

УДК 519.87:004.032.26

DOI: 10.22213/2410-9304-2020-4-109-116

## Применение методов снижения размерности данных к построению нечетких нейронных сетей\*

*В. А. Тенев*, доктор физико-математических наук, профессор, ИжГТУ имени М. Т. Калашникова; Удмуртский федеральный исследовательский центр УрО РАН, Ижевск, Россия  
*А. С. Шаура*, кандидат технических наук, ИжГТУ имени М. Т. Калашникова; Удмуртский федеральный исследовательский центр УрО РАН, Ижевск, Россия

*Многие задачи интеллектуального анализа данных могут быть сведены к задачам классификации и регрессии, современные подходы к решению которых основаны на применении нейронных сетей, деревьев решений и нечеткой логики, а также классических статистических методов. Решение сложных практических задач состоит из нескольких этапов, таких как выявление и отбор признаков, снижение размерности исходного пространства, поиск взаимосвязей и построение математической модели.*

*В данной работе рассматривается задача снижения размерности данных при построении нечеткой нейронной сети с целью сокращения количества исходных признаков и повышения их информативности. Для ее решения рассмотрены несколько подходов: применение метода главных компонент, нейронной сети с архитектурой автокодировщика и ограниченной машины Больцмана. В отличие от метода главных компонент, применение автокодировщика и ограниченной машины Больцмана позволяет учесть нелинейные связи между имеющимися признаками. Важной особенностью предлагаемой в работе нечеткой нейронной сети, основанной на нечетких деревьях решений, является наглядность представления закономерностей в системе данных в виде логических правил с оценкой важности каждого правила. Применение представленных методов снижения размерности входных данных позволило существенно снизить погрешность аппроксимации. По результатам тестирования лучшей способностью обобщать данные среди рассмотренных методов обладает ограниченная машина Больцмана: ее применение для построения правил в нечеткой нейронной сети снижает погрешность аппроксимации в несколько раз по сравнению с методом главных компонент. Полученные в работе результаты могут быть использованы для построения универсальных нечетких аппроксиматоров при решении задач машинного обучения и анализа данных.*

**Ключевые слова:** снижение размерности, метод главных компонент, нечеткая нейронная сеть, дерево решений, автокодировщик, ограниченная машина Больцмана.

### Введение

Снижение размерности данных является неотъемлемой частью Machine Learning и Data Mining. И не зависимо от того, какие модели будут лежать в основе применяемых подходов: нейронные сети различной архитектуры, деревья решений, нечеткий логический вывод или иные, – важность сжатия размерности исходного пространства признаков обусловлена не только сокращением числа признаков, но и повышением их информативности, решением проблемы мультиколлинеарности и т. д. В связи с этим даже несмотря на то, что осуществляется «сжатие с потерями», в ряде случаев анализ данных в редуцированном пространстве более эффективен, чем в исходном [1].

В сверточных нейронных сетях глубокого обучения функцию сжатия данных выполняют чередующиеся сверточные (*convolution*) и субдискретизирующие (*pooling*) слои. В рамках та-

кого подхода специальная архитектура сети реализует известную операцию свертки, широко применяемую в цифровой обработке сигналов, в частности – в обработке и анализе изображений [2, 3]. Признаки, выделяемые сверточными слоями, являются абстракциями и практически не поддаются какой-либо интерпретации, поэтому область применения сверточных сетей часто ограничена обработкой и распознаванием изображений, где операция свертки является естественной, и не распространяется на анализ и сжатие данных в целом.

Излишне сложная структура нейронных сетей помимо того, что ухудшает их обучаемость, еще и увеличивает вероятность возникновения эффекта переобучения, поэтому даже совместно с классическими нейронными сетями, в частности многослойным персептроном, часто используется предварительное преобразование данных методом главных компонент (МГК) [4, 5]. Такой

© Тенев В. А., Шаура А. С., 2020

\* Работа выполнена в рамках темы НИР УдмФИЦ УрО РАН «Искусственный интеллект в разработке, обучении и сопровождении экспертных систем представления и использования знаний в естественно-научных, технических и социогуманитарных направлениях» АААА-А19-119092690104-4.

подход позволяет уменьшить число нейронов входного слоя и упростить топологию сети, улучшая ее обучаемость и способность к обобщению. Возможности классического МГК ограничены только линейными аппроксимациями исходных данных, что не позволяет учитывать существенно нелинейные связи. Говоря о нелинейном анализе главных компонент (NLPCA – Nonlinear Principal Component Analysis), в отечественных и особенно в зарубежных работах [6], в первую очередь выделяют нейросетевые подходы, в основе которых лежит использование автокодировщиков.

