

УДК 004.8

ГИБРИДНЫЙ МЕТОД ОБУЧЕНИЯ ИСКУССТВЕННОЙ НЕЙРОННОЙ СЕТИ НА ОСНОВЕ МОДИФИЦИРОВАННОГО АЛГОРИТМА МУРАВЬЯ

Е. В. Котляров

Ассистент*

Контактный тел.: 050-392-75-36

E-mail: jimis@ua.fm

Т. И. Петрушина

Кандидат физико-математических наук, доцент*

E-mail: tatyana.petrushina@gmail.com

*Кафедра математического обеспечения компьютерных систем

Одесский национальный университет

им. И.И. Мечникова

ул. Дворянская, 2, г. Одесса, Украина, 65026

В статті запропоновано модифікований алгоритм мурашиної колонії для гібридного методу навчання штучної нейронної мережі. Даний алгоритм орієнтований на індивідуальне коректування вагів нейронної мережі

Ключові слова: нейронні мережі, алгоритм мурашиної колонії

В статье предложен модифицированный алгоритм муравьиной колонии для гибридного метода обучения искусственной нейронной сети. Данный алгоритм ориентирован на индивидуальную корректировку весов нейронной сети

Ключевые слова: нейронные сети, алгоритм муравьиной колонии

Постановка проблемы и анализ публикаций

Одной из основных задач теории искусственного интеллекта является обучение искусственной нейронной сети. Многие задачи ввода и обработки информации, передачи информации в коммуникационных системах связи, в интернете, в экономике, бизнесе, медицине, при автоматизации производства, в политологических и социологических технологиях, в охранных системах, в геологоразведке и другие [1-2] решены при использовании нейронных сетей.

Основным методом обучения является возникший в 1974 году одновременно и независимо друг от друга в работах А.И. Галушкина [3] и Пола Дж. Вербоса [4] метод обратного распространения ошибки. В последующие годы этот метод получил существенное развитие в работах Дэвида И. Румельхарта, Дж. Е. Хинтона и Рональда Дж. Вильямса [5], и С.И. Барцева и В.А. Охонина [6] и многочисленные применения к различным задачам. Его появление позволило преодолеть некоторые теоретические трудности и дать дополнительный толчок развитию теории искусственных нейронных сетей. Однако такие недостатки метода обратного распространения ошибки как паралич сети, наличие локальных минимумов, сложность корректного выбора размера шага, временная неустойчивость процесса, не всегда позволяют решить поставленную прикладную задачу.

Возникла потребность для оптимизации обучения нейронной сети усовершенствовать метод обратного распространения ошибки при помощи гибридного алгоритма.

Использование в гибридном алгоритме вместе с методом обратного распространения ошибки алгоритма муравьиной колонии началось в начале XXI века.

В работе [7] гибридное обучение нейронной сети с использованием алгоритма муравья применено для исследования тоннелей угольных шахт. Нейронная сеть обучается комбинацией алгоритмов: нечеткого алгоритма С-средних и алгоритма муравья, что позволило снизить вычислительную сложность задачи. Сеть используется для предсказания обрушения тоннелей, при этом факторы, влияющие на прочность стен, обладают такими характеристиками как изменчивость, нелинейность, нечеткость и др.

В работе [8] гибридный алгоритм обучения применяется для решения задач приближения функций и моделирования количественных активных отношений гербицидов. Сеть обучается двумя алгоритмами: алгоритмом обратного распространения ошибки и алгоритмом муравьиной колонии. Результаты экспериментов показали, что предложенный гибридный алгоритм достигает заданную точность за меньшее количество итераций, чем алгоритм обратного распространения. При этом также изучается вопрос о возможности изменения количества узлов на скрытых слоях.

Использование алгоритма муравья при обучении нейронной сети алгоритмом обратного распространения для комбинационной оптимизации проводится в работе [9]. При этом претерпел изменения базовый алгоритм муравья: обе матрицы – дискретная матрица феромонов и вероятностная матрица – предполагаются не дискретными, а непрерывными.

Основная часть

Рассматривается задача обучить искусственную нейронную сеть алгоритмом муравьиной колонии для решения задач классификации. Т.к. для таких задач в

качестве искусственной нейронной сети, как правило, выбирается многослойный персептрон, то рассмотрим обучение многослойного персептрона гибридным алгоритмом, сочетающим алгоритм муравьиной колонии и алгоритм обратного распространения ошибки с «учителем».

