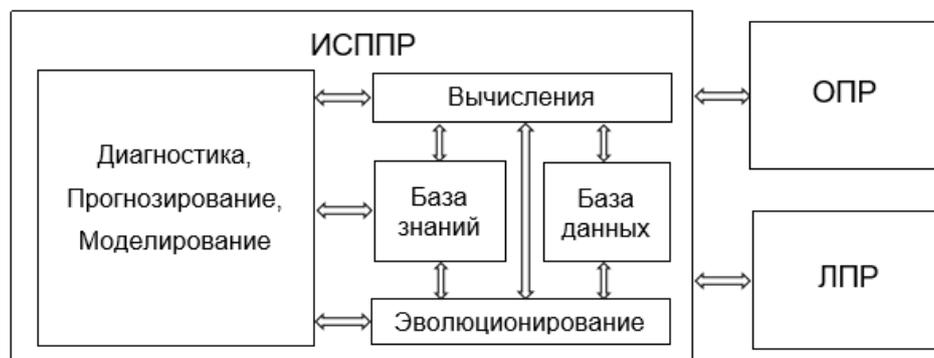


Рис. 1. Структурная схема ИСППР

Для придания системе способности развиваться во времени для адаптации к среде функционирования в состав ИСППР введен блок эволюционирования, который позволяет корректировать как структуру, так и параметры нейронной модели системы в зависимости от изменения характеристик окружающей среды и свойств исследуемого объекта.

Важной функцией системы является также выполнение стандартных функций по технической диагностике, прогнозированию, моделированию, документированию и отображению информации.

Благодаря непрерывной связи между системой и ОПР осуществляется мониторинг его параметров и как можно более раннее обнаружение неблагоприятных тенденций и отклонений в его состоянии.

Соответствующие информационно-аналитические компоненты системы осуществляют сбор, хранение и обработку оперативной информации о состоянии ОПР и происходящих в нем процессах. Она необходима для принятия оперативных решений, при отклонении текущих значений контролируемых параметров от установленных или рабочих значений.

Так как НС для принятия решений представляется в виде многослойной структуры с прямым распространением сигнала, а значение выходной переменной можно изменять, корректируя параметры элементов слоев, то для ее обучения могут быть использованы различные методы, используемые для обучения классических НС [6].

Следует отметить, что не существует универсального алгоритма обучения, подходящего для всех архитектур НС. Существует лишь набор средств, представленный множеством алгоритмов обучения, каждый из которых имеет свои достоинства и недостатки. Алгоритмы отличаются друг от друга способом настройки синоптических весов нейронов.

Существующие методы обучения НС являются трудоемкими, а также выдвигают значительные математические требования к видам целевых функций и ограничений.

Основной их чертой является отсутствие возможности модификации структуры (количества нейронов в промежуточном слое) и параметров НС.

Для устранения недостатков существующих методов обучения НС предлагается использование ИИС. Задача обучения НС рассматривается как требование минимизации целевой функции

$$\sqrt{\frac{1}{N} \sum_{n=1}^N [y(x_n, P) - y_r]^2} \Rightarrow \min_P, \quad (4)$$

где  $y_r$  – требуемое значение выхода НС;

$P$  – вектор параметров НС, в качестве которых используются параметры  $w_{n,m}$ ,  $w_{o,m}$ ,  $v_{m,k}$ ,  $v_{o,k}$  и  $\lambda_m$ ,  $n = \overline{1, N}$ ,  $m = \overline{1, M}$ ;

$y(x_n, P)$  – выходное значение НС при значении входов  $x_n$ , зависящее от параметров  $P$ .

Будем решать задачу обучения всех параметров НС в режиме off-line с использованием ИИС [7], основной идеей которой является представление решаемой задачи в виде антигена, а возможные ее решения – в виде антител. Сформируем популяцию антигенов вида

$$Ag = \{Ag_1, Ag_2, \dots, Ag_S\}, \quad (5)$$

где  $S$  – размер популяции антигенов, соответствующий количеству примеров в обучающей выборке.

