

де e_{ij} – внесок кожного нечіткого правила у вихід мережі; $\bar{\varphi}(R_j)$ – відносний ступінь виконання j -го нечіткого правила; φ_i – чітке число, що задає висновок кожного i -го (отриманого) правила:

$$\varphi_i = \sum_{j=1}^n R_j(P_i), i = 1, \dots, k.$$

Кожний вузол четвертого шару з'єднаний з одним вузлом третього шару, а також із усіма входами мережі.

І нарешті, п'ятий (вихідний) шар реалізує функцію дефазифікації, в результаті якої визначається сума внесків усіх правил. В даному шарі відбувається генерація результуючого висновку на основі нечітких правил (4-го шару):

$$e = \sum_{j=1}^k e_j$$

де e – вихід мережі; $\sum_{j=1}^k e_j$ – сумарний внесок всіх нечітких правил.

Для навчання побудованої мережі використано гібридний алгоритм, в якому застосовано комбінацію градієнтного спуску у вигляді алгоритму зворотного поширення помилки і методу найменших квадратів [2].

Процес навчання відбувається у два етапи. Спочатку на вхід модуля управління подаються значення параметрів вектора P_i , що входять у навчальну вибірку. На його основі формується вихідна дія, яка управляє. Цей сигнал розповсюджується по мережі в прямому напрямку, і послідовно з використанням ітераційного методу найменших квадратів розраховуються значення вихідних сигналів проміжних шарів і вихідного сигналу e .

На другому етапі при використанні методу зворотного поширення помилки вихідна реакція $\varphi_a(R_j)$ порівнюється з еталонним значенням $\varphi_e(R_j)$, і за наслідками порівняння модифікуються значення ваг вхідного вектора P_i .

Ітераційна процедура настроювання продовжується поки відхилення перевищує заздалегідь установлене значення похибки. Навчена ANFIS із використанням описаного алгоритму забезпечує необхідну точність визначення режиму навчання та використовується для керування адаптивними навчальними траєкторіями.

Висновки. Запропонована технологія нейро-нечіткого виведення є ефективним інструментом для моделювання адаптивних траєкторій в навчальних системах, оскільки дозволяє підвищити якість навчальних систем за рахунок високої швидкодії обчислень та можливості оперативного навчання в процесі надходження нових даних.

Список посилань.

1. Штовба С. Д. Проектирование нечетких систем средствами MATLAB. – М.: Горячая линия – Телеком, 2007. – 288 с.
2. Du K. - L., Swamy M. N. S. Neural networks and statistical learning. – London: Springer-Verlag, 2014. – 824 p.

УДК 004.89:004.773

МНОГОАГЕНТНАЯ СИСТЕМА ДЛЯ ПОДДЕРЖКИ ПРИНЯТИЯ РЕШЕНИЙ

к.т.н. Абабий В.В., к.т.н. Судачевски В.М., Мунтяну С., Цуркан А., Технический Университет Республики Молдова, Кишинев, Республика Молдова

Введение. Сегодня, развитие производственных процессов, нельзя планировать без использования интеллектуальных систем поддержки принятия решений [1]. Данные системы в состоянии подсказать пользователю достаточно мудрые и разумные решения, которые без

их участия практически невозможны, из-за отсутствия объективного и полного анализа предметной области [2, 3, 4].

В настоящее время достаточно широко используются множество систем поддержки принятия решений в таких областях как: бизнес и менеджмент, инвестиции, инжиниринг, медицина, окружающая среда и др., где основными элементами являются Интеллектуальные Агенты. Наиболее эффективными считаются Много-Агентные распределенные или мобильные системы, которые способны охватывать большие территории для анализа в процессе накопления знаний и обучения [5].

Основным достоинством любого Интеллектуального Агента можно считать наличие знаний и возможность применения этих знаний для персонального обучения через процессы накопления данных, их анализ и формирование новых более прогрессивных знаний [6, 7, 8].

В данной работе предлагается проектирование Многоагентной системы для поддержки принятия решений на базе многокритериального анализа пространства состояний, формирование новых знаний и вычисление предложений для принятия решений.

Постановка задачи.

Пусть задано множество Агентов $A = \{A_m, \forall m = \overline{1, M}\}$, которая активирует в пространстве $D \subseteq R^N$, где N размерность пространства D . В пространстве D задан процесс P , который описывается вектором состояния $X = \{x_s, \forall s = \overline{1, S}\}$, где $X \subseteq D$. Идеальным соотношением процесса P и пространства D можно считать при условии, что $S = N$ и $X \approx D$.

Каждый Агент A_m преследует собственные интересы и воздействует на процесс P вектором управления (принятия решений) $U_m = \{u_{m,s}, \forall s = \overline{1, S}\}$ с целью получения максимальной прибыли $P_m(D) \rightarrow \max, \forall m = \overline{1, M}$. При выполнении условия лояльной конкуренции [9] будет достигнута и максимальная глобальная прибыль $P = \bigcup_{m=1}^M (P_m) \rightarrow \max_G$.

