

Реализация модели нейро-нечеткой сети на основе байесовского логико-вероятностного подхода для решения задач аппроксимации

Г.А. Хамчичев¹✉, Г.И. Кожомбердиева¹

¹ Петербургский государственный университет путей сообщения Императора Александра I, г. Санкт-Петербург, 190031, Россия

Ссылка для цитирования

Хамчичев Г.А., Кожомбердиева Г.И. Реализация модели нейро-нечеткой сети на основе байесовского логико-вероятностного подхода для решения задач аппроксимации // Программные продукты и системы. 2025. Т. 38. № 1. С. 108–121. doi: 10.15827/0236-235X.149.108-121

Информация о статье

Группа специальностей ВАК: 1.2.1

Поступила в редакцию: 12.06.2024

После доработки: 01.08.2024

Принята к публикации: 12.08.2024

Аннотация. В статье описывается реализация как специализированного программного инструмента модели нейро-нечеткой сети (ННС), основанной на байесовской логико-вероятностной модели нечеткого вывода. Помимо собственно модели ННС, в нем реализованы адаптированные алгоритмы построения (алгоритм сеточного разбиения для генерации нечетких правил) и обучения (алгоритм обратного распространения ошибки и гибридный алгоритм) ННС, известные по использованию для сети ANFIS из пакета MATLAB. Программный инструмент разработан на языке Java, предназначен как для решения практических задач, так и для исследования эффективности и возможностей применения ННС на основе байесовской логико-вероятностной модели нечеткого вывода. Обсуждается опыт построения и обучения ННС предложенного типа с использованием разработанного авторами инструментария. Рассматриваются примеры создания и обучения ННС, ориентированных на решение конкретных задач аппроксимации функций нескольких переменных на основе реальных открытых и синтетических наборов данных. Сравниваются результаты, полученные с помощью разработанного инструмента и инструмента ANFIS из пакета MATLAB. Подтверждается способность предложенной модели ННС служить универсальным аппроксиматором сложных функциональных зависимостей, что позволяет судить об ее эффективности и возможностях применения в разных областях. Включение в программу различных метрик оценки качества аппроксимации позволяет комплексно оценивать качество обучения сети, ее точность, стабильность и адаптивность к новым данным. Ограничение доступа российских потребителей к коммерческому программному обеспечению зарубежных производителей повышает практическую значимость созданного на основе оригинальной модели ННС программного инструмента и делает разработку актуальной и полезной для широкого круга пользователей.

Ключевые слова: нейро-нечеткая сеть, байесовская логико-вероятностная модель нечеткого вывода, машинное обучение, обучение нейро-нечеткой сети, алгоритмы обучения, аппроксимация функций, ANFIS, Java

Благодарности. Авторы искренне признательны д.ф.-м.н. профессору М.М. Луценко за ценные консультации

Введение. Нейросетевой подход к решению задач анализа данных получил широкое распространение, а примеры использования нейронных сетей в системах искусственного интеллекта (включая так называемые генеративные модели) общеизвестны и выглядят впечатляюще. Эффективность применения нейронных сетей в подобных задачах обусловлена тем, что они являются универсальными аппроксиматорами, способными после обучения реализовать на заданном наборе данных любую непрерывную функциональную зависимость.

Менее известны среди непрофессионалов успехи применения в области искусственного интеллекта гибридных нейронных сетей, в которых нейросетевой подход распространяется на системы нечеткого вывода. В России одним из показательных примеров работ в этом направлении является монография [1]. Авторы данной статьи придерживаются классификации гибридных сетей, применяемой современными

зарубежными исследователями [2, 3] и признанной в нашей стране. Различают гибриды двух типов: нечеткие нейронные сети, структура которых сходна с классическими нейронными сетями, но сами нейроны являются нечеткими, то есть имеют нечеткие входы и выходы и/или нечеткие веса, и *нейро-нечеткие сети* (ННС).

Как нейронная сеть особого гибридного типа ННС, с одной стороны, представляет собой нечеткую систему, обладающую способностью формировать выходные сигналы на основе фаззификации значений входных сигналов, использования базы нечетких правил и механизма *нечеткого вывода* (НВ), а далее дефаззификации полученных результатов. С другой стороны, она организована в многослойную нейронную структуру, состоящую как из параметрических, так и из непараметрических слоев, нейроны в которых выполняют расчетные операции в соответствии с выбранной мо-

делью НВ. Благодаря адаптивной настройке значений параметров, задаваемых для нейронов параметрических слоев, сеть способна настраиваться таким образом, чтобы результирующие выходные значения соответствовали ожидаемым, что делает ее способной к обучению.

Известно, что объединение в гибридных системах нейросетевого и нечеткого подходов позволяет как привнести в системы нечеткого вывода способность к обучению и параллелизм вычислений, присущие нейронным сетям, так и усилить интеллектуальные возможности нейронных сетей использованием лингвистически интерпретируемых нечетких правил выработки решений и знаний экспертов. Однако, по мнению авторов, эти преимущества наиболее проявляются в нечетких нейронных сетях. В то же время в ННС аппарат НВ теряет свою семантическую природу, нечеткая терминология лишается своей смысловой значимости, а нечеткие правила упрощаются и приобретают, скорее, формальный характер: лингвистические переменные и функции принадлежности их термов используются просто как элементы модели нейронной сети. В результате такая нейросетевая структура становится самодовлеющей, способной самостоятельно (без создания базы правил экспертами) решать задачи, связанные с аппроксимацией функций нескольких переменных.

Количество новых гибридных сетей и их модификаций постоянно растет. Подробные актуальные обзоры, посвященные гибридным сетям различных типов, тенденциям, проблемам и перспективам их применения, представлены в [2, 3]. В [4] сделан обзор применяемых для таких сетей техник обучения. Имеются многочисленные примеры использования ННС, свидетельствующие об актуальности и интенсивности современных исследований и разработок в этой области искусственного интеллекта. ННС зарекомендовали себя как мощный инструмент, обеспечивающий эффективное решение задач прогнозирования, диагностики, автоматического управления и регулирования, а также других, связанных с анализом и обработкой данных. В ряде работ демонстрируются возможности решения задач прогнозирования временных рядов [5], состояния кабельных линий [6], энергопотребления зданий [7], поверхностного натяжения в материалах [8]. Пример использования ННС для решения задачи классификации показан в работе [9], а для решения задачи анализа и оценки проектных решений – в [10]. ННС успешно применяются в про-

мышленной сфере: в [11, 12] демонстрируется применение ННС в управлении и регулировании, а в [13] – для оптимизации параметров производственного процесса.