Обучаемой нейронной сетью является многослойный персептрон с одним скрытым слоем. При этом рассматривается зафиксированная конфигурация сети (рис. 1): на входном слое – n узлов, на скрытом слое – m узлов и на выходном слое – p узлов. Количество узлов на входном и выходном слоях задаются с учетом конкретной решаемой прикладной задачи.

Заметим, что количество узлов на скрытом слое или задается с учетом смысла конкретной решаемой задачи, или зависит от количества узлов на входном и выходном слоях.

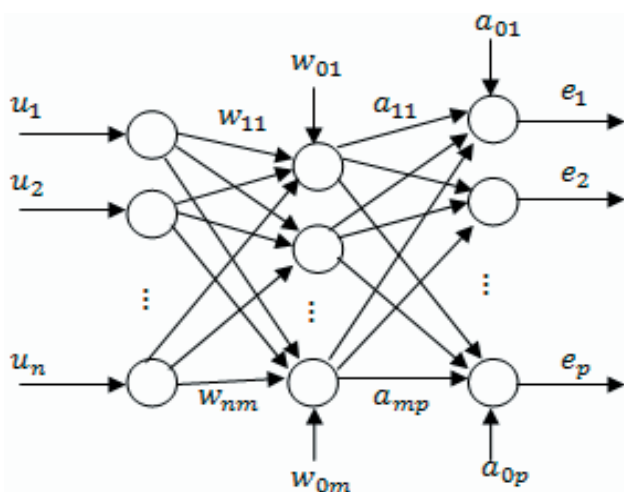


Рис. 1. Структура нейронной сети

Опишем структуру обучаемой нейронной сети. Пусть $u = (u_1, u_2, \dots, u_n)$ – вектор входных сигналов и $e = (e_1, e_2, \dots, e_p)$ – вектор выходных сигналов. Каждый нейрон входного слоя связан с каждым нейроном скрытого слоя со своим весом. Обозначим матрицу значений весов между входными и скрытыми узлами через $W = \|w_{ij}\|$, где w_{ij} – вес между i -м входным и j -м скрытым узлами ($i = 1, 2, \dots, n, j = 1, 2, \dots, m$). Матрицу значений весов между скрытым и выходным слоями обозначим через $A = \|a_{ij}\|$, где a_{ij} – вес между i -м скрытым и j -м выходным узлами ($i = 1, 2, \dots, m, j = 1, 2, \dots, p$). Каждый нейрон скрытого и выходного слоев имеет смещение, т.е. введем вектора смещений: скрытого слоя $W_0 = (w_{01}, w_{02}, \dots, w_{0m})$, где w_{0i} – смещение на i -м узле скрытого слоя, выходного слоя $A_0 = (a_{01}, a_{02}, \dots, a_{0p})$, где a_{0j} – смещение на j -м узле выходного слоя.

В качестве сумматора используем оператор нахождения взвешенной суммы по формуле:

$$\tilde{v}_j = \sum_{i=1}^n w_{ij} u_i + w_{0j}, j = 1, 2, \dots, m, \quad (1)$$

где $\tilde{v}_j, j = 1, 2, \dots, m$ – найденный входной сигнал j -го скрытого узла.

Обозначим через $\tilde{v} = (\tilde{v}_1, \tilde{v}_2, \dots, \tilde{v}_m)$ – вектор входного сигнала скрытого слоя, тогда формула (1) в векторно-матричной форме имеет вид $\tilde{v} = u \cdot W + W_0$.

Пусть $y = f(x)$ – активационная функция. Заметим, что для алгоритма обратного распространения ошибки, как правило, выбирается или $f \in C^1(\mathfrak{X})$, или если $f \in C(\mathfrak{X})$, то функция f дифференцируема почти всюду в \mathfrak{X} . При этом либо $f: \mathfrak{X} \rightarrow [0; 1]$, либо $f: \mathfrak{X} \rightarrow (0; 1)$, либо $f: \mathfrak{X} \rightarrow [0; 1)$, либо $f: \mathfrak{X} \rightarrow (0; 1]$, либо $f: \mathfrak{X} \rightarrow (-1; 1)$.

