КОМПЬЮТЕРНЫЕ НАУКИ 2021. Т. 31. Вып. 1. С. 149–164.

УДК 004.89

© *А. Д. Обухов, М. Н. Краснянский*

НЕЙРОСЕТЕВОЙ МЕТОД ОБРАБОТКИ И ПЕРЕДАЧИ ДАННЫХ В АДАПТИВНЫХ ИНФОРМАЦИОННЫХ СИСТЕМАХ

Рассматривается проблема автоматизации процессов передачи и обработки информации в адаптивных информационных системах. Анализ существующих подходов к решению данной задачи показал перспективность использования нейросетевых технологий. Сформулирован нейросетевой метод обработки и передачи информации в адаптивных информационных системах. Метод включает формализованное описание нейросетевого канала данных — программного инструмента для анализа, обработки данных и выбора протокола передачи данных. Изложены основные этапы предлагаемого метода: классификация структур исходных данных, их преобразование, обработка данных, выбор необходимого протокола передачи информации. Каждый из этапов реализуется посредством нейронных сетей различной архитектуры. Приведено теоретическое обоснование возможности применения нейросетевого метода, полученное на основе доказательства ряда теорем. Новизна предлагаемого метода заключается в переходе от аналитического решения задач классификации, обработки и передачи данных к автоматизированному подходу с применением технологий машинного обучения. Практическая значимость нейросетевого метода состоит в снижении сложности реализации процессов обработки и передачи информации, повышении уровня автоматизации при организации межмодульного взаимодействия. Проведена оценка реализации нейросетевого метода по ряду метрик оценки сложности программного обеспечения. Проанализированы область применения, достоинства и недостатки разработанного метода.

Ключевые слова: нейросетевой метод обработки и передачи информации, нейросетевой канал данных, нейронные сети, адаптивные информационные системы.

DOI: 10.35634/vm210111

Активное развитие технологий искусственного интеллекта и производительности вычислительного оборудования, появление новых высокоуровневых программных инструментов и библиотек позволяет выявить новые закономерности и системные связи, автоматизировать процессы управления, осуществлять обработку и передачу информации с большей эффективностью. В адаптивных информационных системах (АИС) эта тенденция приводит к повышению функциональности, стабильности, расширению возможностей по персонализации и модификации в зависимости от состояния внешней среды.

Разработка современных АИС предъявляет к коллективу разработчиков серьезные требования, появляется необходимость в привлечении экспертов предметной области и дата-инженеров. Если программный продукт не будет обладать достаточно высокой адаптируемостью и гибкостью, то он не сможет стать конкурентоспособным. Однако выполнение этих условий приводит к повышению сложности разработки и, следовательно, ее стоимости и временны́х затрат.

Важной и актуальной проблемой при проектировании АИС является организация межмодульного взаимодействия, проверка соответствия типов данных, передаваемых между компонентами системами. Именно на этом этапе проектирования АИС чаще всего допускаются ошибки. Использование различного рода стандартов (например, микросервисной архитектуры и формата JSON) позволяет снизить вероятность формирования ошибочной структуры данных, однако полностью исключить человеческий фактор не может. Поэтому

актуальной задачей является автоматизация процесса обработки передачи данных между модулями с целью снижения негативного фактора человеческой ошибки [1].

Проведенный анализ существующих подходов показал, что процессы передачи данных сводятся к применению различных протоколов (сетевых, транспортных, прикладных и т. д. [2,3]). Их выбор и применение невозможны без участия и контроля человека. Для автоматизации данных процессов необходима разработка систем поддержки принятия решений, позволяющих снизить степень участия и влияния человека, вероятность ошибки при формировании данных [4]. В основе подобных систем необходимо использование современных технологий искусственного интеллекта, например нейронных сетей, неоднократно доказывающих высокую эффективность при решении задач такого рода [5–7]. Методы машинного обучения активно применяются для анализа, обработки, генерации, классификации информации, прогнозирования событий, поиска новых закономерностей и так далее [8–11].

Оценивая возможность автоматизации процессов обработки данных, можно выделить два основных подхода. Первый (классический) основан на аналитическом решении задачи обработки данных, реализуемом в виде программных функций, процедур, библиотек и т. д. Для успешной реализации классического подхода требуется формализация алгоритма процесса обработки данных группой разработчиков, возможно, с привлечением экспертов в предметной области. Далее полученный алгоритм реализуется на выбранном языке программирования [12]. Второй подход связан с использованием интеллектуальных технологий, начиная от экспертных систем и заканчивая методами машинного обучения. В последнем случае разработчики могут самостоятельно реализовать процедуру обработки данных, не углубляясь в алгоритмы их трансформации. Машинное обучение в настоящее время позволяет реализовать ряд процедур преобразования данных с высокой точностью, обеспечивая лучшие показатели, чем классические подходы [13–16]. Поэтому перспективным направлением является исследование возможности применения нейросетевых технологий для автоматизации обработки и передачи данных в АИС путем обобщения и объединения теоретических и практических подходов в новом методе.

Предметом исследования в данной работе являются нейросетевые каналы данных, используемые для автоматизации обработки и передачи данных. На основе представленной ранее концепции [17] осуществлена разработка нейросетевого метода обработки и передачи данных, включающего формализацию структуры нейросетевых каналов данных, этапы их реализации, классификацию по степени сложности. Теоретическое обоснование возможности применения метода получено на основе формулирования и доказательства ряда теорем. Для практической реализации метода представлена оценка сложности программного обеспечения. Реализация нейросетевого метода осуществлена в рамках общей нейросетевой архитектуры информационных систем [17].

§ 1. Нейросетевой метод обработки и передачи данных

Сформулируем следующую задачу: необходимо осуществить передачу данных X, имеющих структуру SX, из i-го модуля АИС в j-й с возможностью осуществления преобразования $X \to Y$, где Y — ожидаемый в j-м модуле набор данных со структурой SY. Для решения поставленной задачи предлагается нейросетевой метод обработки и передачи данных, основанный на применении нейронных сетей для анализа, обработки информации и выбора способов ее передачи, отличающийся комплексным теоретическим обоснованием реализации и применения нейросетевых каналов данных для автоматизированной организации межмодульного взаимодействия.

Нейросетевой метод обработки и передачи данных заключается в получении по сетевому протоколу CP выходного вектора Y со структурой данных SY на основе преобразования входного вектора X со структурой данных SX. Принцип работы метода основан на применении нейросетевых каналов данных различной степени сложности. Дадим основные определения.