Каждый элемент множества  $Ag$  – пример из обучающей выборки, представленный в виде вектора фиксированной длины

$$Ag_i = \langle x_1^i, x_2^i, \dots, x_n^i, y^i \rangle, \quad i = \overline{1, S}, \quad (6)$$

где  $x_1^i, x_2^i, \dots, x_n^i$  – входные переменные, а  $y^i$  – выходная переменная моделируемого объекта для  $i$ -го примера обучающей выборки.

Таким образом, в виде популяции антигенов  $Ag$  выступает обучающая выборка размерности  $S$  относительно входных  $x_n$ ,  $n = \overline{1, N}$  и выходной  $y$  переменных.

В качестве антител используются векторы настраиваемых параметров. В одном антителе кодируются все настраиваемые параметры НС:  $w_{n,m}$ ,  $w_{o,m}$ ,  $v_{m,k}$ ,  $v_{o,k}$  и  $\lambda_m$ ,  $n = \overline{1, N}$ ,  $m = \overline{1, M}$ . Используется вещественное кодирование антител, при котором каждый параметр вектора антитела описывается отдельным действительным числом.

Для решения задачи используется модель кодирования настраиваемых параметров в виде адаптивного структурированного мультиантитела [8], состоящего из двух частей, каждая из которых может обрабатываться независимо друг от друга (рис. 2).

$w_{1,1}, \dots, w_{1,M}; \dots, w_{N,1}, \dots, w_{N,M}; w_{0,1}, \dots, w_{0,M}; \lambda_1, \dots, \lambda_M$	$v_{1,1}$	...	$v_{M,K}$	$v_{0,M}$
$ab_0$	$ab_1$	...	$ab_M$	$ab_{M+1}$
Часть 1	Часть 2			

Рис. 2. Структура мультиантитела  $mAb$

Популяцию мультиантител представим следующим образом:

$$mAb = \{mAb_1, mAb_2, \dots, mAb_N\}, \quad (7)$$

где  $mAb_i = \{ab_0, ab_1, ab_2, \dots, ab_{L-1}\}$ ,  $i = \overline{1, N}$  –  $i$ -е адаптивное мультиантитело, представляющее собой структурированный вектор, длина которого изменяется в процессе выполнения иммунного алгоритма;

$L$  – длина вектора  $mAb_i$ ;

$N$  – размер популяции мультиантител.

Каждое мультиантитело  $mAb_i$ ,  $i = \overline{1, N}$  популяции характеризуется полным множеством настраиваемых параметров НС.

В части 1 мультиантитела закодированы весовые коэффициенты  $w_{n,m}$ , значения смещений  $w_{o,m}$  и коэффициенты  $\lambda_m$ ; в части 2 закодированы коэффициенты  $v_{m,k}$  и смещения  $v_{o,k}$ . Вторая часть мультиантитела является адаптивной, т.к. в процессе обучения помимо оптимизации коэффициентов, содержащихся в этой части, изменяется и их общее количество, т.е. выполняется структурный синтез.

Структурный синтез, в свою очередь, приводит к изменению количества нейронов в скрытом слое нейронной сети.

Структурированный способ формирования мультиантитела позволяет повысить эффективность иммунного алгоритма за счет раздельного применения иммунных операторов к каждой из частей антитела. Вычисление аффинности выполняется для мультиантитела в целом, оперируя обеими составляющими.

Используется вещественное кодирование параметров антител и антигенов, что позволяет повысить точность решений и уменьшить вычислительную сложность. Размер популяции мультиантител выбирается небольшим, т.к. использование большого количества мультиантител в популяции ведет к большим вычислительным затратам.

В качестве вычислительной модели ИИС используются принципы клонального отбора и сетевого взаимодействия [7].