Так как каждый Агент является конкурентом для других Агентов, обмен данными между ними не предусмотрен. О намерениях Агентов конкурентов можно узнать только в результате анализа состояния процесса P .

Решение поставленной задачи.

Агент это интеллектуальное устройство с индивидуальными характеристиками $A_m = \{E_m, K_m, MM_m, O_m, AC_m\}$, где: E_m - способность восприятия пространства активности D (ввод вектора состояния X); K_m - специфические знания, в области деятельности Агента предназначенные для принятия решений и генерации новых знаний; O_m - цель деятельности Агента; MM_m - математические методы и модели для обработки данных и знаний с целью принятия решений и получения новых знаний; AC_m - методы и средства для воздействия на процесс P (вывод вектора U_m) для достижения поставленных целей, то есть получение максимальной прибыли $P_m(D) \rightarrow \max$.

На Рисунке 1 представлена структура Агента. Функциональность Агента заключается в генерации последовательности операций, которая задается блоком MM_m , в зависимости от содержания блока знаний $K_m(t)$ и условия сходимости для получения максимальной прибыли $P_m(D) \rightarrow \max$:

1. Ввод данных о состоянии процесса P (ввод вектора $X_m(t)$);
2. Преобразование данных в знания (блок $K_m(t)$);
3. Проверка выполнения условия $P_m(D) \rightarrow \max$:

4. Вычисление вектора $U_m(t)$ (принятия решений) и воздействие на процесс P , в результате получаем новое состояние процесса $X(t+1)$;
5. Переход на п. 1.

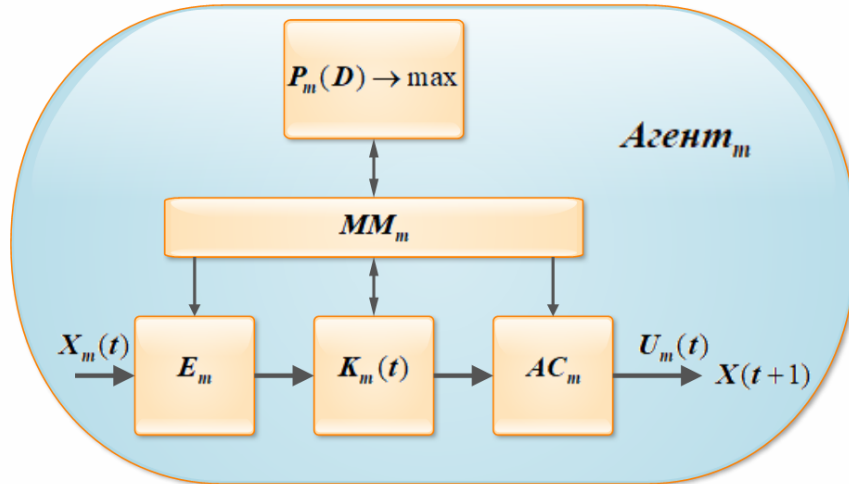


Рисунок 1 – Структура Агента

В данной работе в качестве модели для принятия решений используются нейронные сети. На Рисунке 2 представлена модель нейронной сети Агента A_m для вычисления параметров принятия решений $U_m(t)$, которая состоит из: вектора состояния процесса $X_m(t)$; матрица коэффициентов или набор знаний $W_m^2 = \{w_{m,j,i}, \forall j = \overline{1, S}, i = \overline{1, S}\}$; сумматоры Σ ; вектор результатов суммирования $Q_m = \{q_{m,j}, \forall j = \overline{1, S}\}$; функции для вычисления параметров принятия решений $f_{m,j}, \forall j = \overline{1, S}$; и вектор параметров для принятия решений $U_m(t) = \{u_{m,j}, j = \overline{1, S}\}$, где $f_{m,j} | q_{m,j} \rightarrow u_{m,j}, \forall j = \overline{1, S}$.

Для вычисления условия получения максимальной прибыли $P_m(D) \rightarrow \max$ используется вектор Q_m , вектор коэффициентов или набор знаний $W_m^1 = \{w_{m,j}, \forall j = \overline{1, S}\}$ и преобразование $f_{m,p} | g_m \rightarrow P_m(D)$.

Для моделирования и тестирования функциональности модели нейронной сети достаточно подключить выходы $X(t+1)$ с входами $X(t)$.

Выводы.

В данной работе представлен результат проектирования Многоагентной системы для поддержки принятия решений на базе многокритериального анализ пространства состояний, формирование новых знаний и вычисление предложений для принятия решений. Представлена модель и структура Агента. По условиям задачи обмен данными между Агентами не предусмотрен, каждый Агент самостоятельное и индивидуальное устройство. О действиях Агентов конкурентов можно узнать только в результате анализа пространства состояния. Модель принятия решений и вычисление условия достижения цели реализованы в виде искусственных нейронных сетей.