Определение 1. *Нейросетевой канал данных* (Neural Network Data Channel, NNDC) — программный интерфейс, осуществляющий обработку и передачу данных между модулями АИС и включающий следующие элементы: входные и выходные данные, их структуры, искусственные нейронные сети для анализа и преобразования данных, выбора протоколов передачи данных, программные библиотеки преобразования и передачи данных.

Определение 2. Под *степенью сложности* (NNS) нейросетевого канала данных будем понимать наличие нейронных сетей для выполнения соответствующих операций анализа (NNS = 1), обработки (NNS = 2) и передачи информации (NNS = 3). Допускается, что степень сложности может быть нулевой.

Определение 3. *Категории* (type) нейросетевых каналов данных определяют их функциональность, т. е. перечень задач, решаемых в рамках каждой конкретной категории.

В общем виде процесс обработки и передачи данных в $NNDC$ можно формализовать следующим соотношением:

$$NNDC_{i,j}^{type}(STR_{SX}(X), SY) = STR_{SY}(Y),$$

где $NNDC_{i,j}^{type}$ — нейросетевой канал данных категории $type$, осуществляющий взаимодействие между i-м и j-м модулями АИС; $X = x_n|n = \overline{1, xN}$ — данные размерностью xN со структурой данных SX, поступающие на вход $NNDC$ от i-го модуля; $Y = y_m|m = \overline{1, yM}$ — данные размерностью yM со структурой данных SY, отправляемые на выход $NNDC$ и ожидаемые j-м модулем. Примем обозначение $STR_{SX}(X)$ как выполнение операции приведения данных X к структуре вида SX. Аналогично: $STR_{SY}(Y)$ — приведение данных Y к структуре вида SY.

Рассмотрим реализацию нейросетевого метода обработки и передачи информации, которая включает следующие этапы.

1. Формирование общей структуры нейросетевого канала данных:

$$NNDC = TR(CP_q) \cup a_1 NN^{ct} \cup a_2 NN^{proc} \cup a_3 NN^{tr}, \tag{1.1}$$

   где TR — процедура передачи данных по выбранному сетевому протоколу $CP_q \in CP$ (CP — множество всех сетевых протоколов),

   NN^{ct} — нейронная сеть классификации типов данных,

   NN^{proc} — нейронная сеть обработки и преобразования данных,

   NN^{tr} — нейронная сеть выбора протоколов передачи данных,

   a_i — коэффициенты, определяющие степень сложности нейросетевого канала данных, $i = \overline{1, 3}$.

2. Анализ структур входных и выходных данных.

Нейросетевой канал должен обеспечить соответствие между структурами входных и выходных данных. Для выполнения этого условия необходимо определить структуры SX и SY входных и выходных данных соответственно, используя нейронную сеть классификации типов данных NN^{ct}:

$$NN^{ct}(X) = SX, \quad NN^{ct}(Y) = SY. \tag{1.2}$$

3. Определение категории и степени сложности нейросетевого канала данных.

   Канал $NNDC_{i,j}^{type}$ устанавливает связь между модулями i и j. Если структуры и значения входных и выходных данных совпадают, а сетевой протокол CP_q заранее определен, то связь вырождается в нейросетевой канал данных нулевой степени. В противном случае требуется повышение степени нейросетевого канала для решения конкретных задач анализа, обработки и передачи данных.

   На основе анализа отображения $X \to Y$ и $SX \to SY$ определяется категория нейросетевого канала, определяющая перечень задач по обработке и передаче данных. Данный процесс можно осуществить аналитически или автоматически с помощью различных систем поддержки принятия решений (в том числе на базе нейронных сетей).

   Примерами категорий нейросетевых каналов данных являются: канал безопасности (на основе уровня доступа пользователя формирует множество доступных функций, модулей и областей данных), канал управления (каждому действию пользователя ставит в соответствие множество операций и событий в АИС), канал адаптации (на основе программного и аппаратного обеспечения пользователя и состояния АИС формирует структуру и параметры интерфейса), канал передачи данных (осуществляет анализ, преобразование и передачу информации между двумя модулями), канал отображения данных (преобразует информацию на основе параметров интерфейса), запроса данных (для каждого события осуществляет необходимый запрос к хранилищу данных) и т. д.

4. Преобразование структур данных.

   Если $SX \neq SY$, то необходимо выполнить преобразование структур данных. В рамках данного метода предлагается реализация программной библиотеки преобразования структур. Далее на основе результата функции сравнения структур данных C_{STR} выбирается функция преобразования $FSTR_l$:

$$C_{STR}(SX, SY) \to FSTR_l, \tag{1.3}$$

$$FSTR_l : STR_{SX}(X) \to STR_{SY}(X). \tag{1.4}$$

5. Обработка данных.

   Если $X \neq Y$, то необходимо осуществить обработку данных. Для этого используется нейронная сеть $NN_{i,j,type}^{proc}$:

$$NN_{i,j,type}^{proc} : STR_{SY}(X) \to STR_{SY}(Y). \tag{1.5}$$

   Процесс обработки данных зависит от категории нейросетевого канала. Примерами такой обработки могут быть регрессия, классификация, кластеризация, восстановление потерянных данных, генерация новой информации по заданным условиям и т. д.

6. Передача данных.

   Для автоматизации выбора сетевого протокола передачи данных CP_q необходимо использование нейронной сети $NN^{tr}_{i,j,type}$:

$$NN^{tr}_{i,j,type} : (SX, SY) \to CP_q. \tag{1.6}$$

   Для нейросетевых каналов данных второй степени сложности или ниже сетевой протокол данных зафиксирован или выбирается аналитически. Выбранный протокол CP_q используется для передачи данных в процедуре TR.

Таким образом, изложенный нейросетевой метод обработки и передачи данных реализует весь цикл анализа, преобразования и передачи информации в АИС. Использование нейронных сетей на всех этапах, требующих привлечения человека для осуществления анализа и принятия решений, позволяет автоматизировать процедуру обработки и передачи информации, снизить негативное влияние человеческого фактора на процесс организации взаимодействия между компонентами АИС.

§ 2. Теоретическое обоснование нейросетевого метода обработки и передачи данных

Рассмотрим ряд теорем, доказательство которых позволяет теоретически обосновать возможность реализации и применения нейросетевого метода обработки и передачи данных.