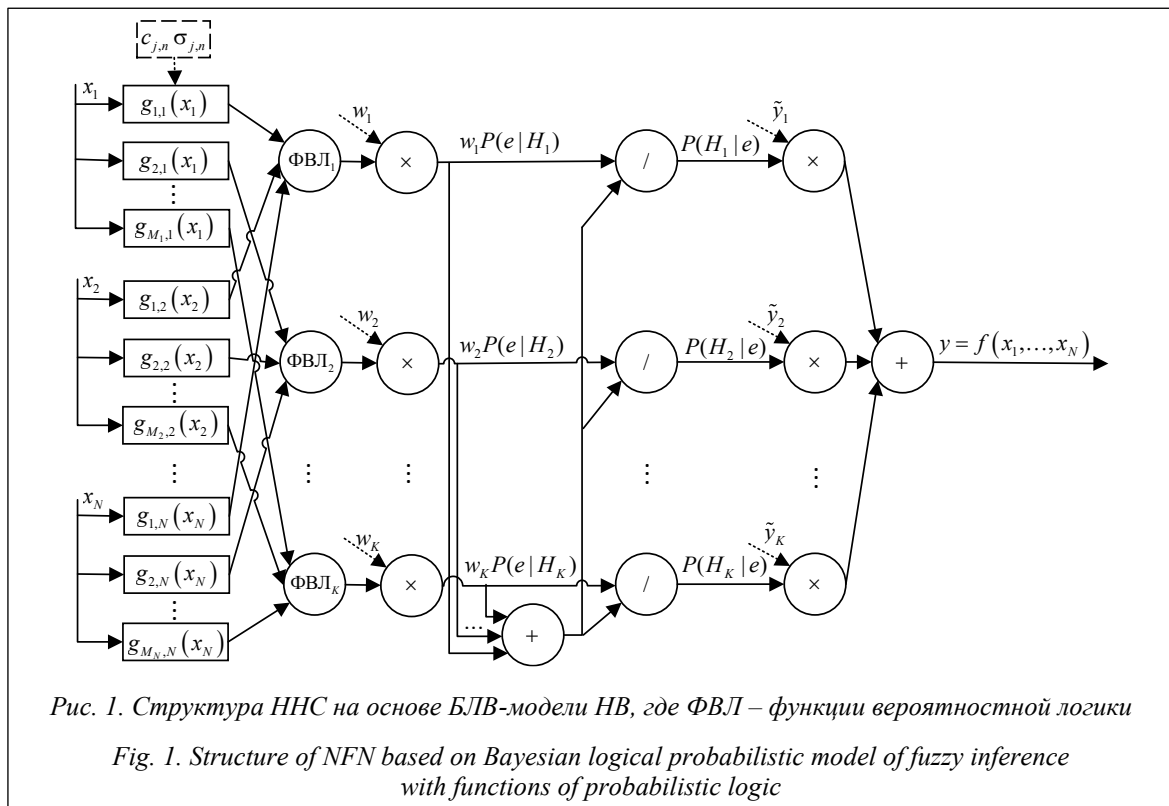
Структуре многослойной ННС на основе байесовской логико-вероятностной модели (БЛВ-модели) НВ и ее апробации посвящены работы [14–16]. Предложенная сетевая структура сопоставима с известными ННС Такаги–Сугено–Канга (TSK) и Ванга–Менделя [17], в частности, с широко применяемой моделью ANFIS (*Adaptive Neuro-Fuzzy Inference System*), также основанной на модели НВ TSK. Особую популярность ANFIS получила в связи с наличием ее готовой реализации в пакете MATLAB (<https://www.mathworks.com/help/fuzzy/neuro-adaptive-learning-and-anfis.html>).

В данной статье демонстрируется практическое применение разработанной программы для создания и обучения ННС на основе БЛВ-модели НВ, ориентированных на решение конкретных задач аппроксимации функций нескольких переменных на реальных и синтетических данных, что позволяет оценить их аппроксимационные способности и эффективность обучения в сравнении с инструментом для создания и обучения сетей ANFIS из MATLAB.

ННС на основе БЛВ-модели

БЛВ-модель НВ предложена в [18]. Позднее она исследована и программно реализована авторами, получено Свидетельство о государственной регистрации программы для ЭВМ RU 2021662943. Опыт этой программной реализации использован при разработке программы для создания и обучения ННС на основе БЛВ-модели НВ. Нейросетевая интерпретация БЛВ-модели НВ и структурная схема ННС на ее основе представлены в [14, 15]. Остановимся кратко на функциональности нейронов, распределенных по слоям сети. Следует отметить, что в процессе программной реализации модели ННС учтены полученные предварительные результаты обучения и внесены полезные изменения в функциональное описание некоторых слоев сети.

Структура сети включает семь слоев, три из которых (первый, третий и шестой) являются параметрическими (рис. 1). Первый слой нейронов выполняет фаззификацию каждой входной переменной x_n , $n = \overline{1, N}$, то есть расчет значения функции принадлежности (ФП) для каждого j -го терма, $j = \overline{1, M_n}$, каждой n -й входной лингвистической переменной (ЛП), при-



чем, в отличие от представления структуры ННС в [14, 15], в программной реализации в качестве ФП используется частный случай ($b = 1$) обобщенной гауссовой функции:

$$g_{j,n}(x_n; c_{j,n}, \sigma_{j,n}) = e^{-\frac{(x_n - c_{j,n})^{2b}}{\sigma_{j,n}^2}}, \quad (1)$$

которая при некоторых значениях показателя степени b может определять треугольную и трапецидальную ФП. Центры $c = (c_{1,1}, \dots, c_{j,n}, \dots)$ и коэффициенты ширины $\sigma = (\sigma_{1,1}, \dots, \sigma_{j,n}, \dots)$ гауссовых ФП являются параметрами первого слоя ННС.

Второй слой нейронов рассчитывает условные вероятности $P(e|H_k)$, $k = 1, \dots, K$, на основе базы нечетких правил, трансформированной в набор функций вероятностной логики. Эти функции в качестве аргументов принимают значения ФП $g_{j,n}$, рассчитанные нейронами первого слоя сети. Отметим, что в качестве алгоритма генерации базы нечетких правил для программной реализации ННС выбран алгоритм сеточного разбиения, чем обусловлен простой вид формулы для расчета условной вероятности k -м нейроном второго слоя:

$$P(e|H_k) = \prod_{n=1}^N g_{j,n}(x_n; c_{j,n}, \sigma_{j,n}). \quad (2)$$

Нейроны третьего слоя выполняют взвешивание значений условных вероятностей параметрами-весами нечетких правил $w = (w_1, \dots, w_K)$.

Нейрон четвертого слоя суммирует взвешенные условные вероятности. Нейроны пятого слоя по формуле, основанной на формуле Байеса, формируют апостериорное распределение вероятностей:

$$P(H_k|e) = \frac{w_k P(e|H_k)}{\sum_{l=1}^K w_l P(e|H_l)}, \quad (3)$$

где K – число термов выходной ЛП (байесовских гипотез H_k), соответствующих числу функций вероятностной логики (оценивающих степень истинности свидетельств в пользу каждой гипотезы); w_k – вес k -го правила, $w_k \in [0; 1]$.

Нейроны шестого слоя и нейрон-сумматор седьмого слоя отвечают за дефазификацию выходной переменной. Итоговая оценка $\hat{y}$ значения выходной переменной y вычисляется по формуле

$$\hat{y} = \sum_{k=1}^K \tilde{y}_k P(H_k|e), \quad (4)$$

где $P(H_k|e)$ – элемент апостериорного распределения вероятностей (3); $\tilde{y}_k$ – характерное значение соответствующего терма выходной ЛП. При этом, в отличие от структуры ННС в [14, 15], в качестве параметров шестого слоя в программной реализации сети выступают не коэффициенты α_k , предназначавшиеся для вычисления значений $\tilde{y}_k$ как выпуклых комбинаций

граничных точек соответствующих термам интервалов на шкале выходной переменной, а непосредственно характерные значения $\tilde{y} = (\tilde{y}_1, \dots, \tilde{y}_K)$ термов выходной ЛП. Результат работы сети – число $\hat{y}$, аппроксимирующее значение некоторой функции $y = f(x_1, \dots, x_N)$.

Таким образом, параметрами ННС на основе БЛВ-модели НВ, которые уточняются и обновляются в процессе обучения сети, являются:

центры c и коэффициенты ширины σ гауссовых ФП (первый слой нейронов);

веса нечетких правил w (третий слой нейронов);

характерные значения термов выходной ЛП $\tilde{y}$ (шестой слой нейронов).