Согласно принципу клонального отбора антитело, распознавшее антиген, клонируется, и полученные клоны подвергаются мутации. Если в результате мутации аффинность клона улучшается, то соответствующее антитело заменяется своим клоном, т.е. осуществляется клональный отбор. В соответствии с принципом сетевого взаимодействия антитела взаимодействуют не только с антигенами, но и с другими антителами, в результате чего обеспечивается эффект взаимной стимуляции и суппресии, что приводит к сокращению числа нейронов в промежуточном слое НС.

Формально иммунный алгоритм обучения НС можно представить следующим образом:

$$\text{ImmAlg} = (P^L, L, mAb, N, Ag, M, Op, n_c, N_c, d, \delta_{net}, Alph, A, Aff, gen, t), \quad (8)$$

где  $P^L$  – пространство поиска;

$L$  – размерность пространства поиска;

$mAb$  – популяция мультиантител вида  $mAb = \{mAb_1, mAb_2, \dots, mAb_N\}$ ;

$mAb_i$  –  $i$ -е мультиантитело популяции  $mAb$ :  $mAb_i = \{ab_0, ab_1, ab_2, \dots, ab_{L-1}\}$ ;

$N$  – размер популяции мультиантител;

$Ag$  – популяция антигенов вида  $Ag = \{Ag_1, Ag_2, \dots, Ag_M\}$ ;

$M$  – размер обучающей выборки;

$Op$  – множество используемых иммунных операторов вида

$Op = \{Clone, Mutate, Edit, Suppress\}$ ;

$n_c$  – количество мультиантител для клонирования;

$N_c$  – количество клонов одного мультиантитела;

$d$  – количество мультиантител с худшей аффинностью, подлежащие замене при редактировании популяции антител;

$\delta_{net}$  – коэффициент сжатия сети;

$Alph$  – алфавит, с использованием которого кодируются антитела;

$A$  – мощность алфавита  $Alph$ ;

$Aff$  – функция аффинности;

$gen$  – поколение работы иммунного алгоритма;

$t$  – критерий окончания работы алгоритма.

Алгоритм обучения НС представляет собой итерационную процедуру последовательной идентификации наблюдений из обучающей выборки. Антигены представляют собой примеры обучающей выборки.

Каждое антитело кодирует одно из возможных решений, а количество антител в популяции соответствует количеству примеров в обучающей выборке.

Алгоритм обучения НС представляет собой следующую последовательность шагов:

1. Инициализация популяции мультиантител  $mAb$ . Инициализация начальной популяции мультиантител выполняется случайным образом.

2. Вычисление аффинности  $Aff_{mAb-Ag}$  для каждого мультиантитела. Для вычисления аффинности мультиантитела необходимо подставить параметры, закодированные в мультиантителе, в НС. На вход НС подаются входные признаки  $x_n$  и вычисляются значения выходных переменных  $y_s$ . Аффинность мультиантитела  $mAb$  с антигеном  $Ag$  вычисляется в виде

$$Aff_{mAb-Ag} = (1 + d_{mAb-Ag})^{-1}, \quad (9)$$

где  $d_{mAb-Ag}$  – расстояние Хэмминга между полученным значением выхода НС  $y_s, s = \overline{1, S}$  и желаемым  $y$  для всех  $S$  антигенов популяции  $Ag$

$$d_{mAb-Ag} = \sum_{s=1}^S y_s, \text{ где } y_s = \begin{cases} 1, & \text{если } y_s \neq y, \\ 0, & \text{если } y_s = y. \end{cases} \quad (10)$$

3. Клонирование мультиантител пропорционально их аффинности и формирование популяции клонов  $Cl$ . Параметрами оператора клонирования являются количество антител для клонирования  $n$  и кратность клонирования антитела  $N_{Cl}$ . В иммунном алгоритме обучения НС будем использовать фиксированное значение параметра  $n$ . Кратность клонирования мультиантитела  $N_{Cl}$  будем регулировать в процессе работы иммунного алгоритма в зависимости от аффинности мультиантитела по соотношению

$$N_{Cl} = \alpha * N_{Cl\_min} + (1 - \alpha) * N_{Cl\_max}, \quad (11)$$

где  $\alpha = \frac{Aff_{best} - Aff_{mAb-Ag}}{Aff_{best}}$ ;  $N_{Cl\_min}$  и  $N_{Cl\_max}$  – минимальная и максимальная

кратность клонирования мультиантитела;

$Aff_{best}$  – лучшее значение аффинности в текущем поколении.