Теорема 1. *Нейросетевой канал данных произвольной степени сложности, в том числе нулевой, для любых i-го и j-го компонентов АИС позволяет осуществить процедуру передачи данных TR между этими компонентами при условии, что существует сетевой протокол CP_q, программная реализация которого реализует передачу данных между i-м и j-м компонентами.*

Доказательство. В соответствии с выражением (1.1) и определением 1 процедура передачи данных в нейросетевых каналах данных любой степени сложности (в том числе и нулевой) основана на применении существующих сетевых протоколов CP_q в рамках процедуры передачи данных TR.

Рассмотрим вариант, когда степень сложности $NNS = 0$. В этом случае $a_1 = a_2 = a_3 = 0$, следовательно, $NNDC = TR(CP_q)$. Так как TR реализует передачу данных произвольной размерности и типа между модулями по существующему сетевому протоколу CP_q, то нейросетевой канал нулевой степени сложности сводится к программной реализации данного сетевого протокола передачи данных.

Предположим, что степень сложности $NNS \neq 0$, следовательно, некоторое a_i (хотя бы одно) является ненулевым: $a_i \neq 0$. Тогда функциональность нейросетевого канала данных будет включать процедуру передачи данных и некоторые операции над данными, выполняемые с использованием нейронных сетей. Это не противоречит исходному утверждению. Аналогично для любого количества a_i. Доказательство теоремы 1 подтверждает эквивалентность нейросетевого канала данных произвольной степени сложности и программной реализации сетевого протокола при решении задач передачи данных. □

Следствие 1. *Если существует процедура передачи данных TR между i-м и j-м компонентами АИС и она реализуется средствами некоторого сетевого протокола CP_q, то в рамках нейросетевой архитектуры такая процедура может быть выражена как нейросетевой канал данных нулевой степени сложности.*

Доказательство. Из теоремы 1 следует, что нейросетевой канал нулевой степени соответствует программной реализации процедуры передачи данных: $NNDC = TR$; значит, верно и обратное: $TR = NNDC$, если степень канала $NNS = 0$. Доказательство следствия 1 позволяет сделать вывод, что в частном случае возможно организовать взаимодействие модулей в нейросетевой архитектуре на основе только сетевых протоколов передачи данных, представляя их как нейросетевые каналы нулевой степени. Таким образом, нейросетевой канал нулевой степени и процедура передачи данных по сетевому протоколу в рамках нейросетевой архитектуры эквивалентны и взаимозаменяемы. □

Определение 4. *Нейросетевой канал данных первой степени сложности* осуществляет решение задачи классификации структур и типов входных и выходных данных, а также их преобразование для обеспечения соответствия типов данных при организации взаимодействия модулей АИС.

Теорема 2. *Нейросетевой канал данных первой степени сложности* (*или выше*) *может поставить в соответствие произвольному вектору данных* $X = \{x_n\}$ *размерности* xN *вектор* $S = \{s_e | e = \overline{1, nS}\}$ *типов данных. Тогда элемент* $s_q \in S, s_q = \max\limits_{e=1..nS}\{s_e = NN^{ct}(X)\}$ *будет однозначно определять правильный тип данных* $SX \in S$ *с ошибкой не более* ε*:*

$$|s_q - SX| \leqslant \varepsilon,$$

при условии, что отображение $X \to S$ *является непрерывным и не является инъекцией, а для любого* X *существует только один* SX*.*

Доказательство. Так как соответствие между входными наборами данных X и их типами данных S является непрерывной функцией f_{STR} на множестве X, то процедура определения типа данных заключается в нахождении непрерывной функции $f_{STR}(X) = S$. Непрерывность отображения $X \to S$ следует из утверждения, что любому вектору X_d можно поставить в соответствие тип данных $s_d \in S$. Предположим, что произвольному вектору X_d не задан тип данных $s_d \in S$, тогда всегда можно поставить ему в соответствие универсальный тип данных $s_k \in S$. Отображение $X \to S$ не является инъекцией, так как равенство типов данных — $f_{STR}(X_1) = s_1$, $f_{STR}(X_2) = s_2$ — не означает равенства исходных данных: $X_1 \neq X_2$.

Из теоремы Цыбенко [18] следует, что двухслойная нейронная сеть может аппроксимировать любую непрерывную функцию многих переменных с ошибкой, не превышающей ε. Следовательно, нейронная сеть NN^{ct} может аппроксимировать функцию f_{STR} с некоторой ошибкой ε. Так как $NN^{ct}(X) = \{s_e | s_e \in S, e = \overline{1, nS}\}$ (нейронная сеть формирует вектор S вероятности принадлежности к каждому типу данных s_e), то $s_q = \max\limits_{e=1..nS}\{s_e\}$ будет единственным решением, наиболее близким к SX. Таким образом, доказательство теоремы 2 гарантирует существование и единственность аргументов SX и SY выражения (1.3), а также возможность их определения выражением (1.2). □

Определение 5. *Нейросетевой канал данных второй степени сложности* включает функциональность канала первой степени и осуществляет решение задачи преобразования исходных данных к форме и содержимому, ожидаемых модулем АИС.

Теорема 3. *Нейросетевой канал данных второй степени сложности* (*или выше*) *может осуществить обработку произвольного многомерного вектора данных* $X = \{x_n\}$ *размерности* xN*, поставив ему в соответствие вектор* $Y = \{y_m | y_m = NN^{proc}_{i,j,type}(X)_m\}$ *размерности* yM *с ошибкой не более* ε *относительно ожидаемого выхода* $Y^r = \{y^r_m\}$*:*

$$\left| \frac{1}{yM} \sum_{m=1}^{yM} (y_m - y^*_m)^2 \right| \leqslant \varepsilon,$$

при условии, что отображение $X \to Y$ является непрерывным, инъекцией (*так как различные векторы X преобразуются в различные векторы Y*) *и сюръекцией* (*так как для $\forall Y\, \exists X (NN^{proc}_{i,j,type}(X) = Y)$*)*, таким образом, $X \to Y$ — биекция.*

Доказательство. Из второй теоремы Хехта–Нильсена [19] и следствия из теоремы Колмогорова–Арнольда [20] имеем, что для любого множества пар векторов произвольной размерности $\{(X, Y)\}$ существует трехслойная нейронная сеть с конечным числом нейронов, формирующая для каждого входного вектора X соответствующий ему выходной вектор Y. Тогда, если процесс отображения $X \to Y$ является некоторой непрерывной функцией на множестве X, то, по теореме Цыбенко [18], ее можно аппроксимировать нейронной сетью $NN^{proc}_{i,j,type}$ с ошибкой, не превышающей ε.