Компоненты векторов c , σ и w отнесем к нелинейным параметрам, поскольку они используются в нелинейных функциях (1) и (3), а $\tilde{y}$ – к линейным, так как они используются в линейной функции (4).

Алгоритмы, используемые при построении и обучении ННС на основе БЛВ-модели НВ

Для полноценной работы с ННС необходимы алгоритмы двух типов. Во-первых, алгоритмы начального формирования ее структуры, которые на основе имеющегося набора данных, выбранного вида ФП и указанного количества термов входных ЛП выполняют генерацию нечетких правил. Во-вторых, алгоритмы ее обучения, выполняющие обновление параметров выражений (1–4) таким образом, чтобы после каждой эпохи обучения сети (одного цикла полного прохода по обучающему набору данных) уменьшалось значение функции потерь:

$$L(y_i, \hat{y}_i(x_i; c, \sigma, w, \tilde{y})) = \frac{1}{2}(y_i - \hat{y}_i)^2, \quad (5)$$

где y_i и $\hat{y}_i$ представляют соответственно целевое и предсказанное значения для i -й строки данных $x_i = (x_1, \dots, x_N)$ из обучающего набора, $i = \overline{1, Z}$, где Z – число строк набора. Обучение заканчивается либо по завершении определенного числа эпох, либо по достижении заданной величины функции потерь L .

Для начального формирования структуры ННС был выбран алгоритм сеточного разбиения, поскольку он доступен в MATLAB и не-

сложен в реализации, а для обучения выбраны алгоритм обратного распространения ошибки (АОРО), основанный на градиентном спуске (ГС), и гибридный алгоритм (ГА) [16], адаптированные для учета особенностей ННС на основе БЛВ-модели НВ. Как показывает практика работы с ANFIS в MATLAB, ГА обеспечивает более быструю сходимость обучения, чем АОРО.

Алгоритм сеточного разбиения

Суть данного алгоритма состоит в равномерном распределении термов каждой n -й входной ЛП, $n = \overline{1, N}$, по шкале значений соответствующей входной переменной. Так, если ФП для термов входной ЛП определяются как гауссовы (1), то шкала делится на $s_n + 1$ равных интервалов, где s_n – выбранное количество термов, и далее в качестве центров $c_{j,n}$ выбираются значения точек соприкосновения этих интервалов, а в качестве коэффициентов ширины $\sigma_{j,n}$ – длины интервалов. С помощью аналогичного разбиения шкалы выходной переменной на $K + 1$ равных интервалов выбираются и $\tilde{y}_k$ – характерные значения термов выходной ЛП. Количество характерных значений K соответствует количеству правил, посылка каждого из которых формируется как конъюнкция утверждений вида $ЛП_n = \text{Терм}_j$, где j – индекс термина n -й входной ЛП. При этом число K равно мощности N -кратного декартова произведения терм-множеств входных ЛП.

Алгоритм сеточного разбиения прост в реализации, имеет высокую скорость работы, однако создает избыточное количество правил, что увеличивает вычислительную сложность.

Алгоритм обратного распространения ошибки

Процесс обучения ННС направлен на минимизацию значения функции (5) путем итеративной коррекции параметров нейронов параметрических слоев ННС. В адаптации АОРО для его применения при обучении ННС на основе БЛВ-модели НВ ключевым элементом является выработка правила обновления параметров сети $c, \sigma, w, \tilde{y}$ на основе ГС при минимизации функции L . Для обновления каждого параметра θ ННС при ее обучении используется выражение

$$\theta^* = \theta - \eta_0 \frac{\partial L}{\partial \theta}, \quad (6)$$

где θ^* – новое значение параметра; θ – его текущее значение; η_0 – темп изменения пара-

метра; $\frac{\partial L}{\partial \theta}$ – частная производная функции (5)

по этому параметру, найденная с использованием цепного правила. В таблице 1 представлены конкретные варианты правила (6), используемые для обновления параметров ННС в ходе обучения.

Таблица 1

Правила обновления параметров ННС

Table 1

Rules for updating NFN parameters

Параметр (θ)	Темп изменения параметра (η_θ)	Ограничение на новое значение параметра (θ^*)
$\tilde{y}_k$	$\eta_{\tilde{y}} = \frac{\lambda_{\tilde{y}}}{\sqrt{\sum_{k=1}^K \left(\frac{\partial L}{\partial \tilde{y}_k} \right)^2}}$	–
w_k	$\eta_w = \frac{\lambda_w}{\sqrt{\sum_{k=1}^K \left(\frac{\partial L}{\partial w_k} \right)^2}}$	$0 \leq w_k^* \leq 1$
$c_{j,n}$	$\eta_{c_{j,n}} = \frac{\lambda_c}{\sqrt{\sum_{n=1}^N \sum_{j=1}^{M_n} \left(\frac{\partial L}{\partial c_{j,n}} \right)^2}}$	–
$\sigma_{j,n}$	$\eta_{\sigma_{j,n}} = \frac{\lambda_\sigma}{\sqrt{\sum_{n=1}^N \sum_{j=1}^{M_n} \left(\frac{\partial L}{\partial \sigma_{j,n}} \right)^2}}$	$\sigma_{j,n}^* \geq 0$

Здесь $\lambda_{\tilde{y}}, \lambda_w, \lambda_c, \lambda_\sigma$ – произвольно подбираемые размеры шага изменения соответствующего параметра сети в ходе обучения ННС. Наряду с числом используемых эпох обучения, пороговым значением функции потерь (5) и направлением градиента в ГС они рассматриваются как параметры обучения ННС.

Осуществляется контроль за тем, чтобы значения весов правил в процессе обучения не выходили за установленный диапазон $[0, 1]$. Для их изменения используется метод спроецированного градиентного спуска [19], а само новое значение веса определяется как $w_k^* = \max(\min(w_k^*, 1), 0)$.

Порядок обновления значений параметров $\tilde{y}_k$ и w_k не имеет значения, так как производные $\frac{\partial L}{\partial \tilde{y}_k}$ и $\frac{\partial L}{\partial w_k}$ не зависят друг от друга. Это

позволяет выполнять их обновление параллельно и независимо, что обеспечивает гибкость в оптимизации функции потерь и упрощает реализацию алгоритма.

Преимуществами АОРО являются высокая эффективность для небольших сетей и простота реализации, однако при его применении возможны проблемы, связанные с попаданием в локальные минимумы, а также он довольно медленный при использовании на сетях с большим числом параметров.

Гибридный алгоритм обучения

Этот алгоритм сочетает в себе как ГС, так и метод наименьших квадратов. Алгоритм ГС применяется для тонкой настройки нелинейных параметров c , σ и w , обеспечивая эффективное снижение общей ошибки (5). В то же время метод наименьших квадратов идеально подходит для быстрой и точной корректировки линейных параметров $\tilde{y}$ благодаря его способности к минимизации ошибок в линейно зависимых данных [17].

Для адаптации ГА часть уже реализованного алгоритма АОРО изменена таким образом, что при обратном проходе не выполняется изменение линейных параметров $\tilde{y}$ с помощью ГС. Вместо этого в начале каждой эпохи обучения осуществляется прямой проход по всему обучающему набору данных и с помощью метода наименьших квадратов линейные параметры $\tilde{y}$ обновляются путем решения системы линейных алгебраических уравнений

$$\tilde{y} = (P^T P)^{-1} P^T Y,$$

где вектор целевых значений Y содержит реальные выходные данные обучающего набора, а матрица P формируется на основе значений выходов пятого слоя сети – апостериорных вероятностей (3), рассчитанных для всех обучающих примеров.