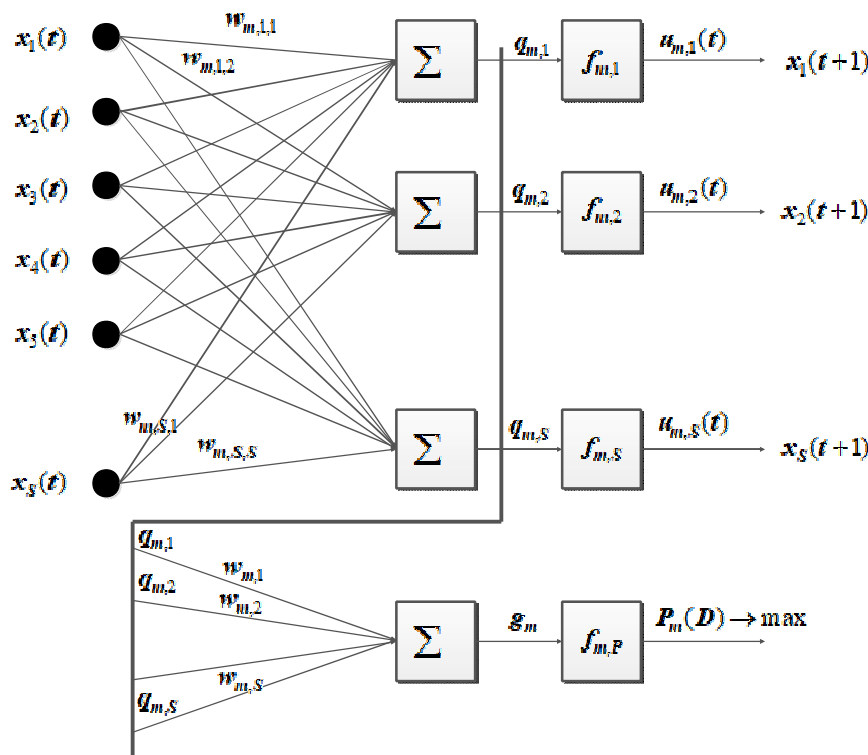


Рисунок 2 - Модель нейронной сети

Список литературы.

1. Городецкий В.И., Карсаев О.В., Самойлов В.В., Серебряков С.В., Прикладные Многоагентные системы группового управления. *Искусственный интеллект и принятие решений*, 2/2009. стр. 3-24.
2. Manolache M. A., *Integrated Decision Making Environment*. PHD Thesis. University Politehnica of Bucharest, Bucharest – 2019, 155p.
3. Coccia M., Critical Decisions in Crisis Management, *Working Paper CocciaLab n. 45/2020*, CNR -- National Research Council of Italy. 2020.
4. Ferasso M., Bergamaschi E. A., Bounded Rationality Effect on Firm's Choices on R&D Investments: A Model for Decision-Making Effectiveness Analysis. *Journal of Research in Emerging Markets*, 2020, 2(1), ISSN: 2663-905X, pp.24-42. doi: 10.30585/jrems.vi4.449.
5. Кузнецов А. В., Краткий обзор Многоагентных моделей. *Управление большими системами. Выпуск 71*, 2018, стр. 6-44.
6. Ababii V., Sudacevschi V., Melnic R., Munteanu S. Multi-Agent System for Distributed Decision-Making. *National Science Journal (Ekaterinburg, Russia)*, Vol 2, No 45, 2019, pp. 19-23, ISSN 2413-5291. DOI: 10.31618/nas.2413-5291.2019.2.45.
7. Ababii V., Sudacevschi V., Munteanu S., Bordian D., Calugari D., Nistiriuc A., Dilevschi S. Multi-Agent Cognitive System for Optimal Solution Search. *The International Conference on Development and Application Systems (DAS-2018) 14th Edition*, May 24-26, 2018, Suceava, Romania, pp. 53-56, IEEE Catalog Number: CFP1865Y-DVD, ISBN: 978-1-5386-1493-8.
8. Safonov Gh., Ababii V., Sudacevschi V. The Decision-Making System Based on Adaptive Agents. Proceedings of the 9th International Conference on Microelectronics and Computer Science & The 6th Conference of Physicists of Moldova, Chişinău, Moldova, October 19-21, 2017. pp. 275-277, ISBN: 978-9975-4264-8-0.
9. Портер Е. М. Конкурентная стратегия: Методика анализа отраслей и конкурентов. Пер. с англ. – М.: Альпина Бизнес Букс, 2005. – 454 с. ISBN: 5-6914-0143-0.