Не менее остро вопрос размерности исходных данных стоит и для деревьев решений. Деревья решений представляют собой широко используемый в последнее время метод классификации, регрессии и анализа данных, основанный на представлении правил в иерархической, последовательной структуре. Для их построения обычно используют методы CART, C4.5, ID3 и др. [7], которые предполагают, что выбор атрибутов для разбиения осуществляется на основе прироста информации. Такой подход далеко не всегда позволяет построить оптимальное дерево – в случае неудачного выбора атрибута уже нет возможности сделать новый выбор, вернувшись на предыдущую итерацию. В результате чего разбиение множества в соответствии с выбранным атрибутом приводит к созданию подмножеств, характеризующихся минимальной статистической ошибкой, и, как следствие, получают излишне детализированные деревья со свойственной им проблемой «переобученности». Для решения этой проблемы необходимо осуществлять регулирование глубины дерева и его усечение. Однако по сравнению с нейронными сетями, которые реализуют модель «черного ящика», не имеют интерпретации и не поддаются объяснению, деревья решений могут быть проанализированы методами булевой алгебры, а каждый путь от корня дерева к его листу может быть записан и интерпретирован как некоторое решающее правило.

Новым этапом в развитии нейронных сетей и деревьев решений стало придание им «нечеткости» и, как следствие, появление нечетких нейронных сетей и нечетких деревьев решений. «Нечеткость» дала моделям большие способности к обобщению и обучению, появилась возможность строить более компактные модели, но проблема снижения размерности и повышения информативности исходных данных осталась.

## Постановка задачи

Имеется набор данных

$$(\mathbf{x}, \mathbf{d})_j, j = \overline{1, N}, \quad (1)$$

представляющий собой таблицу, содержащую  $N$  строк и  $n+m$  столбцов, где  $n, m$  – размерность входного вектора  $\mathbf{x} \in R^n$  и выходного  $\mathbf{d} \in R^m$ . На практике размерность входного вектора может быть большой, часть компонент вектора связаны между собой либо образуют шум. При построении математических моделей, настраиваемых на основе данных (1), желательно повысить информативность входных компонент за счет явного и неявного выявления связей между ними. Этот принцип также закладывается при построении моделей на основе глубокого обучения [8, 9].

Рассмотрим несколько вариантов снижения размерности входного вектора  $\mathbf{x} \in R^n$ . Первый вариант классический – линейное преобразование на основе применения метода главных компонент (МГК). Это преобразование имеет вид

$$\mathbf{z} = \mathbf{W}\mathbf{x}, \quad (2)$$

где  $\mathbf{W} \in R^{M \times n}$ ,  $M \leq n$  – матрица преобразования. Компоненты вектора  $\mathbf{z}$  выстраиваются по степени значимости. Размерность этого вектора определяется отбрасыванием последних незначимых компонент. Считая набор данных (1) случайной выборкой, перейдем от векторов  $\mathbf{x}_j$  к случайным векторам  $\mathbf{u}_j$  с математическим ожиданием, равным нулю:

$$\mathbf{u}_{ij} = \frac{x_{ij} - \bar{x}_i}{\sigma_i}, \quad i = \overline{1, n}, \quad j = \overline{1, N},$$

где

$$\bar{x}_i = \frac{\sum_{j=1}^N x_{ij}}{N}, \quad \sigma_i = \sqrt{\frac{\sum_{j=1}^N (x_{ij} - \bar{x}_i)^2}{N}}.$$

Матрица ковариации формируется на основе векторов  $\mathbf{z}$ :  $\mathbf{Q} = \frac{1}{N} \mathbf{z}\mathbf{z}^T$ . Собственные векторы этой матрицы  $\mathbf{Q}\mathbf{w}_i = \lambda_i \mathbf{w}_i$ ,  $i = \overline{1, n}$  составляют матрицу преобразования  $\mathbf{W} = (\mathbf{w}_i^T) = (w_{ki})$ ,  $k = \overline{1, M}$ ,  $i = \overline{1, n}$  и преобразование (2) распишем в виде

$$z_{kj} = \sum_{i=1}^n w_{ki} x_{ij}. \quad (3)$$

Собственные числа матрицы  $\mathbf{Q}$  расположены по убыванию  $\lambda_1 > \lambda_2 > \dots > \lambda_n$  в соответствии со значимостью компоненты.

Для учета нелинейности связей между составляющими входного вектора воспользуемся способами предварительного обучения глубоких нейронных сетей [10]: автокодировщик (autoencoder) и ограниченная машина Больцмана.