При этом, не загромождая работу лишними обозначениями, если $x = (x_1, x_2, \dots, x_k)$, где $x_j \in \mathfrak{X}, j = 1, 2, \dots, k$, под $f(x)$ будем понимать вектор

$$f(x) = (f(x_1), f(x_2), \dots, f(x_k)). \quad (2)$$

Значения на выходе каждого узла скрытого слоя при помощи активационной функции находим по формуле:

$$v_j = f(\tilde{v}_j), j = 1, 2, \dots, m. \quad (3)$$

Обозначим через $v = (v_1, v_2, \dots, v_m)$ – вектор выходного сигнала скрытого слоя. Тогда значения на выходе скрытого слоя, найденные по формуле (3), согласно (2) в векторно-матричной форме имеют вид: $v = f(\tilde{v})$.

Переходим к выходному слою. j -тый выходной сигнал работы сети вычислим по формуле

$$e_j = f\left(\sum_{i=1}^m a_{ij} v_i + a_{0j}\right), j = 1, 2, \dots, p. \quad (4)$$

Формула (4) согласно (2) в векторно-матричной форме имеет вид $e = f(v \cdot A + A_0)$.

Рассмотрим одну итерацию обучения сети, используя обучение «с учителем». Изначально элементы матриц W, A и компоненты векторов смещения W_0, A_0 зададим произвольными вещественными числами из промежутка $[-1; 1]$. Зададим вектор входных сигналов u и вектор «правильных» выходных сигналов $c = (c_1, c_2, \dots, c_p)$.

Согласно формулам (1), (3) и (4) прогоняем входной сигнал u по сети и получаем выходной сигнал e . Значение ошибки для i -того выходного узла с учетом «правильных» выходных сигналов полагаем равным:

$$err_i = c_i - e_i, i = 1, 2, \dots, p. \quad (5)$$

Вводя вектор ошибки $err = (err_1, err_2, \dots, err_p)$, где его компоненты $err_i, i = 1, 2, \dots, p$, вычисляются по формуле (5), получаем возможность оценивать «близость» значений компонент вектора полученного выходного сигнала e и «правильного» выходного сигнала c .

Описанный в данном пункте алгоритм основан на классическом алгоритме распространения сигнала в многослойных персептронах. В дальнейшем, используя данный алгоритм, ставим задачу его модификации так, чтобы ускорить обучение многослойного персептрона. Для этого будет использован также модифицированный алгоритм муравьиной колонии.

Предложим модификацию алгоритма муравьиной колонии для выбора приоритетных путей движения по нейронной сети при ее обучении. Основная цель,

которая при этом преследуется, состоит в ускорении в дальнейшем процесса обучения нейронной сети.

Пусть количество агентов (муравьев), используемых для обучения нейронной сети, равно $l \cdot n$, где $l \in \mathbb{N}$ – количество муравьев на 1 входной узел. Будем находить l по формуле:

$$l = B \cdot \text{ceil}\left(\frac{\max(m, p)}{n}\right),$$

где $B \in \mathbb{N}$ – параметр, которым может управлять пользователь, чтобы увеличить или уменьшить количество муравьев на 1 входной узел,

$$\text{ceil}(x) = \begin{cases} [x] + 1, x \notin \mathbb{Z}, \\ [x], x \in \mathbb{Z}, \end{cases}$$

$[x]$ – целая часть числа x . Формула подсчета количества муравьев на 1 входной узел обеспечивает посещение муравьями всех узлов на скрытом и выходном слоях сети.

Равномерно распределим муравьев по n узлам входного слоя, по l муравьев на каждом входном узле. Будем считать, что за свое путешествие один муравей посещает всего 3 узла, по одному узлу на каждом слое, а значит, нет необходимости использовать список табу.

Для взаимодействия друг с другом и выбора приоритетных путей муравьи используют феромоны (секреты специальных желез). На каждом участке пути один муравей «оставляет» некоторое количество феромонов, подсчет которого проведем ниже.

Для обеспечения посещения всех узлов при первом путешествии муравьев, когда в сети еще нет феромонов, введен дополнительный алгоритм случайного выбора еще не посещенных узлов муравьями на скрытом и выходном слоях.

Идея алгоритма заключается в том, что муравей, будучи на входном или скрытом слое, имеет определенный набор еще не посещенных узлов «следующего» слоя. Если этот список – пуст, то муравей может выбрать любой следующий узел.

После выбора некоторым муравьем узла, для следующих за ним муравьев этот узел становится недоступным.

Запускаем муравьев.

