Интегрирование модуля нечеткого логического вывода с помощью веб-технологий

Н.Д. Баданина[1], А.А. Зинченко[1], В.А. Судаков[1,2]

[1]МАИ, Москва, Россия
[2]ИПМ им. М.В. Келдыша РАН, Москва, Россия

Аннотация. В данной работе предлагается разработанный модуль нечетких вычислений и возможные способы его интеграции для веб-приложения. В результате применения нечетких вычислений удается улучшить качество принимаемых решений и повысить точность результатов, поэтому возникла потребность в разработке ПО, которое без лишних шагов установки было бы доступно пользователю при помощи веб-технологий.

Ключевые слова: Нечеткий логический вывод, интеграция, Jupyter notebook, Panel, REST API.

Integration of fuzzy inference module using web technologies

N.D. Badanina [1], A.A. Zinchenko [1], V.A. Sudakov[1,2]

[1] MAI, Moscow, Russia
[2] Keldysh Institute of Applied Mathematics of Russian Academy of Sciences (KIAM RAS)

Abstract. This paper proposes a developed fuzzy computing module and possible ways to integrate it for a web application. As a result of the use of fuzzy calculations, it is possible to improve the quality of decisions made and increase the accuracy of the results, so there was a need to develop software that would be available to the user using web technologies without unnecessary installation steps.

Keywords: Fuzzy inference, integration, Jupyter notebook, Panel, REST API.

Нечеткие вычисления (или теория нечетких множеств) – это математическая теория, которая позволяет работать с нечеткими данными. Основой данной теории послужило то, что при описании процессов и объектов люди используют приближенные рассуждения и оценки, которые субъективны. Нечеткий вывод учитывает субъективность экспертов по одному и тому же вопросу, поэтому может быть применен во многих областях, таких как управление, искусственный интеллект, прогнозирование, распознавание образов и медицина.

1. Нечеткая логика

Предметом нечёткой логики является исследование рассуждений в условиях нечёткости, размытости, неопределённости, сходных с рассуждениями в обычном смысле. В основе теории нечётких множеств лежит представление о том, что составляющие множество элементы, обладающие общим свойством, могут обладать этим свойством в различной степени и, следовательно, принадлежать данному множеству с различной степенью.

Пусть E – универсальное множество, X – элемент E, а R – некоторое свойство. Обычное (четкое) подмножество A универсального множества E, элементы которого удовлетворяют свойству R, определяется как множество упорядоченных пар $A = \{\mu_A(x)/x\}$, где $\mu_A(x)$ – характеристическая функция принадлежности (или просто функция принадлежности), принимающая значения в некотором вполне упорядоченном множестве M (например, $M = [0,1]$). Функция принадлежности указывает степень (или уровень) принадлежности элемента X подмножеству A.

В связи с этим нечеткое подмножество A универсального множества E определяется как множество упорядоченных пар. Функция принадлежности $\mu_A(x)$ указывает степень принадлежности элемента X подмножеству A [1].

В 1973 году Л. Заде представил понятие нечёткой логики, в которой множеством истинностных значений является счётное множество лингвистических названий значений истинности. Истинность понимается как лингвистическая переменная, значение которой представлено нечёткими множествами, т.е. словами или предложениями естественного или искусственного языка. Множество истинностных значений лингвистической переменной называется терм-множеством. Лингвистические значения истинности имеют числовые значения, которые также являются нечёткими множествами. Истинность, в отличие от классической логики, тоже является нечёткой, поскольку может принимать не только два привычных значения «истинно» и «ложно».

Понятие лингвистической переменной играет важную роль в нечётком логическом выводе и в принятии решений на основе приближённых

рассуждений. Нечёткая переменная характеризуется набором-тройкой ⟨a, X, A⟩, где:

- a – имя переменной;
- X – универсальное множество (область определения a);
- A – нечёткое множество на X, описывающее ограничение (т.е. $\mu_A(x)$) на значение нечёткой переменной a.