4. Мутация клонов обратно пропорционально аффинности мультиантитела и формирование популяции мутированных клонов  $MC$ . Мутация выбранных параметров мультиантитела  $mAb$  выполняется путем добавления гауссовского шума

$$mAb_{i+1} = mAb_i + N(0, \sigma_i), \quad (12)$$

где  $\sigma_i$  – дисперсия случайной величины, которая ассоциируется с каждым параметром мультиантитела.

Для изменения  $\sigma_i$  используется соотношение

$$\sigma_{i+1} = \sigma_i \frac{Aff_{best}^r - Aff_{mAb-Ag}}{Aff_{best}^r - Aff_{worst}^r}, \quad (13)$$

где  $Aff_{worst}^r$  – худшее значение аффинности в текущем поколении.

5. Вычисление аффинности популяции мутированных клонов  $MC$  в соответствии с (9).

Если в результате мутации аффинность улучшилась, замена соответствующих мультиантител в популяции  $mAb$ .

6. Вычисление аффинности антител внутри части 2 мультиантитела. Супрессия антител, аффинность которых больше заданного порога  $\delta_{net}$ .

Вычисление аффинности выполняется в соответствии с выражением

$$Aff_{ab_i-ab} = (1 + d_{ab_i-ab})^{-1}, \quad (14)$$

где  $d_{ab_i-ab}$  – расстояние между  $i$ -м антителом и остальными антителами части 2 мультиантитела

$$d_{ab_i-ab} = \|ab_i - ab_j\| = \sqrt{\sum_{j=1}^M (v_i - v_j)^2}, i = \overline{1, M}. \quad (15)$$

Выполнение супрессии путем удаления антител  $ab_i$  с аффинностью, большей заданного порога  $\delta_{net}$ , позволяет уменьшить количество нейронов в скрытом слое и устранить, таким образом, их избыточность.

7. Проверка критерия останова. В качестве критерия останова используется либо достижение заданного порога аффинности, либо достижение заданного количества поколений работы алгоритма.

8. Результатом работы алгоритма будет мультиантитело с лучшей по популяции аффинностью, определяющее структуру НС и содержащее настроенные параметры НС.

Шаги 1-5 алгоритма соответствуют принципу клонального отбора. Алгоритм на данных этапах работает с обеими частями мультиантитела. Шаг 6 соответствует принципу сетевого взаимодействия.

Если ранее мультиантитело обрабатывалось как обычное антитело, то на данном шаге работа выполняется только с частью 2 мультиантитела, которая состоит из отдельных антител, представляющих собой параметры  $v_{m,k}$ ,  $m = \overline{0, M}$ ,  $k = \overline{1, K}$ .

В процессе функционирования ИСППР возможно изменение характеристик окружающей среды и свойств исследуемого объекта.

В этом случае необходимо адаптировать систему под эти изменения, т.е. осуществлять ее эволюцию.

Для эволюции модели ИСППР предлагается использовать предложенную процедуру обучения НС на основе ИИС, но в качестве обучающей выборки на вход будут поступать реальные значения входных факторов системы, функционирующей в реальном режиме времени.

Это позволит корректировать в реальном времени структуру (число нейронов в скрытых слоях) и параметры ИСППР.

**Экспериментальные исследования.** В качестве тестовой рассматривалась задача проектирования и анализа локальной компьютерной сети (ЛКС), которая является примером трудно формализуемой задачи, для решения которой необходим интегрированный подход, позволяющий рационально сочетать математические и логические методы исследования сложных систем, интуицию и опыт ЛПР с использованием новых методов представления знаний, прогноза и оценки альтернативных вариантов решений.