Под *одним* путешествием $l \cdot n$ муравьев понимаем последовательный запуск по одному муравью из каждого из n входных узлов до тех пор, пока все $l \cdot n$ муравьев не закончат свои путешествия. При этом каждый муравей запоминает две пары узлов, в которых он находился, а значит, запоминает те участки пути, по которым он прошел. Когда все муравьи закончили очередное *одно* путешествие, проводится распределение феромонов между узлами входного и скрытого слоев и между узлами скрытого и выходного слоев.

Распределение феромонов существенно зависит от того, в какой выходной узел пришел муравей, а точнее, от величины ошибки (5) в i -м выходном узле. Наличие, в том числе и количество, или отсутствие феромонов на фиксированном участке пути являет-

ся сигналом для каждого муравья, который попал в узел начала следующего участка пути, нужно ли на следующем шаге выбирать этот участок или нет. При этом на следующем шаге, выходя из узла, в котором он оказался, муравей будет выбирать тот путь, количество феромонов на котором не меньше, чем на остальных путях, исходящих из данного узла. Здесь существенно то, что при наличии одинакового количества феромонов на нескольких путях, остается случайный выбор участка пути муравьем.

Будем также считать, что с течением времени феромоны испаряются, т.е. «притуляется» информация о выборе следующего участка пути, а значит, также будем учитывать испарение феромонов при помощи коэффициента испарения $\rho \in (0; 1)$. Если на двух последовательных итерациях, т.е. после окончания очередного *одного* путешествия муравьев, пути муравьев не изменились, то останавливаем путешествие муравьев, и считаем, что приоритетные пути движения по нейронной сети выбраны.

Опишем алгоритм выбора приоритетных путей муравьями при помощи формул.

Для хранения значений феромонов между слоями зададим матрицы феромонов Φ и Ψ . Обозначим матрицу значений феромонов между входными и скрытыми узлами через $\Phi = \|\Phi_{ij}\|$, где Φ_{ij} – количество феромонов между i -м входным и j -м скрытым узлами.

Матрицу значений феромонов между скрытым и выходным слоями обозначим через $\Psi = \|\Psi_{ij}\|$, где Ψ_{ij} – количество феромонов между i -м скрытым и j -м выходным узлами.

Изначально все элементы матриц Φ и Ψ полагаем равными нулю.

Приоритет при выборе муравьем следующего участка пути будем определять так:

– выходя из i -того входного узла, муравей выбирает j -тый скрытый узел, если

$$\Phi_{ij} = \max_{k=1,2,\dots,m} \Phi_{ik} \quad (6)$$

– муравей выбирает j -тый выходной узел, выходя из i -того скрытого узла, если

$$\Psi_{ij} = \max_{k=1,2,\dots,p} \Psi_{ik} \quad (7)$$

– при движении или из входного узла, или из скрытого узла, если найдены несколько ребер с максимальной концентрацией феромонов, то муравей случайным образом выбирает любое из них.

Обновление матриц Φ и Ψ , будем проводить тогда, когда все $l \cdot n$ муравьев закончили очередное свое *одно* путешествие, т.е., выйдя из некоторого входного узла, пришли в выбранные, с учетом формул (6) и (7), фиксированные скрытый и, далее, в выходной узлы. При этом количество феромонов, которое муравьи распространяют на пройденных ребрах сети, предлагаем считать зависящим от выходного узла, в который пришел муравей. Для этого:

1. Количество феромонов, которое муравей оставляет на каждом из двух пройденных ребер, вычисляем по формуле:

$$pher_i = \frac{Q}{|err_i|}, i = 1, 2, \dots, p, \quad (8)$$

где Q – задаваемая в каждой конкретной задаче фиксированная постоянная, отвечающая количеству феромонов одного муравья;

2. После того, как все муравьев закончили очередное свое *одно* путешествие, обновление матриц Φ и Ψ проведем по следующей схеме:

а) если муравей не проходил по ребру, то значение соответствующего элемента матрицы феромонов на этом ребре не изменяется;

б) при подсчете количества феромонов на каждом ребре можно использовать или алгоритм «без испарения» ($\rho = 0$) или алгоритм с учетом испарения на каждом ребре ;

с) если муравей прошел путь от i -того входного узла k -тому скрытому узлу и далее k -тому выходному узлу, то обновляются только элементы ϕ_{ij} и ψ_{ij} матриц Φ и Ψ , им присваиваются новые значения ϕ_{ij}^* и ψ_{jk}^* по формулам

$$\phi_{ij}^* = (\phi_{ij} + pher_k) * (1 - \rho), \quad (9)$$

$$\psi_{jk}^* = (\psi_{jk} + pher_k) * (1 - \rho), \quad (10)$$

где ρ – коэффициент испарения феромонов, $\rho \in [0; 1)$. При этом заметим, что учитывается ошибка $pher_k$ только соответствующего k -того выходного узла. Домножение на $(1 - \rho)$ соответствует испарению феромонов, во-первых, для того чтобы «притупить» информацию о выборе следующего участка пути, а во-вторых, чтобы убрать пути, ведущие к большим ошибкам.