Автокодировщик представляет собой нейронную сеть с тремя слоями:

1. Входной слой содержит вектор входных переменных  $\mathbf{x} \in R^n$ .

2. Скрытый слой содержит  $M \leq n$  нейронов.

3. На выход сети подается входной сигнал  $\mathbf{x}$ .

Обучение автокодировщика производится без учителя, то есть без применения выходных переменных  $\mathbf{d} \in R^m$ . Целью обучения является получение нелинейного преобразования входного вектора в себя самого через скрытый слой с нелинейной функцией активации. Размерность вектора  $\mathbf{x}$  увеличивается еще на единицу, добавляя  $x_0 = 1$ . Также добавляется один нейрон с единичным выходом в скрытый слой. Каждый нейрон скрытого слоя осуществляет суммирование входящих сигналов

$$u_i = \sum_{k=0}^n w_{ik}^1 x_k, \quad i = \overline{1, M}.$$

Для способности обрабатывать нелинейные зависимости выходной сигнал нейрона преобразуется с помощью функции активации

$$z_i = G(u_i), \quad i = \overline{1, M}, \quad (4)$$

где  $M$  – число нейронов в скрытом слое;  $z_0 = 1$ .

В качестве функции активации используется сигмоида

$$G(u) = \frac{1}{1 + \exp(-\beta u)}. \quad (5)$$

Для выходного слоя:

$$y_i = G\left(\sum_{k=0}^M w_{ki}^2 z_k\right), \quad i = \overline{1, n}. \quad (6)$$

После обучения автокодировщика должно выполняться соотношение  $y_i \approx x_i$ ,  $i = \overline{1, n}$ ,  $|\mathbf{x} - \mathbf{y}| \leq \delta \rightarrow 0$ . Обучение проводится методом стохастического градиента с вычислением градиента функции ошибки по алгоритму обратного распространения. Нейроны скрытого слоя

соответствуют нелинейному преобразованию из пространства  $\mathbf{x} \in R^n$  в  $\mathbf{z} \in R^M$  (4):

$$z_i = G\left(\sum_{k=0}^n w_{ik}^1 x_k\right), \quad i = \overline{1, M}.$$

Обратное преобразование следует из (6):

$$x_i = G\left(\sum_{k=0}^M w_{ki}^2 z_k\right), \quad i = \overline{1, n}.$$

Коэффициенты  $w_{ik}^l$ ,  $l = 1, 2$  получены при обучении автокодировщика без учителя.

Ограниченная машина Больцмана (Restricted Boltzmann Machine, RBM) [11] состоит из двух слоев: входного  $\mathbf{x} \in R^n$  и выходного  $\mathbf{z} \in R^m$ . Нейроны двух слоев связаны между собой соотношениями для условных вероятностей:  $p(z_i | \mathbf{x})$ ,  $i = \overline{1, M}$  и  $p(x_k | \mathbf{z})$ ,  $k = \overline{1, n}$ , где мера вероятности определяется сигмоидной функцией (5):

$$p(z_k | \mathbf{x}) = \frac{1}{1 + \exp\left[-\left(\sum_{i=1}^n w_{ik} x_i + A_k\right)\right]}, \quad k = \overline{1, M},$$

$$p(x_i | \mathbf{z}) = \frac{1}{1 + \exp\left[\left(\sum_{i=1}^M w_{ik} z_k + B_i\right)\right]}, \quad i = \overline{1, n}.$$

При обучении RBM для нахождения коэффициентов  $w_{ik}$ ,  $A_k$ ,  $B_i$ ,  $i = \overline{1, n}$ ,  $k = \overline{1, M}$  применяется метод контрастной дивергенции, основанный на сэмпировании Гиббса [12]. Сэмпирование предполагает несколько шагов определения значений  $z_j$  на скрытом слое с реконструкцией по ним величин  $x_i$  на входном слое. Вычисления показывают, что достаточно одного шага. Первоначально коэффициенты  $w_{ik}$ ,  $A_k$ ,  $B_i$ ,  $i = \overline{1, n}$ ,  $k = \overline{1, M}$  задаются случайным образом с последующей корректировкой итерационным методом. На вход подается входной вектор  $\mathbf{x}^{(0)} = (x_i)_j$ , вычисляется  $z_k^{(0)} = G\left(\sum_{i=1}^n w_{ik} x_i^{(0)} + A_k\right)$  при  $\beta = 1$ , затем реконструируется  $x_i^{(1)} = G\left(\sum_{i=1}^M w_{ik} z_k^{(0)} + A_i\right)$  и вычисляется  $z_k^{(1)} = G\left(\sum_{i=1}^n w_{ik} x_i^{(1)} + A_k\right)$ . Корректировка проводится по схеме:

$$w_{ik} \rightarrow w_{ik} + \eta \left( x_i^{(0)} z_k^{(0)} - x_i^{(1)} z_k^{(1)} \right),$$

$$A_k \rightarrow A_k + \eta \left( z_k^{(0)} - z_k^{(1)} \right),$$

$$B_i \rightarrow B_i + \eta \left( x_i^{(0)} - x_i^{(1)} \right).$$

Обучение также проводится без учителя на имеющейся выборке. Нелинейное преобразование из пространства  $\mathbf{x} \in R^n$  в  $\mathbf{z} \in R^M$  имеет вид

$$\mathbf{z} = \mathbf{z}^{(1)} = G(\mathbf{x}^{(1)}).$$

### Построение нечетких нейронных сетей

Нечеткие нейронные сети основаны на сочетании правил нечеткого логического вывода и нейросетевой структуры. Известен вариант нечеткой сети Такаги – Сугено – Канга (TSK) [13].

Правила нечеткого логического вывода имеют вид:

$$R_r: \text{ if } \bigcap_i x_i \in A_{ir} \text{ then } y \text{ is } B_r, \quad r = \overline{1, K_R}, \quad (7)$$

где  $A_{ir}, B_r$  – нечеткие подмножества;  $K_R$  – количество правил.

$$\mu^+(x_i) = \begin{cases} \frac{1}{1 + \exp(-\beta(x_i - v_{ik}))} \left( 1 + \exp(-\beta(1 - v_{ik})) \frac{x_i - v_{ik}}{1 - v_{ik}} \right) & | x_i \geq v_{ik}, \\ \frac{1}{1 + \exp(-\beta(x_i - v_{ik}))} v_{ik} & | x_i < v_{ik}, \end{cases}$$

$$\mu^-(x_i) = 1 - \mu^+(x_i).$$

Операции конъюнкции соответствует логическое произведение. Степени истинности  $\alpha_r, r = \overline{1, K_R}$  соответствуют значениям функций принадлежности левых частей правил:  $\alpha_r = \min_k (\mu_k^\pm)$ .

Дальнейшая модификация нечеткой нейронной сети предлагается с заменой линейного полинома в правой части правил на суперпозицию радиальных функций:

$$y_r(\mathbf{x}) = \sum_{h=1}^H w_{hr} \phi_h(\mathbf{x}) = \sum_{h=1}^H w_{hr} \exp\left(-\frac{\|\mathbf{x} - \mathbf{c}_h\|}{\sigma_h}\right).$$

Операция агрегирования правил представляется следующим выражением:

В сети TSK функции принадлежности к подмножествам  $A_{ir}$  заданы в виде обобщенной функции Гаусса

$$\mu_{Ar}(x_i) = \frac{1}{1 + \left(\frac{x_i - c_r}{\sigma_r}\right)^{2b_r}},$$

где  $c_r, b_r, \sigma_r$  – параметры функций. Операция конъюнкции в левой части правил определяется алгебраическим произведением функций принадлежности. Вывод в правой части правила осуществляется по способу Сугено с линейной функцией. Дальнейшим развитием сети TSK стала нечеткая нейронная сеть, основанная на нечетких деревьях решений [14]. Логические правила (7) представляют собой условия прохождения по графу построенного дерева от точки входа до конечного листа с номером интервала для аппроксимируемой функции. Условие  $x_i \in A_{ir}$  соответствует условию разделения множества объектов  $x_i \begin{matrix} > \\ < \end{matrix} (v_{ik}), i = \overline{1, n}, k = \overline{1, q}$ , по признаку  $v_{ik}$  и означает попадание величины  $x_i$  в нечеткий интервал с функциями принадлежности:

$$y(\mathbf{x}) = \frac{1}{\sum_{r=1}^{K_R} \alpha_r} \sum_{r=1}^{K_R} \alpha_r y_r.$$

Обучение данной нечеткой сети проходит два этапа. На первом с применением алгоритма классификации на основе деревьев решений определяется набор правил (7) и величины  $\overline{\alpha_r} = \frac{\alpha_r}{\sum_{r=1}^{K_R} \alpha_r}$  (обучение с учителем). На втором

этапе определяются коэффициенты  $w_{hr}$  с применением операции псевдоинверсии  $\mathbf{G}^+ = (\mathbf{G}^T \mathbf{G})^{-1} \mathbf{G}^T$ , где матрица  $\mathbf{G} = [\overline{\alpha_r} \phi_h(\mathbf{x}_j)]$ ,  $h = \overline{1, H}, j = \overline{1, N}$  и  $\mathbf{w} = (w_{hr}) = \mathbf{G}^+ \mathbf{d}$  (обучение с учителем).