Таким образом, метод наименьших квадратов направлен на нахождение оптимального вектора характерных значений $\tilde{y}$, минимизируя функцию потерь (5). После завершения этого шага происходит обновление нелинейных параметров c , σ и w в результате обратных проходов по каждой строке набора данных, как в АОРО.

Главными преимуществами ГА являются быстрая сходимость (особенно для набора данных с линейными зависимостями) и высокая точность обучения. Недостаток заключается в повышенной по сравнению с АОРО вычислительной сложности.

Метрики оценки качества обученной сети

Для оценки точности и эффективности ННС в задачах аппроксимации в программе используются перечисленные далее известные метрики качества [19, 20].

- MSE (*Mean Squared Error* – среднеквадратическая ошибка). Измеряет среднюю квадратичную разницу между предсказанными $\hat{y}_i$ и реальными значениями y_i , оценивая общую точность модели. Метрика особенно полезна для выявления больших ошибок, так как квадратичная функция штрафует их более сильно.

- RMSE (*Root Mean Squared Error* – корень из среднеквадратической ошибки). Обеспечивает оценку ошибки в тех же единицах, что и исходные данные. В дальнейшем будем ориентироваться на эту ошибку для сравнения предложенной модели ННС с ННС ANFIS из пакета MATLAB, поскольку в ней по умолчанию используется именно эта метрика.

- MAE (*Mean Absolute Error* – средняя абсолютная ошибка). Рассчитывает среднее абсолютное отклонение предсказанных значений $\hat{y}_i$ от фактических y_i , являясь мерой линейной точности.

- MAPE (*Mean Absolute Percentage Error* – средняя абсолютная процентная ошибка). Выражает MAE в процентах, что полезно для оценки относительной точности без привязки к масштабу данных.

- R^2 (коэффициент детерминации). Указывает на долю дисперсии зависимой переменной y , объясненную моделью, оценивая ее предсказательную силу.

Включение данных метрик в программу обусловлено их способностью к комплексной оценке точности, стабильности и адаптивности ННС к новым данным. Полезными для оценки качества обученной сети в контексте аппроксимации также являются скорректированный коэффициент детерминации R^2_{adj} , симметричная средняя абсолютная процентная ошибка SMAPE, взвешенная абсолютная процентная ошибка WAPE, средняя абсолютная масштабированная ошибка MASE. В дальнейшем планируется внедрить их в программу.

Функциональные возможности программы

Разработанная программа обеспечивает построение и обучение ННС на основе БЛВ-модели НВ, которые позволяют эффективно решать задачи аппроксимации функций несколь-

ких переменных. Функциональные возможности программы и поддерживающий их реализацию графический интерфейс пользователя сопоставимы с инструментом ANFIS из пакета MATLAB.

Для разработки программы использован язык Java. Выбор обусловлен не только опытом работы авторов с ним, но и его кроссплатформенностью. К тому же последние версии среды исполнения значительно улучшили производительность (в том числе разработан высокопроизводительный вариант виртуальной машины Java – GraalVM), а язык предоставляет разработчику обширный API и множество библиотек, которые значительно упрощают разработку сложных приложений с продвинутым пользовательским интерфейсом.

Построение сети

Программа поддерживает использование трех типов наборов данных: обучающий, валидационный и тестовый. Для построения сети используется обучающий набор. Пользователь указывает обучающий набор данных, количество термов входных ЛП и начальное значение весов w . На этой основе программа, используя алгоритм сеточного разбиения, автоматически создает нечеткие правила, необходимые для формирования структуры ННС, и определяет начальные значения параметров сети c , σ , $\tilde{y}$.

В программе не поддерживается возможность редактирования правил, сгенерированных на этапе построения сети, причем правила содержат только просто реализуемую операцию конъюнкции. Однако на практике использования в сетях типа ANFIS лишь конъюнкции в нечетких правилах достаточно для получения точного результата в решаемых задачах, что подтверждается не только работой автора сети ANFIS [21], но и широким использованием этой модели в большом количестве приложений.

Обучение сети

Для начала обучения нужно выбрать алгоритм обучения (АОРО или ГА), ввести целевое значение ошибки и количество эпох обучения сети. Во время обучения программа отображает динамические графики изменения ключевых метрик качества аппроксимации (<http://www.swsys.ru/uploaded/image/2025-1/1.jpg>), а также графики изменения значений параметров сети. Пользователь может в любой момент приостановить обучение сети, продолжить этот процесс или завершить его.

В программе реализована возможность настройки параметров обучения, позволяющая

отключать и включать обновление параметров для каждого параметрического слоя ННС, менять шаги их изменения, выбирать направление градиента и стратегию изменения значений параметров обучения (шагов изменения параметров сети) во время обучения, а также вручную настраивать параметры стратегий обучения (<http://www.swsys.ru/uploaded/image/2025-1/2.jpg>).

В программе поддерживаются две стратегии изменения значений параметров обучения. Одна из них – классическая, используемая в MATLAB. Суть ее в том, что при уменьшении ошибки на протяжении четырех эпох значение шага увеличивается, а если ошибка претерпевает две последовательные комбинации из одного увеличения и одного уменьшения, шаг уменьшается. Вторая стратегия – возврат значений параметров на определенное задаваемое число эпох назад при увеличении ошибки.

Проверка сети

После завершения обучения сеть подлежит проверке с использованием наборов данных для валидации и тестирования, что позволяет оценить ее способность к обобщению и точность предсказаний на новых данных. Валидационный набор данных, который не участвует в обучении сети, позволяет настраивать параметры обучения и предотвращает ее переобучение. Тестовый набор аналогично валидационному состоит из данных, не участвующих в обучении, и предназначен для окончательной проверки обобщающей способности обученной сети после тонкой настройки и валидации, имитируя ее реальное применение [20]. Для проверки сети нужно выбрать один из предварительно загруженных наборов данных. Если набор данных корректен, то результаты, включая значения ключевых метрик, отобразятся в специализированном текстовом поле графического интерфейса программы.

Пользователь может проверять работу сети на собственных данных, вводя значения входных переменных для получения дефазифицированного результата. Это позволяет проводить непосредственную проверку эффективности обученной ННС на специфичных для конкретного пользователя данных, упрощая процесс ее тестирования и адаптации под индивидуальные нужды. Результат расчета на основе текущих значений параметров сети отображается в специальном текстовом поле, включающем также и оценку ошибки вычисления в виде границ доверительного интервала $\pm RMSE$.

Пользователь имеет возможность сохранить текущее состояние сети со всеми значениями параметров, чтобы в дальнейшем можно было к нему вернуться. При экспорте и выборе опции «Совместимость» сеть будет экспортирована в формат, совместимый с авторской программой для решения задач нечеткого вывода на основе БЛВ-модели.

Примеры создания сетей, ориентированных на решение практических задач аппроксимации

Для исследования возможностей предложенной модели ННС выбрано несколько наборов данных: два набора (*Boston Housing Dataset* – ВН и *Auto MPG* – АМ), взятых с репозитория для исследователей в области машинного обучения и анализа данных Delve Datasets и UCI Machine Learning Repository, и два *синтетических набора данных* (СНД1 и СНД3), сформированных на основе вычисления заданной функции с последующим зашумлением.

Характеристика наборов данных

1. Набор данных ВН (<https://www.cs.toronto.edu/~delve/data/boston/bostonDetail.html>) содержит информацию о различных характеристиках жилых домов в окрестностях Бостона, включая среднее количество комнат в доме, уровень преступности, налоги на недвижимость и многое другое. Всего в наборе используются 14 характеристик, включая выходную переменную – сведения об их цене.

2. Набор данных АМ (<https://archive.ics.uci.edu/dataset/9/auto+mpg>) описывает различные характеристики автомобилей и содержит 9 переменных.