Как указано выше, одно путешествие муравьев последовательно согласно формулам (6)-(10) происходит до тех пор, пока на двух последовательных итерациях пути муравьев не изменяются. Как только это произошло, останавливаем путешествие муравьев. Приоритетные пути движения по нейронной сети агентами выбраны.

При этом они запомнили ребра, по которым ходили. Переходим к обновлению весов сети.

Для ускорения обучения нейронной сети будем использовать гибридный алгоритм: модифицированный выше алгоритм муравьиной колонии и модифицированный алгоритм обратного распространения ошибки «с учителем».

При этом в последнем предлагаем ввести функцию для учета феромонов.

Изначально элементы матриц W , A и компоненты векторов смещения W_0 , A_0 зададим произвольными вещественными числами из промежутка $[-1; 1]$. Зададим вектор входных сигналов u и вектор «правильных» выходных сигналов ($c = c_1, c_2, \dots, c_p$).

Прогоняем входной сигнал u по сети согласно формулам (1), (3) и (4) и получаем выходной сигнал e . Считаем значения ошибок err_i согласно (5) для каждого выходного узла сети, $i = 1, 2, \dots, p$.

Когда муравьи выбрали приоритетные маршруты по сети, учитывая распределение феромонов муравьями, делаем корректировку весов, $w_{ij}, i = 1, 2, \dots, n, j = 1, 2, \dots, m$ между входным и скрытым слоями, и весов $a_{jk}, j = 1, 2, \dots, m, k = 1, 2, \dots, p$ между скры-

тым и выходным слоями, смещений $w_{0j}, j = 1, 2, \dots, m$ на скрытом слое и смещений $a_{0k}, k = 1, 2, \dots, p$ на выходном слое.

Для этого

– находим сумму элементов матриц Φ и Ψ :

$$\Phi_0 = \sum_{i=1}^n \sum_{j=1}^m \phi_{ij}, \quad \Psi_0 = \sum_{i=1}^m \sum_{j=1}^p \psi_{ij};$$

– величины ошибок для корректировки весов между скрытым и выходным слоями вычисляем по формуле:

$$\delta_{ij} = (c_i - e_i) e_i (1 - e_i) \eta \left(\frac{\psi_{ij}}{\Psi_0} \right), i = 1, 2, \dots, m, j = 1, 2, \dots, p, \quad (11)$$

где $\eta: [0; 1] \rightarrow [1; \eta_0]$ – функция для учета феромонов такая, что $\eta \in C([0, 1])$, является строго монотонно убывающей, $\eta(1) = 1$ и $\eta(0) = \eta_0, \eta_0 > 1$.

– величины ошибок для корректировки весов между входным и скрытым слоями вычисляем по формуле:

$$\delta'_{ij} = \sum_{k=1}^p a_{jk} \delta_{jk} \cdot v_i (1 - v_i) \eta \left(\frac{\phi_{ij}}{\Phi_0} \right), i = 1, 2, \dots, n, j = 1, 2, \dots, m. \quad (12)$$

В итоге обновление элементов матриц A и W , векторов смещения, присвоив им соответственно новые значения $a_{ij}^*, w_{ij}^*, a_{0i}^*$ и w_{0i}^* , предлагаем вычислять по формулам:

$$a_{ij}^* = a_{ij} + \theta \delta_{ij} e_j, i = 1, 2, \dots, m, j = 1, 2, \dots, p, \quad (13)$$

$$w_{ij}^* = w_{ij} + \theta \delta'_{ij} u_i, i = 1, 2, \dots, n, j = 1, 2, \dots, m, \quad (14)$$

$$a_{0i}^* = a_{0i} + \theta a_{0i} \max_{j=1, 2, \dots, m} \delta_{ji}, i = 1, 2, \dots, p, \quad (15)$$

$$w_{0i}^* = w_{0i} + \theta w_{0i} \frac{1}{n} \sum_{j=1}^n \delta'_{ji}, i = 1, 2, \dots, m, \quad (16)$$

где θ – заданный коэффициент обучения сети, регулирующий скорость обучения.