Положительной стороной нечеткой сети подобного типа является наглядность представления закономерностей в системе данных в виде логических правил с оценкой важности каждого правила. Отрицательной стороной является увеличение сложности деревьев при высокой размерности данных, приводящее к увеличению количества коэффициентов  $w_{hr}$  и снижению точности аппроксимации. В связи с этим целесообразно применить рассмотренные методы снижения размерности входных данных при построении нечеткой нейронной сети.

Нечеткую нейронную сеть построим следующим образом:

– правила определяются по набору данных  $(\mathbf{z}, \mathbf{d})_j$ ,  $j = \overline{1, N}$

$$f(\mathbf{x}) = \sum_{i=1}^{20} \sin(2\pi x_i) + 3(1-x_1)^2 \exp(-x_1^2 - (1+x_2)^2) - 10(0.2x_1 - x_1^3 - x_2^5) \exp(-x_1^2 - x_2^2) - \frac{1}{3} \exp(-(1+x_1)^2 - x_2^2) + 3(1-x_9)^2 \exp(-x_9^2 - (1+x_{10})^2) - 10(0.2x_9 - x_9^3 - x_{10}^5) \exp(-x_9^2 - x_{10}^2) - \frac{1}{3} \exp(-(1+x_9)^2 - x_{10}^2),$$

$$x_{i+4} = (x_1)^{\frac{i}{i+1}}, \quad x_{i+12} = (x_1)^{\frac{0.5i}{i+1}}, \quad i = \overline{1, 8}, \quad -3 \leq x_i \leq 3.$$

Часть переменных независимы, часть выражается через другие переменные. Из этих данных получены переменные отображения на пространство меньшей размерности:

$\mathbf{z}_{PCA}$  – соответствуют методу главных компонент;

$\mathbf{z}_{AE}$  – нейроны автокодировщика;

$$R_r: \text{if } \bigcap_i z_i \in A_{ir} \text{ then } y \text{ is } B_r, \quad r = \overline{1, K_R};$$

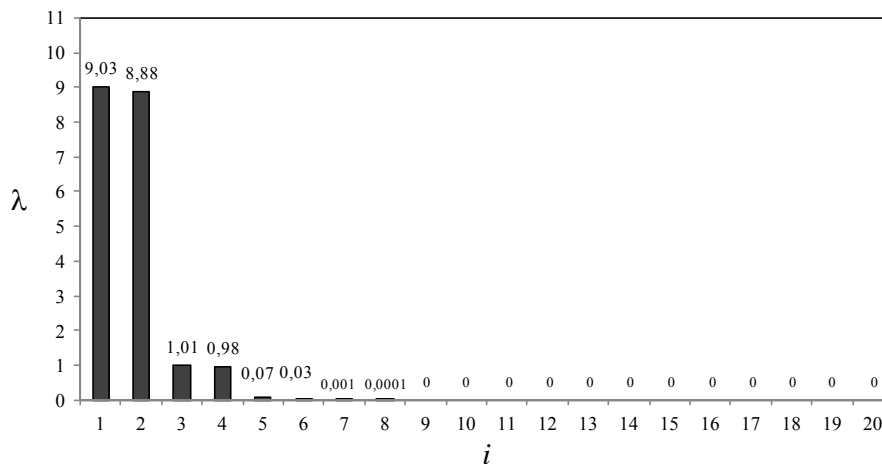
– выход рассчитывается по всем переменным  $\mathbf{x}$  с применением радиальных функций  $\varphi_h(\mathbf{x})$ ,  $h = \overline{1, H}$ .

Для подбора параметров деревьев решений  $(N^{\min}, q, K_y, \beta)$ , где  $N^{\min}$  – минимальный размер разделяемого множества;  $q$  – количество пороговых значений  $v_{ik}$  для переменных  $\mathbf{z}$ ;  $K_y$  – количество классов для выходного значения функции;  $\beta$  – параметр функции принадлежности, применялся гибридный генетический алгоритм [15].

Рассмотрим тестовый набор данных с  $n = 20$ ,  $N = 1000$ , задаваемый функцией

$\mathbf{z}_{RBM}$  – скрытый слой ограниченной машины Больцмана.