Наборы ВН и АМ имеют большую размерность, из-за чего их обработка проблематична даже в MATLAB: при попытке создать ННС на основе полного набора данных возникает нехватка памяти. Это связано с тем, что сложность ННС экспоненциально зависит от количества термов ЛП, построенных на основе входных переменных. Так как входные переменные в разной степени влияют на результат, было принято решение о сокращении размерности НД до двух входных переменных для АМ и трех входных для ВН, а также одной выходной переменной. Для этого была использована техника сокращения размерности пространства признаков на основе корреляционного анализа взаимосвязи между входными переменными и целевой переменной. Кроме того, были удалены строки с отсутствующими значениями переменных.

3. СНД1 сгенерирован на основе вычисления функции одной переменной $f(x) = -4x^3 + 8x^2 + 5x - 1, x \in [-1,5; 3]$. К результатам вычисления $f(x)$ добавлен гауссов шум с распределением $N(0;1)$ для имитации реальных условий измерений.

4. СНД3 сгенерирован на основе функции трех переменных $f(x, y, z) = x^2 - 4y + 5z + 2, x, y, z \in [-4; 4]$. К значениям функции $f(x, y, z)$ добавлен гауссов шум с распределением $N(0; 0,5)$ для имитации реальных условий измерений.

Итоговые характеристики используемых для создания и обучения ННС наборов данных приведены в таблице 2.

Результаты обучения ННС

Сравниваются результаты обучения ННС на основе БЛВ-модели НВ и ННС типа ANFIS с одинаковыми значениями количества термов входных ЛП, равными трем. Для описания термов ЛП используются гауссовы ФП (1). Для обучения применяется ГА, так как он демонстрирует наилучшую скорость обучения для обеих моделей.

Для сравнения качества обучения сетей используется метрика качества RMSE, поскольку, с одной стороны, она позволяет адекватно оценить шум в данных, а с другой, это единственная метрика, доступная как в MATLAB, так и в разработанной авторами программе. Зависимость RMSE от числа использованных эпох обучения на начальном периоде обучения, ко-

гда наблюдается активное уменьшение ошибки, иллюстрируется графиками. Итоговая сходимость, достигнутая на сравниваемых моделях, характеризуется минимальными изменениями RMSE в последующих эпохах и представлена в виде итоговых значений в соответствующих таблицах.

Пример Boston Housing. График на рисунке 2 и таблица 3 иллюстрируют изменение RMSE на первых 50 эпохах обучения ННС. Отмечается значительное улучшение RMSE после обучения обеих моделей ННС. График подтверждает их схожесть по эффективности и темпу обучения. Однако на тестовом наборе данных ННС на основе БЛВ-модели НВ демонстрирует большее значение RMSE, что указывает на более высокую способность ANFIS к обобщению на новых данных. Тем не менее близость значений RMSE на обучающем и валидационном наборах подчеркивает большой потенциал ННС на основе БЛВ-модели НВ в качестве инструмента аппроксимации сложных функциональных зависимостей.

На рисунке 3 приведены графики изменения значения RMSE по мере увеличения числа эпох обучения с использованием алгоритмов АОРО и ГА для обучения ННС ANFIS (а) и ННС на основе БЛВ-модели НВ (б). Графики показывают, что при использовании ГА для обучения ANFIS RMSE быстро уменьшается и стабилизируется на низком уровне, в то время

Характеристики наборов данных, используемых для создания и обучения ННС

Таблица 2

Table 2

Characteristics of datasets for creating and training NFN

Набор данных	Число входных переменных		Мощность набора данных			
	Исходное	Принятое	Всего строк	Обучающий	Валидационный	Тестовый
ВН	13	3	452	271	90	91
АМ	8	2	398	238	79	81
СНД1	1	1	200	120	40	40
СНД3	3	3	1000	600	200	200

Сравнение ННС по метрике RMSE для набора ВН

Таблица 3

Table 3

Comparison of NFNs by RMSE metric for the Boston Housing dataset

Тип ННС	ННС на основе БЛВ-модели НВ			ANFIS		
Набор данных	обуч.	вал.	тест.	обуч.	вал.	тест.
До обучения	7,41	6,15	13,86	26,53	25,77	20,81
Лучшая эпоха	2,58	2,86	10,20	2,61	2,67	7,47

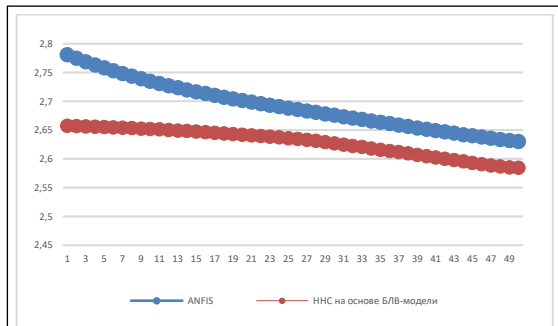


Рис. 2. График изменения RMSE для набора BH на первых 50 эпохах обучения

Fig. 2. Diagram of RMSE variation for the Boston Housing dataset during the first 50 training epochs

как при использовании АОРО уменьшение происходит медленнее и не достигает таких низких значений. Аналогичная тенденция наблюдается и для ННС на основе БЛВ-модели НВ, где ГА также обеспечивает более быстрое и эффективное снижение RMSE по сравнению с АОРО. Это подтверждает эффективность применения ГА для обучения ННС, поскольку он объединяет преимущества ГС и метода наименьших квадратов, позволяющие более эффективно находить оптимальные значения нелинейных и линейных параметров нейронов сети в процессе ее обучения.

Пример Auto MPG. Результаты обучения ННС на этом наборе приведены в таблице 4.

Видно, что разница значений RMSE у обеих моделей ННС невелика как на начальном этапе, так и после обучения, что свидетельствует о сравнимом качестве аппроксимации данных обеими моделями. При этом ННС на основе БЛВ-модели НВ в отличие от ANFIS демонстрирует тенденцию уменьшения значения этого показателя в ходе обучения.

Пример СНД1. Результаты обучения сетей приведены в таблице 5.

Анализ результатов показывает, что обе модели ННС достигли значительного уменьшения RMSE по итогам обучения. График изменения RMSE (<http://www.swsys.ru/uploaded/image/2025-1/3.jpg>) демонстрирует стабильное улучшение качества аппроксимации для обеих моделей по мере их обучения. По завершении этапа обучения ННС на основе БЛВ-модели НВ достигает более высокого качества, чем ANFIS. Согласно данным таблицы 5, она показывает лучшие результаты на всех типах наборов данных, что подтверждает ее высокую адаптивность и способность к аппроксимации нелинейных зависимостей в условиях шума.

Пример СНД3. Результаты обучения сетей приведены в таблице 6.