Лингвистическая переменная характеризуется набором-пятёркой ⟨b,T, X, G, M⟩, где:

- b – имя лингвистической переменной;
- T – множество его значений (терм-множество), представляющие имена нечётких переменных, областью определения, которых является множество X. Множество T называется базовым терм-множеством лингвистической переменной;
- G – синтаксическая процедура, позволяющая оперировать элементами терм-множества T, в частности, генерировать новые термы (значения);
- M – семантическая процедура, позволяющая преобразовать новое значение лингвистической переменной, образованной процедурой G, в нечёткую переменную, то есть сформировать соответствующее нечёткое множество [2].

Для задания функций принадлежности существуют разные типовые формы, самые распространенные из которых:

1. Треугольная.

$$MF(x) = \begin{cases} 1 - \dfrac{b-x}{b-a}, a \le x \le b \\ 1 - \dfrac{x-b}{c-b'} b \le x \le c \\ 0, \text{в остальных случаях} \end{cases} \quad (1)$$

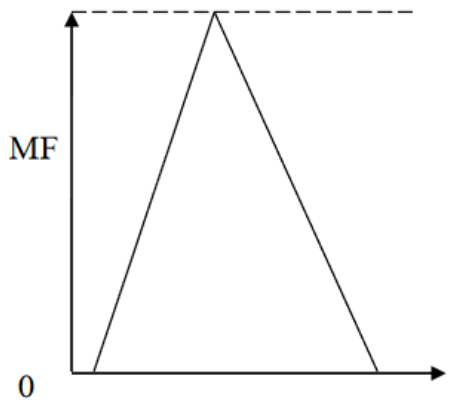

Рис. 1. Треугольная функция принадлежности

2. Трапецеидальная.

$$MF(x) = \begin{cases} 1 - \dfrac{b-x}{b-a}, a \le x \le b \\ 1, b \le x \le c \\ 1 - \dfrac{x-b}{c-b'} b \le x \le c \\ 0, \text{в остальных случаях} \end{cases} \quad (2)$$

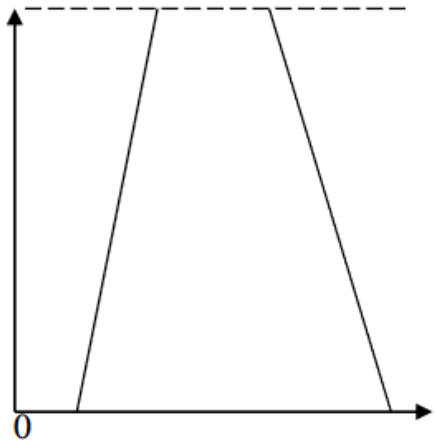

Рис. 2. Трапецеидальная функция принадлежности

3. Гуассова.

$$MF(x) = exp\left[-\left(\frac{x-c}{\varphi}\right)^2\right] \quad (3)$$

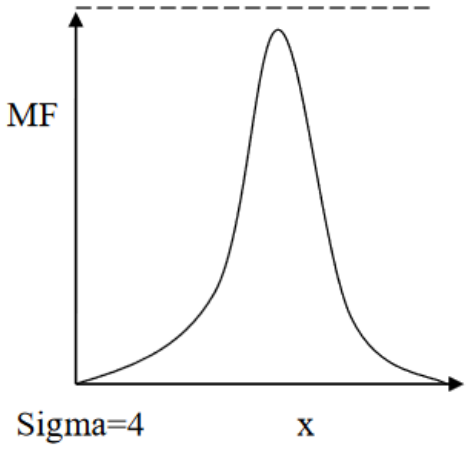

Рис. 3. Гауссова функция принадлежности

2. Нечеткая модель

Понятие нечеткого вывода занимает центральное место в нечеткой логике и в теории нечеткого управления.

Нечеткая модель – математическая модель, в основе вычисления которой лежит нечеткая логика.