Размерность этого пространства выбрана  $M = 6$  из условия минимальной погрешности и полного сохранения информации по методу главных компонент. На рисунке приведены значения собственных чисел, соответствующих номеру компоненты.



Значения собственных чисел, соответствующих номеру компоненты

Наибольшую информативность имеют первые две компоненты, 8-я компонента характеризуется значением собственного числа 0,00012. Все остальные компоненты не содержат никакой информации.

В таблице приведены параметры деревьев решений:  $N^{\min}$ ,  $q$ ,  $K_y$ ;  $K_{R0}$  – количество ветвей неупорядоченного дерева;  $K_R$  – количество ветвей упорядоченного дерева (правил);  $\delta_{\text{learn}}$ ,  $\delta_{\text{test}}$  – погрешности на обучающей выборке (900 точек) и тестовой (100 точек).

### Результаты аппроксимации

Метод	$N^{\min}$	$q$	$K_y$	$K_{R0}$	$K_R$	$\delta_{\text{learn}}$ , %	$\delta_{\text{test}}$ , %
Без снижения	108	5	23	19	10	5,23	7,64
РСА	151	5	21	25	13	6,13	7,15
АЕ	102	5	25	29	15	2,46	2,78
RBM	100	5	25	25	13	2,2	2,38

Первая строка, содержащая числовые показатели, соответствует аппроксимации функциональной зависимости без применения сжатия входных данных. Ошибка на тестовой выборке является довольно большой – 7,64 %. Применение первых шести компонент незначительно улучшает качество аппроксимации до 7,15 %. Это объясняется тем, что многие переменные связаны между собой существенно нелинейными связями. Учет нелинейности учитывается в автокодировщике с сигмоидной функцией активации. Ошибка аппроксимации уменьшается до 2,78 %. Наилучший результат показывает применение ограниченной машины Больцмана. Ошибка аппроксимации равна 2,38 %. Следует отметить, что применение RBM по схеме  $20 \rightarrow 12 \rightarrow 6$  не дает видимых отличий на результат построения нечеткой нейронной сети, основанной на деревьях решений, от схемы  $20 \rightarrow 6$ . Применение традиционного нечеткого вывода в виде полинома первой степени дает несколько большую ошибку 2,78 % для RBM и 3,61 % для автокодировщика.

### Заключение

Попытки анализа «сырых» данных приводят к избыточной сложности моделей, худшей обучаемости и, как следствие, большей ошибке аппроксимации. Предварительный анализ данных позволяет избежать этих проблем. Даже при использовании современных подходов глубокого обучения целесообразна предварительная обработка данных для выявления (*feature detec-*

*tion*) и отбора признаков (*feature selection*). В общем случае, когда нет каких-либо сведений о характере имеющихся зависимостей, их линейности и т. д., имеет смысл отдавать предпочтение нелинейным методам сжатия размерности.

Сравнительный анализ методов снижения размерности системы данных показал преимущество в поиске закономерностей среди массива данных нелинейных методов по сравнению с методом главных компонент.

Лучшей способностью обобщать данные среди рассмотренных нелинейных методов обладает ограниченная машина Больцмана. Применение RBM для построения правил в нечеткой нейронной сети снижает погрешность аппроксимации в 3 раза. Применение нечеткого вывода с применением радиальных функций вместо линейного полинома также улучшает качество аппроксимации. Количество радиальных функций может быть меньше размерности входного вектора, что снижает количество неизвестных коэффициентов  $w_{hr}$ .

### Библиографические ссылки

1. Antonio Rico-Sulayes. Reducing Vector Space Dimensionality in Automatic Classification for Authorship Attribution // Revista Ingeniería Electrónica, Automática y Comunicaciones. — 2017. — Т. 38, № 3.
2. Sakshi Indolia, Anil Kumar Goswami, S.P. Mishra, Pooja Asopa, Conceptual Understanding of Convolutional Neural Network - A Deep Learning Approach, Procedia Computer Science, Volume 132, 2018, Pages 679-688, ISSN 1877-0509, <https://doi.org/10.1016/j.procs.2018.05.069>.
3. Rawat, Waseem & Wang, Zenghui. (2017). Deep Convolutional Neural Networks for Image Classification: A Comprehensive Review. Neural Computation. 29. 1-98. 10.1162/NECO\_a\_00990.
4. Mostaar A, Sattari MR, Hosseini S, Deevband MR. Use of Artificial Neural Networks and PCA to Predict Results of Infertility Treatment in the ICSI Method. J Biomed Phys Eng. 2019;9(6): 679-686. Published 2019 Dec 1. doi:10.31661/jbpe.v0i0.1187
5. M. Seuret, M. Alberti, M. Liwicki and R. Ingold, "PCA-Initialized Deep Neural Networks Applied to Document Image Analysis," 2017 14th IAPR International Conference on Document Analysis and Recognition (ICDAR), Kyoto, 2017, pp. 877-882. doi: 10.1109/ICDAR.2017.148
6. Kruger U., Zhang J., Xie L. (2008) Developments and Applications of Nonlinear Principal Component Analysis – a Review. In: Gorban A.N., Kégl B., Wunsch D.C., Zinovyev A.Y. (eds) Principal Manifolds for Data Visualization and Dimension Reduction. Lecture Notes in Computational Science and Engineering, vol 58. Springer, Berlin, Heidelberg. [https://doi.org/10.1007/978-3-540-73750-6\\_1](https://doi.org/10.1007/978-3-540-73750-6_1)

