

Н. С. Гумар, В. Г. Баевич, М. О. Кострома
(ГГУ имени Ф. Скорины, Гомель)
Науч. рук. **Ю. В. Никитюк**, канд. физ.-мат. наук, доцент

НЕЧЕТКОЕ И НЕЙРОНЕЧЕТКОЕ МОДЕЛИРОВАНИЕ В MATLAB И PYTHON

Нечеткое и нейронечеткое моделирование являются основными направлениями искусственного интеллекта для работы со сложными системами, которые эффективно используются в различных областях науки и техники [1–4].

Эти направления моделирования представляют собой современные подходы к обработке информации в условиях неопределенности, нечеткости и частичной достоверности исходных данных. В отличие от традиционных методов данные подходы имитируют особенности человеческого мышления, обеспечивая возможность формирования гибких моделей сложных систем.

Целью нечеткого моделирования является построение формализованных моделей, способных учитывать экспертные знания и неполноту информации. Нейронечеткое моделирование представляет собой синтез нечеткой логики и искусственных нейронных сетей, что позволяет объединить интерпретируемость нечетких систем с обучаемостью нейросетевых структур.

Для реализации таких моделей используются различные инструменты, в числе которых наиболее популярными являются Fuzzy Logic Toolbox в среде MATLAB и библиотека scikit-fuzzy для Python. Для нейронечеткого моделирования применяются также встроенные средства MATLAB (например, anfis) и специализированные Python-фреймворки, такие как anfis, skfuzzy-neuro, а также модули на базе TensorFlow и PyTorch. Сравним перечисленные выше инструменты [5–7].

MATLAB требует наличия лицензии, в том числе и для использования Fuzzy Logic Toolbox и ANFIS. В отличие от этого, библиотеки Python, такие как scikit-fuzzy и anfis, являются открытыми и доступны бесплатно.

В MATLAB предоставляется графическая среда проектирования нечетких систем (FIS Editor), позволяющая визуализировать переменные, функции принадлежности и правила логического вывода (рисунок 1). В Python подобный функционал реализуется через код, что требует определенного уровня подготовки, но обеспечивает гибкость и автоматизацию (рисунок 2).