Ход построения модели можно разделить на три основных этапа:

1. Определение входных и выходных параметров модели.
2. Построение базы знаний.
3. Выбор одного из методов нечеткого логического вывода.

База правил – это совокупность нечетких правил вида: "если, то", определяющих взаимосвязь между входами и выходами исследуемого объекта в формате:

Если условие (посылка) правила, то заключение правила.

Иначе, это система нечетких продукционных правил, отражающая знания экспертов о методах управления объектом в различных ситуациях, характере его функционирования в различных условиях и т.п., т.е. содержащая формализованные человеческие знания.

Каждое правило в системе имеет вес – данный параметр характеризует значимость правила в модели. Весовые коэффициенты присваиваются правилу в диапазоне [0, 1].

Условия, накладываемые на нечеткую модель:

1. Существует хотя бы одно правило для каждой лингвистической выходной переменной.
2. Для любого терма выходной переменной имеется хотя бы одно правило, в котором этот терм используется в качестве целевой части правила.

Нечетким логическим выводом называется получение заключения в виде нечеткого множества, соответствующего текущим значениях входов, с использованием нечеткой базы знаний и нечетких операций.

Рассмотрим схему нечеткого вывода (Рис. 4)

Рис. 4. Процедура нечеткого логического вывода

Фаззификация (введение нечеткости) – это установка соответствия между численным значением входной переменной системы нечеткого вывода и значением функции принадлежности соответствующего ей терма лингвистической переменной.

Агрегирование – это процедура определения степени истинности условий по каждому из правил системы нечеткого вывода. При этом используются полученные на этапе фаззификации значения функций принадлежности термов лингвистических переменных.

Активизация – это процедура или процесс нахождения степени истинности каждого из элементарных логических высказываний (подзаключений).

Дефаззификация – это процесс перехода от функции принадлежности выходной лингвистической переменной к её четкому (числовому) значению. Цель дефаззификации состоит в том, чтобы, используя

результаты аккумуляции всех выходных лингвистических переменных, получить количественные значения для каждой выходной переменной.

Рассмотренные выше этапы нечеткого вывода не определены однозначно, возможны различные способы выполнения данных шагов. Например, активация может проводиться различными методами нечеткой композиции, как и процедура деффазификации нечеткого числа в четкое может проводиться различными путями. Существуют известные алгоритмы нечеткого вывода, которые и определяют выбор конкретных способов обработки данных на каждом этапе.

3. Алгоритм Мамдани

Алгоритм Мамдани был предложен в 1975 году английским ученым Э. Мамдани. Он помогает избегать больших объемов вычислений и используется повсеместно в задачах нечеткого моделирования. Алгоритм использует набор нечетких правил и нечетких функций принадлежности для определения нечетких выходных данных на основе нечетких входных данных.

Шаг 1. Формирование базы правил.

На первом этапе алгоритма происходит формирование базы правил вида: "если, то", с весовыми коэффициентами F_i $(i = 1..q)$ – степени уверенности в истинности получаемого подзаключения, где q – общее число подзаключений в базе правил.

$RULE_1: IF«Condition_1»THEN«Conclusion_1»(F1)AND«Conclusion_2»(F2);$

$RULE_2: IF«Condition_2»AND«Condition_3»THEN«Conclusion_3»(F3);$

...

$$RULE_n: IF«Condition_k»THEN«Conclusion_{(q-1)}»\left(F_{q-1}\right) AND«Conclusion_q»\ (Fq);$$

Шаг 2. Фаззификация входных переменных.

Далее следует этап фаззификации – приведения к нечеткости. На вход модели поступают сформированная база правил и массив значений всех входных переменных $A = \{a_1, \dots, a_m\}, m$ – количество входных переменных. Целью этого этапа является получение значений истинности для всех подусловий из базы правил путем нахождения значений $b_i = \mu(a_i)$, где $i = 1..k$.

Шаг 3. Агрегирование подусловий.