7. Classification and regression trees // L. Breiman, J. H. Friedman, R. A. Olshen, C. J. Stone. – California: Wadsworth & Brooks, 1984. – 368 p.

8. Antonio Hernández-Blanco, Boris Herrera-Flores, David Tomás, Borja Navarro-Colorado, "A Systematic Review of Deep Learning Approaches to Educational Data Mining", *Complexity*, vol. 2019, Article ID 1306039, 22 pages, 2019. <https://doi.org/10.1155/2019/1306039>

9. Emmert-Streib F, Yang Z, Feng H, Tripathi S and Dehmer M (2020) An Introductory Review of Deep Learning for Prediction Models With Big Data. *Front. Artif. Intell.* 3:4. doi: 10.3389/frai.2020.00004

10. Созыккин А. В. Обзор методов обучения глубоких нейронных сетей // Вестник ЮУрГУ. Серия: Вычислительная математика и информатика. 2017. Т. 6, № 3. С. 28–59. DOI: 10.14529/cmse170303.

11. Salakhutdinov, R. & Hinton, G. (2009). Deep Boltzmann Machines. Proceedings of the Twelfth International Conference on Artificial Intelligence and Statistics, in PMLR5:448-455

12. Hinton G.E., Osindero S., Teh Y.-W. A Fast Learning Algorithm for Deep Belief Nets. *Neural Computing*. 2006. vol. 18, no. 7. pp. 1527–1554. DOI: 10.1162/neco.2006.18.7.1527.

13. Осовский, Станислав. Нейронные сети для обработки информации / Станислав Осовский ; пер. с пол. И. Д. Рудинского. М. : Финансы и статистика, 2004 (Великолук. гор. тип.). 43 с. : ил.; 24 см. ISBN 5-279-02567-4 (в обл.)

14. Тенев В. А., Ворончак В. И. Решение задач классификации и аппроксимации с применением нечетких деревьев решений // Интеллектуальные системы в производстве. 2005. № 2. С. 46–54.

15. Тенев В. А., Тенева А. В. Обучение нечетких нейронных сетей генетическим алгоритмом // Интеллектуальные системы в производстве. 2010. № 1. С. 76–85.

## References

1. Antonio Rico-Sulayes. Reducing Vector Space Dimensionality in Automatic Classification for Authorship Attribution // *Revista Ingeniería Electrónica, Automática y Comunicaciones*. — 2017. — Т. 38, № 3.

2. Sakshi Indolia, Anil Kumar Goswami, S.P. Mishra, Pooja Asopa, Conceptual Understanding of Convolutional Neural Network - A Deep Learning Approach, *Procedia Computer Science*, Volume 132, 2018, Pages 679-688, ISSN 1877-0509, <https://doi.org/10.1016/j.procs.2018.05.069>.

3. Rawat, Waseem & Wang, Zenghui. (2017). Deep Convolutional Neural Networks for Image Classification: A Comprehensive Review. *Neural Computation*. 29. 1-98. 10.1162/NECO\_a\_00990.