В противном случае, если отображение $X \to Y$ является дискретным, то есть заданным табличными значениями, его можно заменить сплайном $S_n(X)$ некоторой степени n, являющимся непрерывным. $S_n(X)$ можно аппроксимировать нейронной сетью $NN^{proc}_{i,j,type}$ с ошибкой, не превышающей ε. Доказательство теоремы обеспечивает возможность выполнения преобразования (1.5), повторяемость результатов работы нейросетевого канала данных (т. к. $X \to Y$ — инъекция), взаимно однозначное соответствие исходных и преобразованных векторов (т. к. $X \to Y$ — биекция). □

Определение 6. *Нейросетевой канал данных третьей степени сложности* включает функциональность каналов первой и второй степеней и осуществляет решение задачи выбора сетевого протокола передачи данных.

Теорема 4. *Нейросетевой канал данных третьей степени сложности может осуществить автоматический выбор сетевого протокола $CP_q \in CP$ передачи данных на основе анализа упорядоченной пары типов данных (SX, SY) при условии, что отображение $\{(SX, SY)\} \to CP$* (*в том числе для случая $SX = SY$*) *непрерывно и не является инъекцией* (*ввиду универсальности сетевых протоколов и возможности их применения для различных сочетаний (SX, SY)*)*.*

Доказательство. Аналогично доказательству теоремы 2. Соответствие между типами данных (SX, SY) и протоколами передачи данных CP_q является непрерывной функцией f_{TR} на множестве упорядоченных пар векторов $\{(SX, SY)\}$. Тогда, по теореме Цыбенко [18], аппроксимируем функцию $f_{TR}(SX, SY) = CP_q$ двухслойной нейронной сетью $NN^{tr}_{i,j,type}$. Доказательство теоремы 4 обеспечивает возможность выполнения преобразования (1.6) и автоматизации процедуры передачи данных за счет перехода от аналитического выбора сетевого протокола к автоматическому. □

В ходе доказательства вышеперечисленных теорем и исследования закономерной функционирования нейросетевых каналов данных определены следующие ограничения для использования нейросетевого метода:

- набор данных для обучения нейронных сетей задан на области определения возможных значений входных и выходных данных;
- каждому входному вектору данных должен соответствовать выходной вектор, причем единственный;
- содержимое входных и выходных данных можно привести к некоторому численному значению.

Изложенные определения, теоремы и ограничения отражают основные закономерности и принципы функционирования нейросетевого метода и нейросетевых каналов данных.

§ 3. Практическое применение нейросетевого метода обработки и передачи данных

Рассмотрим несколько примеров применения разработанного метода для решения конкретных задач обработки и передачи информации.

Пример 1. Необходимо передать данные из модуля $M1$ в модуль $M2$. Модуль $M2$ ожидает данные со структурой SY. Содержимое данных должно остаться неизменным. Сетевой протокол передачи данных CP_q задан.

Пусть заданы многомерный входной вектор X и выходной вектор $Y = X$. Определим структуру данных SX, используя многослойную нейронную сеть NN^{ct}. Тогда имеем следующие отображения: $X \to Y(Y = X)$ и $SX \to SY(SX \neq SY)$. На основе анализа SX и SY получаем, что необходимо применение нейросетевого канала данных первой степени сложности, категория — канал передачи данных ($type = T$).

По формулам (1.3) и (1.4) определяем необходимую функцию преобразования структур $FSTR_l = STR_{SX}(X) \to STR_{SY}(X)$. Так как протокол передачи данных CP_q задан, то решение задачи сводится к применению функции $FSTR_l$ к входному вектору X и передаче результата по протоколу CP_q.

Пример 2. Модулю $M1$ необходимо получить информацию Y по запросу X из базы данных (модуль $M2$). Сетевой протокол передачи данных CP_q задан.

Пусть задан многомерный входной вектор запроса X, для которого ожидается выходной вектор $Y \neq X$. Определим структуры данных SX и SY при помощи многослойной нейронной сети NN^{ct}. Получим следующие отображения: $X \to Y(Y \neq X)$ и $SX \to SY(SX \neq SY)$. На основе анализа SX и SY получаем, что необходимо применение нейросетевого канала данных второй степени сложности, категория — канал запроса данных ($type = R$).

По формулам (1.3) и (1.4) определяем необходимую функцию преобразования структур $FSTR_l = STR_{SX}(X) \to STR_{SY}(X)$. Далее обучаем нейронную сеть обработки данных $NN^{proc}_{M1,M2,R}$. Если все ограничения на применение нейросетевого метода выполнены, получим $NN^{proc}_{M1,M2,R} : STR_{SY}(X) \to STR_{SY}(Y)$. Нейронная сеть будет решать задачу регрессии. На основе входящего запроса X формируется такое сообщение для модуля $M2$, ответом на которое будет требуемые данные Y. Так как протокол передачи данных CP_q задан, то для завершения решения задачи необходимо передать преобразованные данные Y по протоколу CP_q.

Пример 3. Необходимо осуществить настройку интерфейса АИС индивидуально для каждого пользователя. Заданы векторы X — характеристики пользователя, отправленные из модуля $M1$, и Y — настройки интерфейса модуля $M2$.

Аналогично предыдущим примерам осуществляем определение структур данных SX и SY нейронной сетью NN^{ct}, после чего проводим их анализ. Так как протокол передачи данных не задан, а $X \to Y(Y \neq X)$ и $SX \to SY(SX \neq SY)$, получаем, что для решения задачи необходимо применение нейросетевого канала данных третьей степени сложности, категория — канал адаптации ($type = A$).

Далее повторяем этапы преобразования структур (по формулам (1.3) и (1.4)) и содержимого (нейронной сетью $NN^{proc}_{M1,M2,A}$) данных. Нейронная сеть обработки данных $NN^{proc}_{M1,M2,A}$ решает задачу регрессии, формируя значения параметров интерфейса (элементы вектора Y) на основе значений вектора X.

Заключительный этап состоит в анализе структур данных SX и SY нейронной сетью $NN^{tr}_{i,j,type}$, которая позволяет осуществить отображение $(SX, SY) \to CP_q$. Определенный нейронной сетью протокол CP_q используется для организации связи между модулями $M1$ и $M2$. Представленные примеры отражают алгоритмы обработки и передачи данных для нейросетевых каналов различной степени сложности.