На этом этапе происходит определение степени истинности условий для каждого правила системы нечеткого вывода $c_j = \min(b_i)\ j = 1..n$.

Шаг 4. Активизация подзаключений.

Цель этого этапа – получение совокупности «активизированных» нечетких множеств D_i для каждого из подзаключений в базе правил $(i = 1..q)$.

Для каждого подзаключения находится степень истинности $d_i = c_i * F_i$, где $i = 1..q$.

Затем, каждому i-му подзаключению, сопоставляется множество D_i с новой функцией принадлежности $\mu_i'(x) = min\{d_i, \mu_i(x)\}$, где $\mu_i'(x)$ – «активизированная» функция принадлежности; $\mu_i(x)$ – функция принадлежности терма из подзаключения; d_i – степень истинности i-го подзаключения.

Шаг 5. Аккумулирование заключений.

Аккумуляция заключений – получение нечеткого множества (или их объединения) для каждой из выходных переменных. $E_i = \cup D_j$, $\mu_i'(x) = max\{\mu_1(x), \mu_2(x)\}$, где $\mu_1(x), \mu_2(x)$ – функции принадлежности объединяемых множеств.

Шаг 6. Дефаззификация.

На последнем шаге алгоритма нечеткого логического вывода происходит дефаззификация – получение количественного значения для каждой из выходных лингвистических переменных. По алгоритму Мамдани используется метод центра тяжести по формуле (12).

$$y_i = \frac{\int_{Min}^{Max} x \cdot \mu_i(x)\, dx}{\int_{min}^{Max} \mu_i(x)\, dx}, \qquad (12)$$

где $\mu_i(x)$ – функция принадлежности соответствующего нечеткого множества Ei; Min и Max – границы универсума нечетких переменных; y_i – результат дефаззификации [3].

Алгоритм Мамдани широко используется благодаря своей простоте и интерпретируемости. Он позволяет прозрачно обрабатывать неточную и неопределенную информацию, что делает ее подходящей для приложений, где требуется рассуждение и принятие решений, сходных с человеческими.

4. Пути интеграции разработанной модели

Применяя описанные выше алгоритмы, была разработана нечеткая модель кластеризации объектов интереса на языке программирования Python 3.9. Объектами интереса в данном случае являются позы человека, для которых необходимо вывести степени принадлежности к каждому из четырех заданных классов (“squat” (объект приседает), “walk” (объект идет), “stand” (объект стоит) и “wave” (объект машет рукой)). Была разработана база нечетких правил, состоящая из 6 правил, и функции принадлежности термов лингвистических переменных (“position” (высота позы), “hands” (положение рук), “legs” (состояние ног) и “head” (положение головы)). На вход программы подаются 4 числа – параметры объекта для каждого из термов. Результат работы программы – степени принадлежности объекта ко всем 4 кластерам.

Для решения задачи интеграции разработанного модуля было решено использовать библиотеку Panel, которая используется для того, чтобы конвертировать файлы Jupyter notebook в безопасные веб-приложения.

Panel – это библиотека Python с открытым исходным кодом, которая позволяет создавать настраиваемые интерактивные веб-приложения и информационные панели. С помощью этой библиотеки можно создавать пользователем виджеты, которые могут быть подключены к графикам, изображениям, таблицам или тексту, чтобы сделать их интерактивными и более информативными.

К файлу Jupyter notebook необходимо добавить несколько простых команд, чтобы создать собственное веб-приложение. Оно упрощает работу пользователя, ведь ему не надо поочередно запускать ячейки с кодом. Было решено входные данные вводить с помощью слайдера от 1 до 100. Входными параметрами являются субъективные суждения о четырех составляющих фигуры человека, которая подлежит классификации: информация о высоте позиции в общем, положение рук, ног и головы. На рисунке 5 отображен внешний вид полученного интерфейса.