В итоге, предложенный модифицированный алгоритм обучения нейронной сети с помощью алгоритма муравья обладает следующими свойствами:

– в отличие от классического алгоритма обратного распространения ошибки, в котором движение по многослойному персептрон не имеет приоритетов, предложен при помощи модифицированного алгоритма муравья выбор приоритетных путей для ускорения обучения нейронной сети;

– при выборе приоритетных путей движения муравьев введены две матрицы учета феромонов Φ и Ψ , которые позволяют упростить учет распространения феромонов на каждом участке пути движения муравьев;

– если в классическом алгоритме оптимизации по принципу муравьиной колонии учитывается расстояние, пройденное каждым муравьем на каждом участке пути, то в предложенном алгоритме расстояние заменяется величиной ошибки (5) в каждом выходном узле, в который пришел любой муравей;

– для увеличения скорости сходимости процесса обучения нейронной сети введена функция для учета феромонов $\eta = \eta(x)$;

– изменен, по сравнению с классическим алгоритмом муравья, вероятностный учет движения муравья, а именно, в настоящей работе в качестве аргумента функции для учета феромонов рассматривается $\frac{\Phi_{ij}}{\Phi_0}$;

– изменены формулы для подсчета ошибок между входным и скрытым слоями и между скрытым и выходным слоями, а именно, используются формулы (11) и (12), в которых существенную роль играет функция для учета феромонов, предложенная в настоящей работе;

– корректировка всех весов сети (13), (14) и всех смещений (15), (16) проводится не только при помощи функции обратной активационной функции $f(x) = \frac{1}{1+e^{-x}}$, как в классическом алгоритме обратного распространения ошибки, но и при помощи введенной функции для учета феромонов.

Остановка алгоритма происходит тогда, когда с заданной точностью сеть «правильно» отвечает или на все обучающие примеры, или на указанный пользователем процент таких примеров.

Заключение

Предложен гибридный алгоритм обучения искусственной нейронной сети на основе алгоритма муравьиной колонии и алгоритма обратного распространения ошибки с «учителем».

Алгоритм муравьиной колонии используется для выбора оптимальных путей движения по нейронной сети и оптимального изменения весов сети при помощи феромонов.

Равномерное распределение муравьев по всем узлам входного слоя позволяет избежать проблемы попадания в локальный минимум. Алгоритм обеспе-

чивает посещение муравьями всех узлов на скрытом и выходном слоях сети.

По сравнению с базовым, алгоритм муравьиной колонии изменен. В базовом алгоритме за основу взят учет расстояния между каждой парой соединенных узлов. В предложенном алгоритме используется учет величины ошибки в конце путешествия каждого муравья.

Кроме того, изменение элементов матриц феромонов происходит с учетом величины ошибки на каждом выходном узле вместо учета общей ошибки сети (как в базовом алгоритме).

В модифицированном алгоритме корректировка всех весов сети и всех смещений происходит тогда, когда приоритетные пути движения по сети муравьями выбраны.

При этом веса сети изменяются с учетом значений элементов матрицы феромонов, ошибки на каждом узле выходного слоя и предложенной функции для учета феромонов. Алгоритм позволяет изменять каждый вес сети со своим коэффициентом. Все это в целом позволяет поднять вопрос обучения нейронной сети на качественно новый уровень.

Работа алгоритма проверена на широко известных базах данных [<http://archive.ics.uci.edu/ml/>]. Эксперименты были проведены на нескольких классах задач. Результаты экспериментов показали, что по сравнению с классическим алгоритмом обратного распространения ошибки предложенный модифицированный алгоритм ускоряет обучение нейронной сети. Детальный анализ тестирования алгоритма для конкретных задач будет предложен в последующем.

Внедрение алгоритма апробировано на Microsoft SQL Server 2005 в службе Analysis Services, что позволит применять предложенный алгоритм для решения других задач классификации. При этом особый интерес представляет апробация алгоритма на реальных прикладных задачах.

В настоящее время проводятся эксперименты на больших нейронных сетях и больших объемах данных.