Программная реализация нейросетевого метода обработки и передачи информации осуществлялась на основе объектно-ориентированного подхода. Первоначально разрабатывается класс-родитель $NNDC$, реализующий основные этапы предлагаемого метода. Далее нейросетевые каналы данных конкретных категорий и степеней сложности реализуются посредством наследования от класса-родителя и переопределения отдельных методов. Это позволяет упростить процесс программной реализации метода и автоматически внести коррективы в работу всех классов-потомков при изменении функциональности класса-родителя.

Для реализации нейросетевого метода каждый его этап оформляется в виде отдельного программного метода. Если этап основан на применении нейронных сетей, то осуществляется подготовка данных, формирование структуры, обучение и проверка точности нейронной сети. Архитектура нейронной сети может быть различной (многослойная, сверточная, рекуррентная, регрессионная и т. д.) в зависимости от решаемой задачи и категории нейросетевых каналов. В качестве дополнительных методов, выделенных в отдельную библиотеку, выступают функции преобразования информации из одного типа структуры в другой, а также функции предварительной обработки информации (лемматизация, перевод значений в бинарные матрицы, очистка данных от искажений, нормирование и т. д.).

§ 4. Анализ эффективности нейросетевого метода обработки и передачи данных

Для оценки эффективности разработанного нейросетевого метода обработки и передачи данных проведено его экспертное сравнение с аналитическим подходом (рис. 1). На данной схеме отражены основные этапы реализации обоих подходов к обработке и передаче данных при организации взаимодействия между модулями АИС. Блоки, выделенные пунктирной линией, обозначают действия, выполняемые автоматически на основе методов машинного обучения либо с минимальным участием человека. Тогда из рис. 1 можно сделать следующие выводы:

- большая часть этапов нейросетевого метода выполняется автоматически благодаря использованию методов машинного обучения;
- требования к пониманию предметной области для нейросетевого метода ниже, так как выполнение анализа и обработки данных осуществляется на основе «черного ящика» — нейронной сети;
- процедуры анализа и обработки информации в нейросетевом методе могут осуществляться небольшим коллективом разработчиков;
- аналитический подход при наличии высокой квалификации и тщательном анализе данных позволяет получить более точное решение;
- нейросетевой метод для корректной работы требует сбора большого объема информации для последующего обучения нейронных сетей;
- нейросетевой метод неприменим для ряда задач, не удовлетворяющих ограничениям, представленным в § 2.

Рис. 1. Сравнение аналитического подхода и нейросетевого метода обработки и передачи данных

Далее осуществлен анализ реализации нейросетевого метода с точки зрения сложности программного кода. В ходе экспериментальных исследований осуществлена разработка нейросетевого канала адаптации интерфейса (со степенью сложности $NNS = 3$) и программного обеспечения, позволяющего аналитически выставить параметры интерфейса АИС на основе некоторой экспертной информации. Таким образом, аналитический подход будет представлен сочетанием сетевого протокола и программного обеспечения, реализующего процессы анализа и обработки данных.

Для сравнения обоих подходов использовались следующие метрики [21–25]: вычислительная сложность, цикломатическая сложность, сложность программного кода Холстеда, индекс поддерживаемости кода, сложность программного кода Джилба. При сравнении будем пользоваться следующими обозначениями: ALG — аналитический подход, $NNDC$ — нейросетевой метод.

В ходе сравнительного анализа получены следующие результаты.

- Вычислительная сложность ALG и функционирования $NNDC$ сопоставимы и равны $O(N)$. Для процесса обучения нейронной сети в $NNDC$ вычислительная сложность определяется как $O(m^r), r \geqslant 1$ [26], где m — размер обучающего набора. Однако с совершенствованием программных библиотек сложность обучения падает (например, уже существуют алгоритмы автоматической настройки и обучения нейронных сетей [27–29]).

- Циклометрическая сложность (выраженная суммарной сложностью программного кода в числах McCabe [30]) для ALG составила 67, для $NNDC$ — 41. Такая разница обусловлена меньшим количеством операторов и более простой структурой программного кода $NNDC$. Большой эффект оказывает тот факт, что разработчику не требуется анализировать исходные данные, так как эту работу принимает на себя нейросетевой канал данных.

- Сложности программного кода Холстеда включает следующие параметры: $n1$ — число различных операторов, $n2$ — количество различных операндов, $N1$ — общее

число операторов, $N2$ — общее количество операндов. Тогда получим для ALG: $n1 = 8, n2 = 73, N1 = 56, N2 = 110$; для $NNDC$: $n1 = 9, n2 = 62, N1 = 34, N2 = 69$.

- Индекс поддерживаемости кода для обоих подходов сравним и оценивается как A (наилучший). Это обусловлено относительно небольшим объемом программного кода для обоих вариантов.

- Сложность программного кода Джилба для ALG — 0.55, для $NNDC$ — 0.22. Обусловлено значительным сокращением ветвлений и циклов в $NNDC$.

Подводя итог сравнительному анализу программной реализации, можно отметить, что сложность программного кода при использовании нейросетевого метода обработки и передачи информации снизилась. Улучшение данной метрики также влияет на сокращение времени разработки, экономических затрат, вероятности возникновения ошибок в программном обеспечении.

Заключение

В статье рассмотрена задача автоматизации процессов обработки и передачи данных между модулями АИС. Для ее решения на основе анализа существующих подходов к автоматизации разработки информационных систем предлагается использование технологий машинного обучения в рамках комплексного подхода.

Разработан нейросетевой метод обработки и передачи информации в АИС, основанный на применении нейронных сетей для анализа, обработки информации и выбора способов ее передачи, отличающийся комплексным теоретическим обоснованием реализации и применения нейросетевых каналов данных для автоматизированной организации межмодульного взаимодействия. Формализованы основные этапы метода: классификация типов исходных данных, их преобразование, обработка данных, выбор необходимого протокола передачи информации. Основным инструментом обработки и анализа информации в предлагаемом методе являются нейросетевые каналы данных.

Приведено теоретическое обоснование возможности реализации и применения нейросетевого метода, полученное в ходе доказательства ряда теорем. Проанализирована область применения разработанного метода, представлены примеры и описание практической реализации.