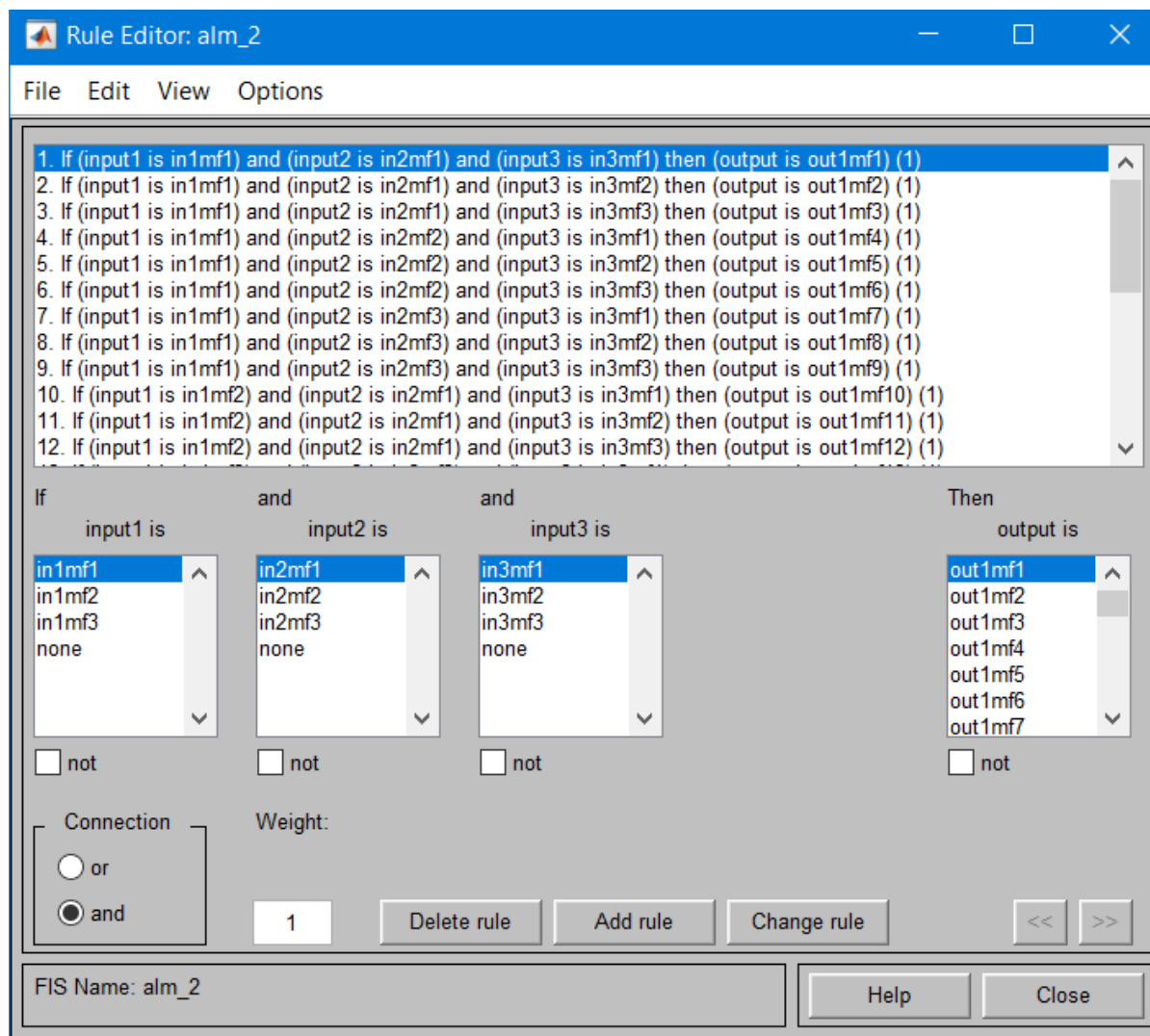


Рисунок 1 – Графический интерфейс редактора правил в MATLAB

Нейронечеткие системы в MATLAB можно обучать с помощью встроенной функции `anfis`, которая реализует адаптивную нейронечеткую систему вывода Сугено. В Python аналогичный функционал обеспечивается библиотеками `anfis`.

Инструменты на Python легко интегрируются с современными библиотеками машинного обучения (`scikit-learn`, `TensorFlow`, `PyTorch`), что делает их удобными при построении гибридных интеллектуальных систем. MATLAB же обеспечивает комплексную среду для моделирования и управления, что может быть полезно в инженерных задачах.

```

1 #rule1 = ctrl.Rule(V['M'] & P['L'] & R['M'], T_ctrl['V_L'])
2 #rule2 = ctrl.Rule(V['L'] & P['L'] & R['L'], T_ctrl['W_L'])
3 #rule3 = ctrl.Rule(V['H'] & P['L'] & R['L'], T_ctrl['W_L'])
4 #rule4 = ctrl.Rule(V['L'] & P['L'] & R['H'], T_ctrl['H'])
5 #rule5 = ctrl.Rule(V['H'] & P['L'] & R['H'], T_ctrl['H'])
6 #rule6 = ctrl.Rule(V['M'] & P['M'] & R['M'], T_ctrl['L'])
7 #rule7 = ctrl.Rule(V['L'] & P['M'] & R['M'], T_ctrl['L'])
8 #rule8 = ctrl.Rule(V['H'] & P['M'] & R['M'], T_ctrl['L'])
9 #rule9 = ctrl.Rule(V['M'] & P['M'] & R['L'], T_ctrl['W_L'])
10 #rule10 = ctrl.Rule(V['M'] & P['M'] & R['H'], T_ctrl['V_H'])
11 #rule11 = ctrl.Rule(V['M'] & P['H'] & R['M'], T_ctrl['M'])
12 #rule12 = ctrl.Rule(V['L'] & P['H'] & R['L'], T_ctrl['W_L'])
13 #rule13 = ctrl.Rule(V['H'] & P['H'] & R['L'], T_ctrl['W_L'])
14 #rule14 = ctrl.Rule(V['L'] & P['H'] & R['H'], T_ctrl['W_H'])
15 #rule15 = ctrl.Rule(V['H'] & P['H'] & R['H'], T_ctrl['W_H'])

```

Рисунок 2 – Набор правил в scikit-fuzzy

Выбор между MATLAB и Python зависит от задач и ресурсов: MATLAB подходит для быстрого прототипирования, а Python обеспечивает гибкость, масштабируемость. Оба подхода позволяют эффективно решать соответствующие задачи, сочетая прозрачность нечеткой логики с точностью машинного обучения.

Литература

1. Determination of parameters for controlled laser cleaving of silicate glasses using regression, neural network and fuzzy models / Yu. V. Nikityuk, A. F. Vasilyev, L. N. Marchenko [et al.] // Problems of Physics, Mathematics and Technics. – 2024. – No. 2(59). – P. 32–38.
2. Определение параметров двухлучевого лазерного раскалывания силикатных стекол с использованием нейро-нечетких сетей / Ю. В. Никитюк, [и др.] // Информационные технологии и технические средства управления (ICST-2024) : Материалы VIII Международной научной конференции, Владикавказ, 01–05 октября 2024 года. – М. : Институт проблем управления им. В. А. Трапезникова РАН, 2024. – С. 76–78.
3. Определение параметров лазерной обработки алмазов с применением метода конечных элементов и нечетких моделей / Ю. В. Никитюк, А. Н. Сердюков, Д. Г. Пилипцов [и др.] // Информационные технологии и технические средства управления (ICST-2024) : Материалы VIII Международной научной конференции, Владикавказ, 01–05 октября 2024 года. – М. : Институт проблем управления им. В. А. Трапезникова РАН, 2024. – С. 240–242.
4. Determination of parameters of laser cutting of diamonds using neural-fuzzy networks / Yu. V. Nikitjuk, V. A. Prokhorenko, D. G. Piliptsov, B. Zhou // Проблемы взаимодействия излучения с веществом : сборник материалов VI Международной научной конференции, посвященной академику Б. В. Бокутю (Гомель, 14 ноября 2024 г.). – Гомель : ГГУ им. Ф. Скорины, 2025. – С. 235–238.
5. Леоненков, А. В. Нечеткое моделирование в среде MATLAB и fuzzyTECH / А. В. Леоненков. – СПб. : БХВ-Петербург, 2005. – 736 с.
6. Pedregosa, F., et al. (2011). Scikit-learn: Machine Learning in Python. Journal of Machine Learning Research, 12, 2825-2830.
7. MATLAB Fuzzy Logic Toolbox Documentation. MathWorks.