При проектировании ЛКС существует ряд важных вопросов, которые подлежат рассмотрению перед реализацией данной задачи.

К таким вопросам относятся выходные параметры, которым должна удовлетворять ЛКС, и начальные условия (входные параметры), которые задаются непосредственно перед процессом проектирования.

Анализ проектирования ЛКС позволил выделить начальные условия, наиболее существенные из которых следующие:  $x_1$  – стоимость компьютерной сети;  $x_2$  – количество и расположение пользователей;  $x_3$  – простота установки и изменения конфигурации сети;  $x_4$  – пропускная способность сети;  $x_5$  – надежность сети;  $x_6$  – безопасность сети;  $x_7$  – возможность расширения сети.

Исходя из требований проектирования ЛКС, имеем следующие основные выходные параметры: 1) сетевая операционная система (ОС); 2) топология сети; 3) сетевая технология.

Варианты выбора ОС: 1) Novell NetWare; 2) Microsoft Windows (NT, Vista, 7, 8, 10); 3) UNIX системы (Solaris, FreeBSD); 4) GNU/Linux системы; 5) IOS; 6) ZyNOS компании ZyXEL.

Варианты выбора топологии сети: 1) звезда; 2) шина; 3) кольцо; 4) дерево; 5) полносвязная; 6) ячеистая; 7) смешанная.

Варианты выбора сетевой технологии: 1) Ethernet; 2) Token Ring; 3) FDDI.

Реализация данной задачи методом простого перебора вариантов неприменима, поскольку комбинаторная емкость перебора составляет несколько порядков.

Более того, большая часть комбинаций вариантов никогда не будет реализована из-за высоких материальных затрат.

Поэтому реализация была выполнена с применением эволюционирующих нейронных сетей.

Задача условно была разбита на три параллельные задачи, каждая из которых решает проблему по одному из трех выходных параметров. При этом каждый из выходных параметров может определяться значениями не всех 7 входных, а только нескольких из них.

Для каждой из трех задач была создана своя нейронная сеть, содержащая определенное число входов и выходов, например, для выбора ОС – это 7 и 6 соответственно.

Количество скрытых слоев – один, содержащий 15 и более нейронов. Структура спроектированной НС для определения ОС приведена на рис. 2.

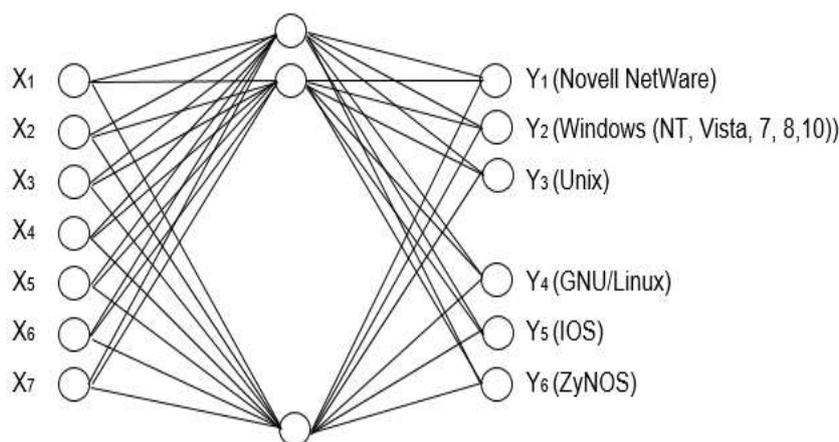


Рис. 2. Структура нейронной сети для определения ОС

Для каждой из сетей созданы обучающие выборки, содержащие не менее 50 вариантов. Начальные значения всех весов нейронной сети определялись случайным образом в диапазоне от -0,5 до +0,5.

На выходе каждый вариант обучающей выборки имеет комбинацию из нулей и одной единицы.