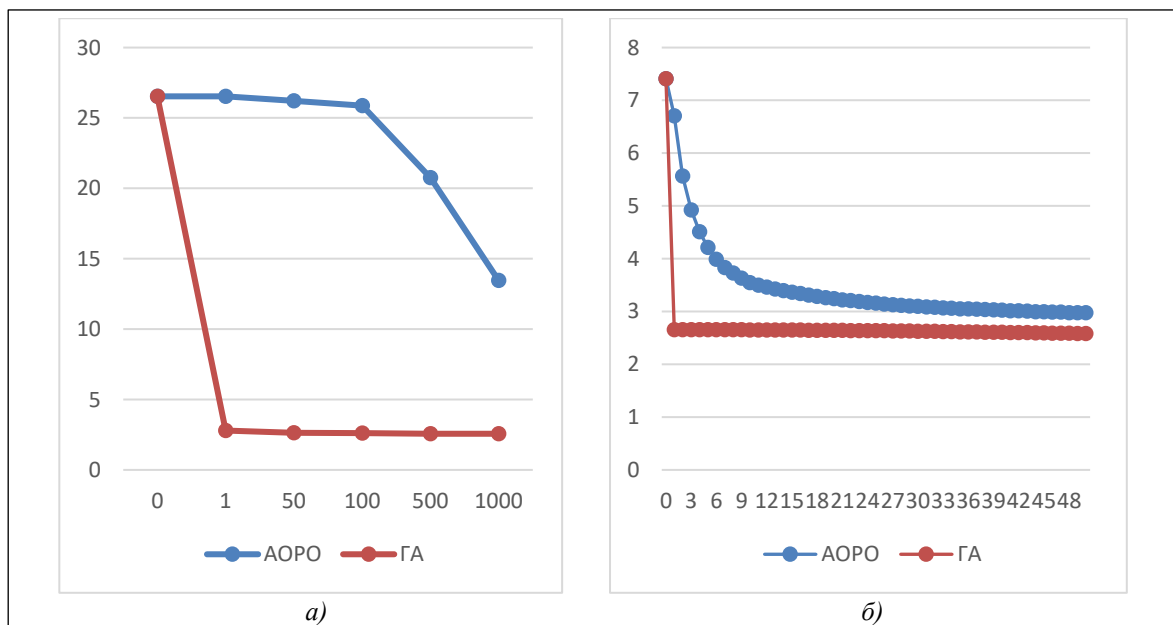


Рис. 3. Графики изменения RMSE на наборе BH для алгоритмов обучения АОРО и ГА: а) для ННС ANFIS; б) для ННС на основе БЛВ-модели НВ

Fig. 3. Diagrams of RMSE variation on the Boston Housing dataset for error backpropagation and hybrid training algorithms: а) for ANFIS NFN; б) for NFN based on the Bayesian logical probabilistic model of fuzzy inference

На СНДЗ было проведено множество экспериментов с обучением ННС на основе БЛВ-модели НВ и подбора наилучших значений параметров обучения. В таблице 6 представлен результат с наименьшей полученной ошибкой, достигнутый при лимите эпох 2 500, начальных шагах для весов правил и параметров ФП 0,00005 и 0,0005 соответственно, при количестве термов входных ЛП, равном 3, и начальных весах правил 0,5. Параметры, установленные для обучения ННС ANFIS (количество термов входных ЛП – 3, количество эпох – 10 000), при этом не менялись.

График (<http://www.swsys.ru/uploaded/image/2025-1/4.jpg>) показывает, что ННС на основе БЛВ-модели НВ достигает стабильно низких значений ошибки быстрее, чем ANFIS, что свидетельствует в пользу предложенного подхода.

Из таблицы 6 видно, что ANFIS показывает лучшие результаты на тестовом наборе данных по сравнению с предложенной моделью ННС. Тем не менее результаты говорят и о высоком потенциале ННС на основе БЛВ-модели НВ как универсальном аппроксиматоре сложных функциональных зависимостей.

Несмотря на то, что для ННС на основе БЛВ-модели НВ при работе с СНДЗ не удалось получить значение RMSE, меньшее 0,7, скорее всего, это возможно при подборе подходящей стратегии обучения, поскольку ННС данного типа имеет больше параметрических слоев, чем ANFIS, и поэтому может быть настроена более гибко. Однако такая настройка ведет к возрастанию сложности обучения и требует тщательного подбора значений таких параметров обучения, как шаги изменения параметров сети λ_0 . Это критически важно для обеспечения сходимости, следовательно, требует дополнительных исследований и экспериментов для достижения наилучших результатов.

По мнению авторов, ННС как гибридная система, объединяющая нейросетевой и нечеткий подходы, имеет особые перспективные достоинства в контексте исследований в области искусственного интеллекта. Поскольку в ходе обучения ННС меняются сгенерированные при ее создании параметры ФП термов ЛП, а также веса правил, по завершении ее обучения нейронная структура ННС реализует нечеткие правила, подобные правилам предметных экспертов и релевантные той задаче, на решение

Сравнение ННС по метрике RMSE для набора AM

Таблица 4

Table 4

Comparison of NFN using RMSE metric for the auto MPG

Тип ННС	ННС на основе БЛВ-модели НВ			ANFIS		
	обуч.	вал.	тест.	обуч.	вал.	тест.
До обучения	13,915	14,234	13,428	24,95	24,426	24,609
Лучшая эпоха	4,04	4,2844	4,1762	4,0648	5,0002	4,2639

Таблица 5

Сравнение ННС по метрике RMSE для СНД1

Table 5

Comparison of NFN by RMSE metric for synthetic dataset 1

Тип ННС	ННС на основе БЛВ-модели НВ			ANFIS		
	обуч.	вал.	тест.	обуч.	вал.	тест.
До обучения	9,914	12,569	9,367	8,238	10,238	7,599
Лучшая эпоха	3,726	4,986	4	3,941	4,313	3,758

Таблица 6

Сравнение ННС по метрике RMSE для СНД3

Table 6

Comparison of NFN by RMSE metric for synthetic dataset 3

Тип ННС	ННС на основе БЛВ-модели НВ			ANFIS		
	обуч.	вал.	тест.	обуч.	вал.	тест.
До обучения	70,33	68,958	70,663	61,783	62,496	59,67
Лучшая эпоха	0,7153	0,7908	0,7220	0,5237	0,5203	0,5426

которой ориентирована данная ННС. После оптимизации эти правила могут быть извлечены из ННС и представлены в лингвистически интерпретируемом виде, что позволяет объяснить закономерности, выявленные ННС в предъявленном ей наборе данных. Это открывает возможности использования ННС в актуальном направлении исследований технологий искусственного интеллекта, известном как объяснимый искусственный интеллект и ставящем целью разработку интеллектуальных систем, решения которых могут быть объяснены человеком [22].

Заключение

В статье рассмотрена модель ННС на основе БЛВ-модели НВ. Представлен разработанный авторами программный инструмент по созданию и обучению ННС этого типа, который предназначен для решения задач, сводимых к аппроксимации функциональных зависимостей, а также для исследования эффективности и возможностей использования ННС на основе БЛВ-модели НВ при решении таких задач. Эксперименты по построению ННС на основе БЛВ-модели НВ с использованием программы показали следующее.

– ННС, обученные с помощью алгоритма обучения ГА, достигают лучших результатов аппроксимации, чем при использовании АОРО, что связано с объединением в ГА преимуществ градиентного спуска и метода наименьших квадратов. При этом скорость обучения ННС зависит от степени линейности зависимости выходной переменной от входных. Если в данных присутствует нелинейность, то ННС обучается медленнее даже при использовании ГА.

– Шаги изменения нелинейных параметров ННС (весов нечетких правил и параметров ФП входных ЛП) должны быть сравнимы друг с другом и не превышать 0,01, а шаг для линейных параметров выходной ЛП должен изменяться в пределах 0,1–1, чтобы обновление параметров сети выполнялось согласованно и ошибка постоянно уменьшалась в процессе обучения.

В целом проведенные эксперименты позволяют сделать оптимистичные выводы о способности ННС этого типа к обучению и решению практических задач в областях, в которых обычно применяются ННС типа ANFIS, а также о сопоставимой эффективности этих сетей.