4. Mostaar A, Sattari MR, Hosseini S, Deevband MR. Use of Artificial Neural Networks and PCA

to Predict Results of Infertility Treatment in the ICSI Method. *J Biomed Phys Eng*. 2019;9(6): 679-686. Published 2019 Dec 1. doi:10.31661/jbpe.v0i0.1187

5. M. Seuret, M. Alberti, M. Liwicki and R. Ingold, "PCA-Initialized Deep Neural Networks Applied to Document Image Analysis," 2017 14th IAPR International Conference on Document Analysis and Recognition (ICDAR), Kyoto, 2017, pp. 877-882. doi: 10.1109/ICDAR.2017.148

6. Kruger U., Zhang J., Xie L. (2008) Developments and Applications of Nonlinear Principal Component Analysis – a Review. In: Gorban A.N., Kégl B., Wunsch D.C., Zinovyev A.Y. (eds) *Principal Manifolds for Data Visualization and Dimension Reduction. Lecture Notes in Computational Science and Engineering*, vol 58. Springer, Berlin, Heidelberg. [https://doi.org/10.1007/978-3-540-73750-6\\_1](https://doi.org/10.1007/978-3-540-73750-6_1)

7. Classification and regression trees // L. Breiman, J. H. Friedman, R. A. Olshen, C. J. Stone. – California: Wadsworth & Brooks, 1984. – 368 p.

8. Antonio Hernández-Blanco, Boris Herrera-Flores, David Tomás, Borja Navarro-Colorado, "A Systematic Review of Deep Learning Approaches to Educational Data Mining", *Complexity*, vol. 2019, Article ID 1306039, 22 pages, 2019. <https://doi.org/10.1155/2019/1306039>

9. Emmert-Streib F, Yang Z, Feng H, Tripathi S and Dehmer M (2020) An Introductory Review of Deep Learning for Prediction Models With Big Data. *Front. Artif. Intell.* 3:4. doi: 10.3389/frai.2020.00004

10. Sozykin A.V. [Review of methods for teaching deep neural networks]. *Vestnik YUURGU. Seriya: Vychislitel'naya matematika i informatika*, 2017, vol. 6, no. 3, pp. 28–59. DOI: 10.14529/cmse170303. (in Russ.)

11. Salakhutdinov, R. & Hinton, G. (2009). Deep Boltzmann Machines. Proceedings of the Twelfth International Conference on Artificial Intelligence and Statistics, in PMLR5:448-455.

12. Hinton G.E., Osindero S., Teh Y.-W. A Fast Learning Algorithm for Deep Belief Nets. *Neural Computing*. 2006. vol. 18, no. 7. pp. 1527–1554. DOI: 10.1162/neco.2006.18.7.1527.

13. Osovsky S. Neural networks for information processing. Per. from Poland. / Stanislav Osovskiy. Moscow: Finance and statistics, 2004, ISBN 5-279-02567-4.

14. Tenenev V.A., Voronchak V.I. [Solving classification and approximation problems using fuzzy decision trees]. *Intellektual'nyye sistemy v proizvodstve*, 2005, no. 2, pp. 46-54. (in Russ.)

15. Tenenev V.A., Teneneva A.V. [Training fuzzy neural networks with a genetic algorithm]. *Intellektual'nyye sistemy v proizvodstve*, 2010, no. 1, pp. 76-85. (in Russ.)

\*\*\*

### Application of Data Dimensionality Reduction Methods to the Construction of Fuzzy Neural Networks

V. A. Tenenev, DSc (Physics and Mathematics), Kalashnikov ISTU, Udmurt Federal Research Center of the Ural Branch of the Russian Academy of Sciences, Izhevsk, Russia

A. S. Shaura, PhD in Engineering, Udmurt Federal Research Center of the Ural Branch of the Russian Academy of Sciences, Izhevsk, Russia

*Many data mining problems can be reduced to classification and regression problems; modern approaches to their solving are based on the use of neural networks, decision trees, fuzzy logic, and classical statistical methods. The solution to complex practical problems consists of several stages, such as feature detection and feature selection, reducing the initial space's dimension, searching for relationships, and building a mathematical model.*

*The paper considers data dimension reduction when constructing a fuzzy neural network to reduce the number of initial features and increase their informativeness. We consider several approaches to solve it: application of the principal component analysis (PCA), a neural network with an autoencoder architecture, and a restricted Boltzmann machine. In contrast to the method of principal components, the use of an autoencoder and a limited Boltzmann machine makes it possible to take into account the nonlinear connections between the existing features. An important feature of the proposed fuzzy neural network based on fuzzy decision trees is the visibility of the presentation of relationships in the data system in the form of logical rules to assess the importance of each rule. Using the presented methods for reducing the input data dimension allowed us to reduce the approximation error significantly. According to the test results, the best ability to generalize data among the considered methods is possessed by the restricted Boltzmann machine: its application for constructing rules in a fuzzy neural network reduces the approximation error several times compared with the principal component analysis. The results obtained in this work can be used to construct universal fuzzy approximators for solving machine learning and data analysis problems.*

**Keywords:** data reduction, principal component analysis, fuzzy neural network, decision tree, autoencoder, restricted Boltzmann machine.

Получено: 21.09.2020