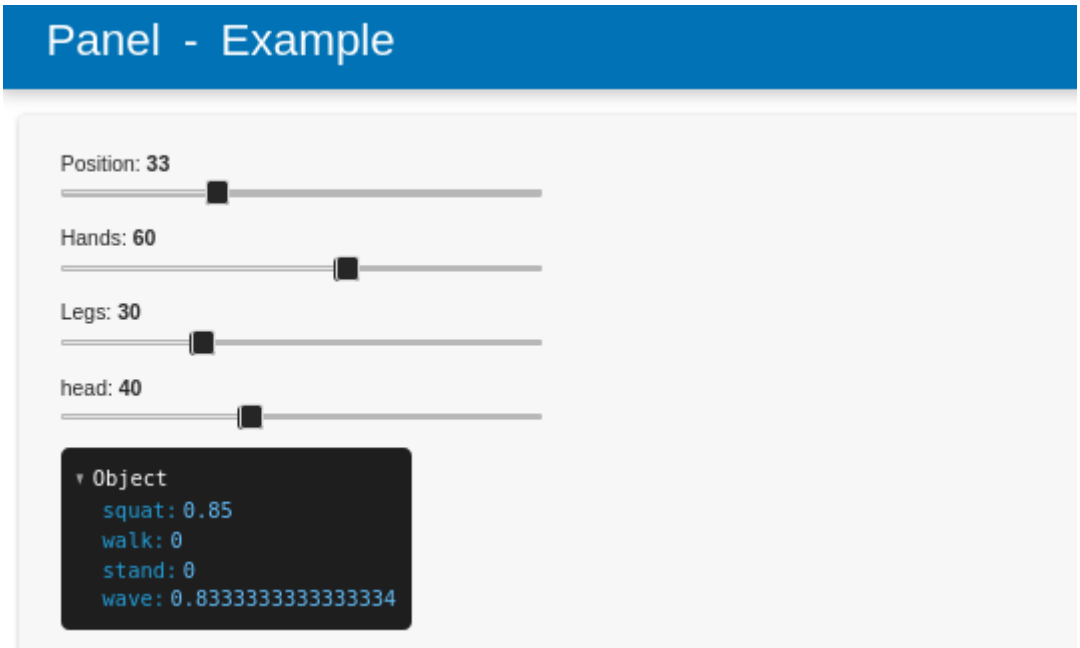

Рис. 5. Внешний вид интерфейса

В черной области располагается полученный результат – степени принадлежности объекта интереса ко всем классам.

Существуют и другие методы интеграции модулей на Python в веб-приложение. Библиотеки Voila, Streamlit и Dash, вместе с Panel, являются лидерами в экосистеме информационных панелей на Python, но при этом у них есть определенные ограничения, которые повлияли на наш выбор – например, Panel подходит для многостраничных приложений с поддержкой

обмена информацией между несколькими страницами, что в других трех библиотеках происходит неявно и требует дополнительных настроек [4].

Большинство современных веб-приложений работают на основе REST API. Это обеспечивает гибкость и масштабируемость, а также позволяет разработчикам отделить внешний код от внутренней логики. REST API также облегчает обновление приложения и внедрение новых функций, не затрагивая его основной функционал. Кроме того, пользователи могут динамически взаимодействовать с интерфейсом, что обеспечивает более удобное и интуитивно понятное использование приложения. Jupyter notebook является мощным инструментом для задач машинного обучения, но при этом для его запуска необходимо открывать файл и запускать все ячейки. REST API позволяет сделать из исполняемого ноутбука бэкэнд и запустить его без открытия самого кода. В источнике [5] представлена документация для создания конечных точек REST API с помощью Jupyter Notebook.