Для оценки эффективности нейросетевого метода проведены экспериментальные сравнительные исследования оценки сложности программного обеспечения с алгоритмическим подходом, проанализированы достоинства и недостатки каждого варианта. Получено, что нейросетевой метод снижает сложность решения задачи обработки и передачи данных, позволяет упростить процесс разработки, но выдвигает ряд требований к исходным данным.

В ходе дальнейших исследований планируется расширение функциональных возможностей нейросетевых каналов данных, апробация нейросетевого метода обработки и передачи данных в различных подсистемах АИС для решения задачи автоматизации их разработки.

Финансирование. Работа выполнена при финансовой поддержке Министерства науки и высшего образования РФ в рамках гранта Президента РФ МК-74.2020.9.

СПИСОК ЛИТЕРАТУРЫ

1. Keller A. Challenges and directions in service management automation // Journal of Network and Systems Management. 2017. Vol. 25. Issue 4. P. 884–901. https://doi.org/10.1007/s10922-017-9437-9
2. van Steen M., Tanenbaum A. S. A brief introduction to distributed systems // Computing. 2016. Vol. 98. Issue 10. P. 967–1009. https://doi.org/10.1007/s00607-016-0508-7

3. Zhang C., Wang X., Li F., Huang M. NNIRSS: neural network-based intelligent routing scheme for SDN // Neural Computing and Applications. 2019. Vol. 31. Issue 10. P. 6189–6205. https://doi.org/10.1007/s00521-018-3427-z
4. Genga L., Alizadeh M., Potena D., Diamantini C., Zannone N. Discovering anomalous frequent patterns from partially ordered event logs // Journal of Intelligent Information Systems. 2018. Vol. 51. Issue 2. P. 257–300. https://doi.org/10.1007/s10844-018-0501-z
5. Еремеев А. П., Кожухов А. А., Голенков В. В., Гулякина Н. А. О реализации средств машинного обучения в интеллектуальных системах реального времени // Программные продукты и системы. 2018. № 2. С. 239–245. https://doi.org/10.15827/0236-235X.122.239-245
6. Dong S., Li R. Traffic identification method based on multiple probabilistic neural network model // Neural Computing and Applications. 2019. Vol. 31. Issue 2. P. 473–487. https://doi.org/10.1007/s00521-017-3081-x
7. Kraus M., Feuerriegel S. Decision support from financial disclosures with deep neural networks and transfer learning // Decision Support Systems. 2017. Vol. 104. P. 38–48. https://doi.org/10.1016/j.dss.2017.10.001
8. Weng Y. C., Hsieh S. L. Design and implementation of a web-based medical drawing management system // Journal of Intelligent Information Systems. 2017. Vol. 49. Issue 3. P. 391–405. https://doi.org/10.1007/s10844-017-0452-9
9. Zhang J., Ding G., Zou Y., Qin S., Fu J. Review of job shop scheduling research and its new perspectives under Industry 4.0 // Journal of Intelligent Manufacturing. 2019. Vol. 30. Issue 4. P. 1809–1830. https://doi.org/10.1007/s10845-017-1350-2
10. Shokeen J., Rana C. Social recommender systems: techniques, domains, metrics, datasets and future scope // Journal of Intelligent Information Systems. 2020. Vol. 54. Issue 3. P. 633–667. https://doi.org/10.1007/s10844-019-00578-5
11. Wang T. Intelligent employment rate prediction model based on a neural computing framework and human–computer interaction platform // Neural Computing and Applications. 2020. Vol. 32. Issue 21. P. 16413–16426. https://doi.org/10.1007/s00521-019-04019-w
12. Dourish P. Algorithms and their others: Algorithmic culture in context // Big Data and Society. 2016. Vol. 3. Issue 2. P. 1–11. https://doi.org/10.1177/2053951716665128
13. Zamula A., Kavun S. Complex systems modeling with intelligent control elements // International Journal of Modeling, Simulation, and Scientific Computing. 2017. Vol. 8. No. 1. P. 1750009:1–1750009:19. https://doi.org/10.1142/S179396231750009X
14. Lin H. W., Tegmark M., Rolnick D. Why does deep and cheap learning work so well? // Journal of Statistical Physics. 2017. Vol. 168. Issue 6. P. 1223–1247. https://doi.org/10.1007/s10955-017-1836-5
15. Zhou L., Pan S., Wang J., Vasilakos A. V. Machine learning on big data: Opportunities and challenges // Neurocomputing. 2017. Vol. 237. P. 350–361. https://doi.org/10.1016/j.neucom.2017.01.026
16. Mirzaey M., Jamshidi M. B., Hojatpour Y. Applications of artificial neural networks in information system of management accounting // International Journal of Mechatronics, Electrical and Computer Technology. 2017. Vol. 7. Issue 25. P. 3523–3530.
17. Обухов А. Д., Краснянский М. Н. Нейросетевая архитектура информационных систем // Вестник Удмуртского университета. Математика. Механика. Компьютерные науки. 2019. Т. 29. Вып. 3. С. 438–455. https://doi.org/10.20537/vm190312
18. Cybenko G. V. Approximation by superpositions of a sigmoidal function // Mathematics of Control, Signals and Systems. 1989. Vol. 2. Issue 4. P. 303–314. https://doi.org/10.1007/BF02551274
19. Hecht–Nielsen R. Replicator neural networks for universal optimal source coding // Science. 1995. Vol. 269. Issue 5232. P. 1860–1863. https://doi.org/10.1126/science.269.5232.1860
20. Колмогоров А. Н. О представлении непрерывных функций нескольких переменных в виде суперпозиций непрерывных функций одного переменного и сложения // Доклады Академии наук СССР. 1957. Т. 114. № 5. С. 953–956. http://mi.mathnet.ru/dan22050
21. Kuznetsov M. A., Surkov V. O. Analysis of complexity metrics of a software code for obfuscating transformations of an executable code // IOP Conference Series: Materials Science and Engineering. 2016. Vol. 155. P. 012008. https://doi.org/10.1088/1757-899X/155/1/012008