В дальнейшем планируется продолжить исследование способов согласования параметров обучения, в том числе использование стратегий с асинхронным изменением шагов обучения. Помимо этого, предполагается расширить набор доступных в программе типов ФП и алгоритмов генерации правил (например, основанных на методах кластеризации данных, а также на генетических алгоритмах) и обучения сети, метрик оценок качества обучения сети, а также дополнить список стратегий обучения. Кроме того, планируется задействовать методы оптимизации баз нечетких правил для улучшения производительности ННС и провести сравнение этих методов для оценки их эффективности при решении различных задач. Для этого могут быть задействованы генетические алгоритмы, методы роя частиц и другие подходы автоматической оптимизации нечетких правил.

Необходимо отметить, что у авторов вызывает интерес и такое возможное направление использования ННС на основе БЛВ-модели НВ, как объяснимый искусственный интеллект.

Список литературы

1. Борисов В.В., Круглов В.В., Федулов А.С. Нечеткие модели и сети. М.: Горячая линия–Телеком, 2018. 285 с.
2. de Campos Souza P.V. Fuzzy neural networks and neuro-fuzzy networks: A review the main techniques and applications used in the literature. *Applied Soft Computing*, 2020, vol. 92, art. 106275. doi: 10.1016/j.asoc.2020.106275.
3. Talpur N., Abdulkadir S.J., Alhussian H., Hasan M.H., Aziz N., Bamhdi A. Deep Neuro-Fuzzy System application trends, challenges, and future perspectives: A systematic survey. *Artificial Intelligence Review*, 2023, vol. 56, pp. 865–913. doi: 10.1007/s10462-022-10188-3.
4. Karaboga D., Kaya E. Adaptive network based fuzzy inference system (ANFIS) training approaches: A comprehensive survey. *Artificial Intelligence Review*, 2019, vol. 52, pp. 2263–2293. doi: 10.1007/s10462-017-9610-2.
5. Ярушев С.А., Аверкин А.Н., Павлов В.Ю. Когнитивные гибридные системы поддержки принятия решений и прогнозирования // Программные продукты и системы. 2017. Т. 30. № 4. С. 632–642. doi: 10.15827/0236-235X.120.632-642.
6. Бирюлин В.И., Куделина Д.В., Серебровский В.И., Сафронов Р.И., Гнездилова Ю.П. Использование нечетких нейронных сетей для изучения процессов старения изоляции кабельных линий // Современные ресурсоэффективные технологии и технические средства в АПК: матер. науч.-практич. конф. 2021. С. 208–213.

7. Ghenai C., Al-Mufti O.A.A., Al-Isawi O.A.M., Amirah L.H.L., Merabet A. Short-term building electrical load forecasting using adaptive neuro-fuzzy inference system (ANFIS). *J. of Building Engineering*, 2022, vol. 52, art. 104323. doi: 10.1016/j.jobee.2022.104323.
8. Soleimani R., Saeedi Dehaghani A.H. Insights into the estimation of surface tensions of mixtures based on designable green materials using an ensemble learning scheme. *Sci. Reports*, 2023, vol. 13, art. 14145. doi: 10.1038/s41598-023-41448-z.
9. Ravichandran, B.D., Keikhosrokiani, P. Classification of Covid-19 misinformation on social media based on neuro-fuzzy and neural network: A systematic review. *Neural Computing and Applications*, 2023, vol. 35, pp. 699–717. doi: 10.1007/s00521-022-07797-y.
10. Szafranko E., Srokosz P.E., Jurczak M., Śmieja M. Application of ANFIS in the preparation of expert opinions and evaluation of building design variants in the context of processing large amounts of data. *Automation in Construction*, 2022, vol. 133, art. 104045. doi: 10.1016/j.autcon.2021.104045.
11. Васильев С.Н., Кудинов Ю.И., Пашенко Ф.Ф., Дургарян И.С. и др. Интеллектуальные системы управления и нечеткие регуляторы. Часть II. Обучаемые нечеткие регуляторы, нечеткие ПИД-регуляторы // Датчики и системы. 2017. № 2. С. 3–12.
12. Дивеев А.И., Полтавский А.В., Алхатем А. Нейронечеткое управление процессом сушки пиломатериалов // Надежность и качество сложных систем. 2021. № 3. С. 93–97. doi: 10.21685/2307-4205-2021-3-12.
13. Mondal N., Banik S., Paul S., Sarkar S., Mandal S., Ghosh S. ANFIS-TLBO-based optimization of drilling parameters to minimize burr formation in aluminum 6061. *Multiscale and Multidisciplinary Modeling, Experiments and Design*, 2024, vol. 7, pp. 3607–3623. doi: 10.1007/s41939-024-00433-3.
14. Кожомбердиева Г.И., Бураков Д.П., Хамчиев Г.А. Структура нейро-нечеткой сети на основе байесовской логико-вероятностной модели // Мягкие измерения и вычисления. 2022. Т. 61. № 12. С. 52–64. doi: 10.36871/2618-9976.2022.12.004.
15. Kozhombardieva G.I., Burakov D.P., Khamchichev G.A. Neural network interpretation of Bayesian logical-probabilistic fuzzy inference model. *Proc. ISAIC*, vol. 1, pp. 50–56. doi: 10.5220/0011901700003612.
16. Хамчиев Г.А., Кожомбердиева Г.И. О настройке и возможностях обучения нейро-нечеткой сети на основе байесовской логико-вероятностной модели // Междунар. конф. по мягким вычислениям и измерениям. 2023. Т. 1. С. 29–33.
17. Осовский С. Нейронные сети для обработки информации; [пер. с польск.]. М.: Горячая линия–Телеком, 2018. 448 с.
18. Кожомбердиева Г.И. Байесовская логико-вероятностная модель нечеткого вывода // Междунар. конф. по мягким вычислениям и измерениям. 2019. Т. 1. С. 35–38.
19. Мэрфи К.П. Вероятностное машинное обучение: введение; [пер. с англ.]. М.: ДМК Пресс, 2022. 940 с.
20. Бишоп К.М. Распознавание образов и машинное обучение; [пер. с англ.]. СПб: ООО Диалектика. 2020. 960 с.
21. Jang J.-S.R. ANFIS: Adaptive-network-based fuzzy inference system. *IEEE Transactions on Systems, Man, and Cybernetics*, 1993, vol. 23, no. 3, pp. 665–685. doi: 10.1109/21.256541.
22. Аверкин А.Н., Ярушев С.А. Исследование развития систем объяснительного искусственного интеллекта // ИММВ-2022: сб. науч. тр. 2022. С. 127–134.

Solving approximation problems: Implementing a neuro-fuzzy network model based on a Bayesian logical-probabilistic approach

Georgy A. Khamchichev , Gulnara I. Kozhombardieva ¹

¹ Emperor Alexander I St. Petersburg State Transport University,
St. Petersburg, 190031, Russian Federation

For citation

Khamchichev, G.A., Kozhombardieva (2025) ‘Solving approximation problems: Implementing a neuro-fuzzy network model based on a Bayesian logical-probabilistic approach’, *Software & Systems*, 38(1), pp. 108–121 (in Russ.). doi: 10.15827/0236-235X.149.108-121

Article info

Received: 12.06.2024

After revision: 01.08.2024

Accepted: 12.08.2024

Abstract. The paper describes the implementation of a neuro-fuzzy network (NFN) model based on Bayesian logical-probabilistic fuzzy inference model as a specialized software tool. In addition to the NFN model itself, it implements

adapted algorithms for constructing (grid partitioning algorithm for generating fuzzy rules) and training (error backpropagation algorithm and hybrid algorithm) NFN known from ANFIS network in MATLAB. The designed software tool is written in Java. It solves practical problems and investigates the effectiveness and applications of NFN based on the Bayesian logical-probabilistic fuzzy inference model. The authors discuss the experience of building and training the proposed NFN type using the developed toolkit. The authors also consider the creation and training examples of NFNs designed to solve specific approximation problems of multivariable functions based on real open and synthetic data sets. The paper compares the results obtained using the developed tool and the ANFIS tool from MATLAB. The authors confirmed that the proposed NFN model can be a universal approximator of complex functional dependencies. This confirms its efficiency and possibilities of using it in different fields. The inclusion of various approximation quality metrics in the program allows evaluating network training quality, accuracy, stability, and adaptability to new data comprehensively. Access restrictions for Russian users to foreign commercial software enhance the practical significance of the developed software tool based on the original NFN model; it makes the development relevant and useful for a wide range of users.