Ещё один путь интеграции заключается в том, чтобы, например, написать свое приложение на C/C++ и обогатить код, встроив в него модуль, написанный в нашем случае на языке программирования Python. Документация представлена в источнике [6]. В итоге пользователь получит программу на C/C++, а от разработчика при необходимости добавления новой функциональности потребуется дописать модуль на языке Python, который более понятен и прост в изучении, чем C/C++. Модуль нечеткого логического вывода был интегрирован данным методом. Из программы, написанной на языке программирования C, происходит вызов модуля и передача ему входных параметров. Результат работы выглядит следующим образом:

Squat 0.8500
Walk 0.0000
Stand 0.0000
Wave 0.8333

5. Выводы

Нечеткая логика является обобщением классической формальной логики. Возникнув сравнительно недавно, она уже стала полноценной методикой принятия решения в управлении. Важно отметить, что для модели нечеткого вывода потребовалось помимо данных вводить базу правил и параметры для модели, что говорит о том, что при её настройке и результатах необходимы знания эксперта.

В качестве ПО для интеграции разработанного модуля нечеткого логического вывода на Python в веб-приложение была выбрана библиотека Panel с учетом её достоинств и простоты в использовании.

Литература

1. Осипов В. П., Судаков В. А. Многокритериальный анализ решений при нечетких областях предпочтений // Препринты ИПМ им. М. В. Келдыша. 2017. № 6. – 16 с.
2. Карпенко А.С. Логика нечёткая / Гуманитарный портал: Концепты [Электронный ресурс] // Центр гуманитарных технологий, 2002–2022 (последняя редакция: 18.11.2022). – URL: https://gtmarket.ru/concepts/6941
3. Рубанов В.Г., Филатов А.Г., Рыбин И.А. Интеллектуальные системы автоматического управления. Нечеткое управление в технических системах. // БГТУ им. В. Г. Шухова. – URL: http://nrsu.bstu.ru/introduction.html
4. Streamlit vs Dash vs Voilà vs Panel - Battle of The Python Dashboarding Giants. – URL: https://medium.datadriveninvestor.com/streamlit-vs-dash-vs-voil%C3%A0-vs-panel-battle-of-the-python-dashboarding-giants-177c40b9ea57
5. The REST API. – URL: https://jupyter-server.readthedocs.io/en/latest/developers/rest-api.html
6. Embedding Python. – URL: https://docs.python.org/3/c-api/intro.html#embedding-python
7. Баданина Н.Д., Зинченко А.А., Судаков В.А. Ранжирование объектов на основе нечеткой кластеризации // Препринты ИПМ им. М.В.Келдыша. 2022. № 68. – 12 с.

References

1. Osipov V. P., Sudakov V. A. Mnogokriterial'nyj analiz reshenij pri nechetkih oblastyah predpochtenij // Preprinty IPM im. M. V. Keldysha. 2017. 006. – 16 p.
2. Karpenko A.S. Logika nechyotkaya / Gumanitarnyj portal: Koncepty [Elektronnyj resurs] // Centr gumanitarnyh tekhnologij, 2002–2022 (poslednyaya redakciya: 18.11.2022). – URL: https://gtmarket.ru/concepts/6941
3. Rubanov V.G., Filatov A.G., Rybin I.A. Intellektual'nye sistemy avtomaticheskogo upravleniya. Nechetkoe upravlenie v tekhnicheskih sistemah. // BGTU im. V. G. SHuhova. – URL: http://nrsu.bstu.ru/introduction.html
4. Streamlit vs Dash vs Voilà vs Panel - Battle of The Python Dashboarding Giants. – URL: https://medium.datadriveninvestor.com/streamlit-vs-dash-vs-voil%C3%A0-vs-panel-battle-of-the-python-dashboarding-giants-177c40b9ea57
5. The REST API. – URL: https://jupyter-server.readthedocs.io/en/latest/developers/rest-api.html

6. Embedding Python. – URL: https://docs.python.org/3/c-api/intro.html#embedding-python
7. Badanina N.D., Zinchenko A.A., Sudakov V.A. Ranzhirovanie obektov na osnove nechetkoj klasterizacii // Preprinty IPM im. M.V.Keldysha. 2022. No 68. –12 p.