22. Antinyan V., Staron M., Sandberg A. Evaluating code complexity triggers, use of complexity measures and the influence of code complexity on maintenance time // Empirical Software Engineering. 2017. Vol. 22. Issue 6. P. 3057–3087. https://doi.org/10.1007/s10664-017-9508-2
23. Цветков В. Я., Буравцев А. В. Метрики сложной детерминированной системы // Онтология проектирования. 2017. Т. 7. Вып. 3 (25). С. 334–346. https://doi.org/10.18287/2223-9537-2017-7-3-334-346
24. Savchenko D., Hynninen T., Taipal O. Code quality measurement: Case study // 2018 41st International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO). IEEE, 2018. P. 1455–1459. https://doi.org/10.23919/mipro.2018.8400262
25. Евдокимов И. В., Байкалов И. С., Зуденков А. И., Радионов Т. В., Цирюльникова А. М. К вопросу о метриках трудоёмкости разработки мобильных приложений // Фундаментальные исследования. 2017. № 9-1. С. 54–58. http://fundamental-research.ru/ru/article/view?id=41703
26. Лесик И. А. Решение задачи прогнозирования с использованием нейронных сетей прямого распространения на примере построения прогноза роста курса акций // Программные продукты и системы. 2015. № 2. С. 70–74. https://doi.org/10.15827/0236-235X.110.070-074
27. Jin H., Song Q., Hu X. Auto-keras: An efficient neural architecture search system // Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining. Association for Computing Machinery, 2019. P. 1946–1956. https://doi.org/10.1145/3292500.3330648
28. Mendoza H., Klein A., Feurer M., Springenberg J. T., Urban M., Burkart M., Dippel M., Lindauer M., Hutter F. Towards automatically-tuned deep neural networks // Automated Machine Learning. 2019. P. 135–149. https://doi.org/10.1007/978-3-030-05318-5_7
29. Gong X., Chang S., Jiang Y., Wang Z. Autogan: Neural architecture search for generative adversarial networks // International Conference on Computer Vision (ICCV). IEEE, 2019. P. 3223–3233. https://doi.org/10.1109/ICCV.2019.00332
30. Pan W., Chai C. Measuring software stability based on complex networks in software // Cluster Computing. 2019. Vol. 22. Issue 2. P. 2589–2598. https://doi.org/10.1007/s10586-017-1353-y

Поступила в редакцию 20.07.2020

Обухов Артем Дмитриевич, к. т. н., доцент, кафедра систем автоматизированной поддержки принятия решений, Тамбовский государственный технический университет, 392000, Россия, г. Тамбов, ул. Советская, 106.
ORCID: https://orcid.org/0000-0002-3450-5213
E-mail: obuhov.art@gmail.com

Краснянский Михаил Николаевич, д. т. н., профессор, ректор, Тамбовский государственный технический университет, 392000, Россия, г. Тамбов, ул. Советская, 106.
ORCID: https://orcid.org/0000-0002-8751-7445
E-mail: kras@tambov.ru

Цитирование: А. Д. Обухов, М. Н. Краснянский. Нейросетевой метод обработки и передачи данных в адаптивных информационных системах // Вестник Удмуртского университета. Математика. Механика. Компьютерные науки. 2021. Т. 31. Вып. 1. С. 149–164.

A. D. Obukhov, M. N. Krasnyanskiy

Neural network method of data processing and transmission in adaptive information systems

Keywords: data processing and transmission neural network method, neural network data channel, neural networks, adaptive information systems.

MSC2020: 68T01, 68T05

DOI: 10.35634/vm210111

The problem of automation of the processes of information transmission and processing in adaptive information systems is considered. An analysis of existing approaches to solving this problem showed the prospects of using neural network technologies. A neural network method for processing and transmitting information in adaptive information systems is formulated. The method includes a formalized description of a neural network data channel — a software tool for analysis, data processing and selection of data transfer protocol. The main stages of the proposed method are outlined: classification of the structures of the source data, their transformation, data processing, selection of the necessary protocol for transmitting information. Each of the stages is implemented through neural networks of various architectures. The theoretical rationale of the possibility of using the neural network method is given, obtained on the basis of the proof of a number of theorems. The novelty of the proposed method consists in the transition from an analytical solution of the problems of classification, processing and data transfer to an automated approach using machine learning technologies. The practical significance of the neural network method is to reduce the complexity of the implementation of information processing and transmission processes, to increase the level of automation in the organization of intermodular interaction. The implementation of the neural network method has been assessed using a number of software complexity assessment metrics. The application, virtues and failings of the developed method are analyzed.

Funding. This work was supported by the Ministry of Science and Higher Education of the Russian Federation within the grant of the President of the Russian Federation, MK-74.2020.9.

REFERENCES

1. Keller A. Challenges and directions in service management automation, *Journal of Network and Systems Management*, 2017, vol. 25, issue 4, pp. 884–901. https://doi.org/10.1007/s10922-017-9437-9
2. van Steen M., Tanenbaum A. S. A brief introduction to distributed systems, *Computing*, 2016, vol. 98, issue 10, pp. 967–1009. https://doi.org/10.1007/s00607-016-0508-7
3. Zhang C., Wang X., Li F., Huang M. NNIRSS: neural network-based intelligent routing scheme for SDN, *Neural Computing and Applications*, 2019, vol. 31, issue 10, pp. 6189–6205. https://doi.org/10.1007/s00521-018-3427-z
4. Genga L., Alizadeh M., Potena D., Diamantini C., Zannone N. Discovering anomalous frequent patterns from partially ordered event logs, *Journal of Intelligent Information Systems*, 2018, vol. 51, issue 2, pp. 257–300. https://doi.org/10.1007/s10844-018-0501-z
5. Eremeev A. P., Kozhuhov A. A., Golenkov V. V, Gulyakina N. A. On the implementation of machine learning tools in real-time intelligent systems, *Programmnye Produkty i Sistemy*, 2018, no. 2, pp. 239–245 (in Russian). https://doi.org/10.15827/0236-235X.122.239-245
6. Dong S., Li R. Traffic identification method based on multiple probabilistic neural network model, *Neural Computing and Applications*, 2019, vol. 31, issue 2, pp. 473–487. https://doi.org/10.1007/s00521-017-3081-x
7. Kraus M., Feuerriegel S. Decision support from financial disclosures with deep neural networks and transfer learning, *Decision Support Systems*, 2017, vol. 104, pp. 38–48. https://doi.org/10.1016/j.dss.2017.10.001