Так, например, если производится выбор ОС, то НС имеет 6 нейронов на выходе, а вариант «0001000» соответствует решению «GNU/Linux системы».

После создания нейронных сетей каждая из них обучается с помощью ИИС согласно своей обучающей выборке, достигая заданного предела обучения.

Число эпох обучения сети зависит не только от значения предела, но и от количества вариантов в выборке.

При проверке, достигла ли сеть предела обучения, результат, полученный на некотором шаге, дает ошибку по каждому из вариантов выборки, поэтому при увеличении числа вариантов сеть учится дольше.

После окончания процесса обучения подача на вход варианта из набора входных переменных формирует результат из трех параметров.

Таким образом, проведенные исследования показали, что многослойные модульные нейронные сети персептронного типа с иммунным обучением целесообразно использовать для прогнозирования возможных вариантов решений при выборе параметров ЛКС, при этом наблюдается высокая устойчивость получаемых результатов.

**Выводы.** В работе рассмотрено решение актуальной задачи повышения качества функционирования ИСППР на основе использования нейронных сетей и искусственных иммунных систем для формирования модели принятия решений и эволюции ее структуры и параметров в условиях изменения внешней среды и свойств объекта принятия решений.

Эффективность решения этой задачи определяется выбором модели принятия решений в виде многослойной нейронной сети, а также разработкой метода ее обучения и эволюции на основе иммунного подхода.

Эволюция модели ИСППР рассматривается как задача ее адаптации к условиям изменения внешней среды и свойств объекта принятия решений, состоящая из процедур коррекции структуры (числа нейронов в скрытых слоях) и параметров модели системы, использующих иммунные модели клонального отбора и иммунной сети.

Осуществлена программная реализация и проведены экспериментальные исследования предложенного подхода на примере проектирования локальной компьютерной сети, которые подтвердили эффективность предложенной эволюционной модели ИСППР, использующей нейросетевой и иммунный подходы.

## ЛИТЕРАТУРА

1. Снитюк В.Е. *Эволюционные технологии принятия решений в условиях неопределенности* / В.Е. Снитюк. – К.: МП Леся, 2015. – 347 с.
2. Стадниченко С.Ю. *Интеллектуальные системы поддержки принятия решений* / С.Ю. Стадниченко // *Молодой ученый*. – 2010. – № 6. – С. 61-63.
3. Комарцова Л.Г. *Эволюционные методы формирования нечетких баз правил* / Л.Г. Комарцова // *Материалы международной конференции OSTIS-2011*. – 2011. – С. 181-184.

- 
4. Комарцова Л.Г. Исследование нейросетевых алгоритмов обучения в интеллектуальных эволюционных системах / Л.Г. Комарцова, Ю.Н. Лавренков, О.В. Антипова // *Материалы IV-й международной конференции OSTIS-2014.* – 2014. – С. 233-238.
  5. Korablev N.M. *Intelligent Evolution Decision Support System based on Fuzzy Logic and Immune Approach* / N.M. Korablev, D.N. Soloviov, R.R. Malyukov // *System technologies – Dnipro.* – 2016. – № 6 (107). – P. 98-106.
  6. Бодянский Е.В. *Искусственные нейронные сети: архитектуры, обучение, применения* / Е.В. Бодянский, О.Г. Руденко. – Харьков: ТЕЛТЕХ, 2004. – 372 с.
  7. Dasgupta D. *Recent Advanced in Artificial Immune Systems: Models and Applications* / D. Dasgupta, S. Yu, F. Nino // *Applied Soft Computing.* Elsevier. –2011. – P. 1574-1587.
  8. Korablev N. *Immune Approach for Neuro-Fuzzy Systems Learning Using Multiantibody Model* / N. Korablev, I. Sorokina // *ICARIS 2011, Springer Lecture Notes in Computer Science.* – 2011. – Vol. 6825. – P. 395-405.

Стаття надійшла до редакції 03.09.2017