Keywords: neuro-fuzzy network, Bayesian logical-probabilistic model of fuzzy inference, machine learning, neuro-fuzzy network training, learning algorithms, function approximation, ANFIS, Java

Acknowledgements. The authors are sincerely grateful to Dr. Sci. (Phys. and Math.), Prof. M.M. Lutsenko for providing valuable consultations

References

1. Borisov, V.V., Kruglov, V.V., Fedulov, A.S. (2018) *Fuzzy Models and Networks*. Moscow, 285 p. (in Russ.).
2. de Campos Souza, P.V. (2020) 'Fuzzy neural networks and neuro-fuzzy networks: A review of the main techniques and applications used in the literature', *Applied Soft Computing*, 92, art. 106275. doi: 10.1016/j.asoc.2020.106275.
3. Talpur, N., Abdulkadir, S.J., Alhussian, H., Hasan, M.H., Aziz, N., Bamhdi, A. (2023) 'Deep Neuro-Fuzzy System application trends, challenges, and future perspectives: A systematic survey', *Artificial Intelligence Review*, 56, pp. 865–913. doi: 10.1007/s10462-022-10188-3.
4. Karaboga, D., Kaya, E. (2019) 'Adaptive network based fuzzy inference system (ANFIS) training approaches: a comprehensive survey', *Artificial Intelligence Review*, 52, pp. 2263–2293. doi: 10.1007/s10462-017-9610-2.
5. Averkin, A.N., Yarushev, S.A., Pavlov, V.Yu. (2017) 'Cognitive hybrid systems for decision support and forecasting', *Software & Systems*, 30(4), pp. 632–642 (in Russ.). doi: 10.15827/0236-235X.120.632-642.
6. Biryulin, V.I., Kudelina, D.V., Serebrovsky, V.I., Safronov, R.I., Gnezdilova, Yu.P. (2021) 'The use of fuzzy neural networks for studying the aging processes of the cable lines insulation', *Proc. Sci.-Pract. Conf. Modern Resource-Efficient Technologies and Technical Means in Agriculture*, pp. 208–213 (in Russ.).
7. Ghenai, C., Al-Mufti, O.A.A., Al-Isawi, O.A.M., Amirah, L.H.L., Merabet, A. (2022) 'Short-term building electrical load forecasting using adaptive neuro-fuzzy inference system (ANFIS)', *J. of Building Engineering*, 52, art. 104323. doi: 10.1016/j.job.2022.104323.
8. Soleimani, R., Saeedi Dehaghani, A.H. (2023) 'Insights into the estimation of surface tensions of mixtures based on designable green materials using an ensemble learning scheme', *Sci. Reports*, 13, art. 14145. doi: 10.1038/s41598-023-41448-z.
9. Ravichandran, B.D., Keikhosrokiani, P. (2023) 'Classification of Covid-19 misinformation on social media based on neuro-fuzzy and neural network: A systematic review', *Neural Computing and Applications*, 35(1), pp. 699–717. doi: 10.1007/s00521-022-07797-y.
10. Szafranko, E., Srokosz, P.E., Jurczak, M., Śmieja, M. (2022) 'Application of ANFIS in the preparation of expert opinions and evaluation of building design variants in the context of processing large amounts of data', *Automation in Construction*, 133, art. 104045. doi: 10.1016/j.autcon.2021.104045.
11. Vassilyev, S.N., Kudinov, Yu.I., Pashchenko, F.F., Durgaryan, I.S. et al. (2017) 'Intelligent control systems and fuzzy controllers. Part II. Trained fuzzy controllers, fuzzy PID controllers', *Sensors and Systems*, (2), pp. 3–12 (in Russ.).
12. Diveev, A.I., Poltavskiy, A.V., Alhatem, A. (2021) 'Neuro-fuzzy control of the lumber drying process', *Reliability and Quality of Complex Systems*, (3), pp. 93–97 (in Russ.). doi: 10.21685/2307-4205-2021-3-12.
13. Mondal, N., Banik, S., Paul, S., Sarkar, S., Mandal, S., Ghosh, S. (2024) 'ANFIS-TLBO-based optimization of drilling parameters to minimize burr formation in aluminum 6061', *Multiscale and Multidisciplinary Modeling, Experiments and Design*, 7, pp. 3607–3623. doi: 10.1007/s41939-024-00433-3.
14. Kozhombierdieva, G.I., Burakov, D.P., Khamchichev, G.A. (2022) 'The structure of a neuro-fuzzy network based on Bayesian logical-probabilistic model', *Soft Measurements and Computing*, 61(12), pp. 52–64 (in Russ.). doi: 10.36871/2618-9976.2022.12.004.
15. Kozhombierdieva, G.I., Burakov, D.P., Khamchichev, G.A. (2022) 'Neural network interpretation of Bayesian logical-probabilistic fuzzy inference model', *Proc. ISAIC*, 1, pp. 50–56. doi: 10.5220/0011901700003612.
16. Khamchichev, G.A., Kozhombierdieva, G.I. (2023) 'On setting up and the training capabilities of a neuro-fuzzy network based on a Bayesian logical-probabilistic model', *Proc. Int. Conf. SCM*, 1, pp. 29–33 (in Russ.).
17. Osowski, S. (2000) *Sieci Neuronowe Do Przetwarzania Informacji*. 420 p. (Russ. ed.: (2018) Moscow, 448 p.).
18. Kozhombierdieva, G.I. (2019) 'Bayesian logical-probabilistic model of fuzzy inference', *Proc. Int. Conf. SCM*, 1, pp. 35–38 (in Russ.).
19. Murphy, K.P. (2022) *Probabilistic Machine Learning: An Introduction*. Mit Press, 864 p. (Russ. ed.: (2022) Moscow, 940 p.).

20. Bishop, C.M. (2006) *Pattern Recognition and Machine Learning*. Springer Publ, 738 p. (Russ. ed.: (2020) St. Petersburg, 940 p.).
21. Jang, J.-S.R. (1993) ‘ANFIS: Adaptive-network-based fuzzy inference system’, *IEEE Transactions on Systems, Man, and Cybernetics*, 23(3), pp. 665–685. doi: 10.1109/21.256541.
22. Averkin, A.N., Yarushev, S.A. (2022) ‘Study of the development of explainable artificial intelligence systems’, *Proc. IMMV-2022*, pp. 127–134 (in Russ.).

Авторы

Хамчиев Георгий Алексеевич¹,
аспирант, gyqn@yandex.ru
Кожомбердиева Гульнара Исмановна¹,
к.т.н., доцент,
kgi-liizht@yandex.ru

¹ Петербургский государственный университет
путей сообщения Императора Александра I,
г. Санкт-Петербург, 190031, Россия

Authors

Georgy A. Khamchichev¹,
Postgraduate Student, gyqn@yandex.ru
Gulnara I. Kozhomberdieva¹,
Cand. of Sci. (Engineering),
Associate Professor, kgi-liizht@yandex.ru

¹ Emperor Alexander I St. Petersburg State
Transport University, St. Petersburg,
190031, Russian Federation