8. Weng Y. C., Hsieh S. L. Design and implementation of a web-based medical drawing management system, *Journal of Intelligent Information Systems*, 2017, vol. 49, issue 3, pp. 391–405. https://doi.org/10.1007/s10844-017-0452-9
9. Zhang J., Ding G., Zou Y., Qin S., Fu J. Review of job shop scheduling research and its new perspectives under Industry 4.0, *Journal of Intelligent Manufacturing*, 2019, vol. 30, issue 4, pp. 1809–1830. https://doi.org/10.1007/s10845-017-1350-2
10. Shokeen J., Rana C. Social recommender systems: techniques, domains, metrics, datasets and future scope, *Journal of Intelligent Information Systems*, 2020, vol. 54, issue 4, pp. 633–667. https://doi.org/10.1007/s10844-019-00578-5
11. Wang T. Intelligent employment rate prediction model based on a neural computing framework and human–computer interaction platform, *Neural Computing and Applications*, 2020, vol. 32, issue 21, pp. 16413–16426. https://doi.org/10.1007/s00521-019-04019-w
12. Dourish P. Algorithms and their others: Algorithmic culture in context, *Big Data and Society*, 2016, vol. 3, issue 2, pp. 1–11. https://doi.org/10.1177/2053951716665128
13. Zamula A., Kavun S. Complex systems modeling with intelligent control elements, *International Journal of Modeling, Simulation, and Scientific Computing*, 2017, vol. 8, no. 1, pp. 1750009:1–1750009:19. https://doi.org/10.1142/S179396231750009X
14. Lin H. W., Tegmark M., Rolnick D. Why does deep and cheap learning work so well?, *Journal of Statistical Physics*, 2017, vol. 168, issue 6, pp. 1223–1247. https://doi.org/10.1007/s10955-017-1836-5
15. Zhou L., Pan S., Wang J., Vasilakos A. V. Machine learning on big data: Opportunities and challenges, *Neurocomputing*, 2017, vol. 237, pp. 350–361. https://doi.org/10.1016/j.neucom.2017.01.026
16. Mirzaey M., Jamshidi M., Hojatpour Y. Applications of artificial neural networks in information system of management accounting, *International Journal of Mechatronics, Electrical and Computer Technology*, 2017, vol. 7, issue 25, pp. 3523–3530.
17. Obukhov A. D., Krasnyanskiy M. N. Neural network architecture of information systems, *Vestnik Udmurtskogo Universiteta. Matematika. Mekhanika. Komp'yuternye Nauki*, 2019, vol. 29, issue 3, pp. 438–455 (in Russian). https://doi.org/10.20537/vm190312
18. Cybenko G. V. Approximation by superpositions of a sigmoidal function, *Mathematics of Control, Signals and Systems*, 1989, vol. 2, issue 4, pp. 303–314. https://doi.org/10.1007/BF02551274
19. Hecht–Nielsen R. Replicator neural networks for universal optimal source coding, *Science*, 1995, vol. 269, issue 5232, pp 1860–1863. https://doi.org/10.1126/science.269.5232.1860
20. Kolmogorov A. N. On the representation of continuous functions of many variables by superposition of continuous functions of one variable and addition, *Doklady Akademii Nauk SSSR*, 1957, vol. 114, no. 5, pp. 953–956 (in Russian). http://mi.mathnet.ru/eng/dan22050
21. Kuznetsov M. A., Surkov V. O. Analysis of complexity metrics of a software code for obfuscating transformations of an executable code, *IOP Conference Series: Materials Science and Engineering*, 2016, vol. 155, pp. 012008. https://doi.org/10.1088/1757-899X/155/1/012008
22. Antinyan V., Staron M., Sandberg A. Evaluating code complexity triggers, use of complexity measures and the influence of code complexity on maintenance time, *Empirical Software Engineering*, 2017, vol. 22, issue 6, pp. 3057–3087. https://doi.org/10.1007/s10664-017-9508-2
23. Tsvetkov V. Ya., Buravtsev A. V. Metrics of a complex determinate system, *Ontology of Designing*, 2017, vol. 7, issue 3, pp. 334–346 (in Russian). https://doi.org/10.18287/2223-9537-2017-7-3-334-346
24. Savchenko D., Hynninen T., Taipal O. Code quality measurement: Case study, *2018 41st International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO)*, IEEE, 2018, pp. 1455–1459. https://doi.org/10.23919/mipro.2018.8400262
25. Evdokimov I. V., Baikalov I. S., Zudenkov A. I., Radionov T. V., Tsiryul'nikova A. M. Considering the question of the metrics of the labority of development of mobile application, *Fundamental Research*, 2017, no. 9-1, pp. 54–58 (in Russian). http://fundamental-research.ru/ru/article/view?id=41703
26. Lesik I. A. Forecasting stock price growth using feedforward neural networks, *Software and Systems*, 2015, no. 2, pp. 70–74 (in Russian). https://doi.org/10.15827/0236-235X.110.070-074

27. Jin H., Song Q., Hu X. Auto-keras: An efficient neural architecture search system, *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery Data Mining*, Association for Computing Machinery, 2019, pp. 1946–1956. https://doi.org/10.1145/3292500.3330648
28. Mendoza H., Klein A., Feurer M., Springenberg J. T., Urban M., Burkart M., Dippel M., Lindauer M., Hutter F. Towards automatically-tuned deep neural networks, *Automated Machine Learning*, 2019, pp. 135–149. https://doi.org/10.1007/978-3-030-05318-5_7
29. Gong X., Chang S., Jiang Y., Wang Z. Autogan: Neural architecture search for generative adversarial networks, *International Conference on Computer Vision (ICCV)*, IEEE, 2019, pp. 3223–3233. https://doi.org/10.1109/ICCV.2019.00332
30. Pan W., Chai C. Measuring software stability based on complex networks in software, *Cluster Computing*, 2019, vol. 22, issue 2, pp. 2589–2598. https://doi.org/10.1007/s10586-017-1353-y

Received 20.07.2020

Obukhov Artem Dmitrievich, Candidate of Engineering, Associate Professor, Department of Automated Systems of Decision-Making Support, Tambov State Technical University, ul. Sovetskaya, 106, Tambov, 392000, Russia.
ORCID: https://orcid.org/0000-0002-3450-5213
E-mail: obuhov.art@gmail.com

Krasnyanskiy Mikhail Nikolaevich, Doctor of Engineering, Rector, Tambov State Technical University, ul. Sovetskaya, 106, Tambov, 392000, Russia.
ORCID: https://orcid.org/0000-0002-8751-7445
E-mail: kras@tambov.ru

Citation: A. D. Obukhov, M. N. Krasnyanskiy. Neural network method of data processing and transmission in adaptive information systems, *Vestnik Udmurtskogo Universiteta. Matematika. Mekhanika. Komp'yuternye Nauki*, 2021, vol. 31, issue 1, pp. 149–164.