Литература

1. Саймон Хайкин, Нейронные сети полный курс второе издание – М. Вильямс, 2006, 1104 с.
2. Anil K. Jain, Jianchang Mao, K.M. Mohiuddin Artificial Neural Networks: A Tutorial, Computer, Vol.29, No.3, March/1996, pp. 31-44. Translated from the original English version and reprinted with permission. (IEEE)
3. Галушкин А. И. Синтез многослойных систем распознавания образов. – М.: «Энергия», 1974.
4. Werbos P. J., Beyond regression: New tools for prediction and analysis in the behavioral sciences. Ph.D. thesis, Harvard University, Cambridge, MA, 1974.
5. Rumelhart D.E., Hinton G.E., Williams R.J., Learning Internal Representations by Error Propagation. In: Parallel Distributed Processing, vol. 1, pp. 318 – 362. Cambridge, MA, MIT Press. 1986.
6. Барцев С. И., Охонин В. А. Адаптивные сети обработки информации. Красноярск: Ин-т физики СО АН СССР, 1986. Препринт N 59Б. – 20 с.
7. Neural Network Based on Ant Colony Clustering Algorithm Applied to Predict the Stability of the Roof in Coal Mining, ISDA '06 Proceedings of the Sixth International Conference on Intelligent Systems Design and Applications – Volume 01, 2006.
8. Evolving Neural Networks Using the Hybrid of Ant Colony Optimization and BP Algorithms, Advances in Neural Networks – ISSN 2006, Lecture Notes in Computer Science, 2006, Volume 3971/2006, 714-722, DOI: 10.1007/11759966_105.
9. Study on the Feed forward Neural Network Based on Ant Colony Optimization, Journal of Wuhan University of Technology [Transportation Science & Engineering], 2009-03.

Abstract

Back propagation of error algorithm is one of the training methods of artificial neural network. Its creation gave an additional incitement to the development of artificial neural network theory. An increase and appearance of the new tasks had led to the appearance of hybrid algorithms based on classical and stochastic ones. They emphasize the advantages and decrease the drawbacks of each other. The article suggests the hybrid training algorithm of neural network based on the ant colony and back propagation of error algorithms. The ant colony algorithm is used to choose the priority ways of moving along the neural network and to change the network balance with the help of pheromones.

In comparison with the basic algorithm the ant colony one is changed. Instead of distance records between each pair of connected nodes and total network error records, they use the record of error value on the each exit node at the end of each ant travel.

In the modified algorithm, each network weight changes the coefficient with the glance of the value of pheromone matrix elements, the error on the each node of exit layer and suggested function for pheromone accounting. The algorithm performance was checked on the famous databases [<http://archive.ics.uci.edu/ml/>]. The results of the experiments show that in comparison with another algorithm the modified one accelerates the training of neural network. The algorithm implementation was approved on the Microsoft SQL Server 2005 in Analysis Services. It will help to apply suggested algorithm to solve another problems of classification

Keywords: neural network, ant colony optimization, back propagation algorithm, hybrid algorithm, neural network training

УДК 004.827

ПРИМЕНЕНИЕ МЕТОДОВ КЛАСТЕРНОГО АНАЛИЗА ДЛЯ ОЦЕНКИ НЕЧЕТКОЙ НЕСОГЛАСОВАННОЙ ЭКСПЕРТНОЙ ИНФОРМАЦИИ

Ю. В. Ульяновская

Кандидат технических наук, доцент
Кафедра информационных систем и технологий
Академия таможенной службы Украины
ул. Дзержинского, 2/4, г. Днепропетровск, Украина,
49000

Контактный тел.: 095-833-79-33, 067-975-5115

E-mail: uyv@rambler.ru

У статті розглядається питання обробки нечіткої експертної інформації. Запропоновано графічне представлення отриманих даних. Розглянуто питання про отримання групової оцінки альтернатив при неузгодженій думці експертів з використанням дискретної функції приналежності та методів кластерного аналізу

Ключові слова: експерти оцінки, нечітка інформація, кластерний аналіз

В статье рассматривается вопрос обработки нечеткой экспертной информации. Предложено графическое представление полученных данных. Рассмотрен вопрос о получении групповой оценки альтернатив при несогласованных мнениях экспертов с использованием дискретной функции принадлежности и методов кластерного анализа

Ключевые слова: экспертные оценки, нечеткая информация, кластерный анализ

Вступлення

Класс задач, решение которых основано на экспертных данных, достаточно велик. К таким задачам относятся задачи планирования, оценки платежеспособности предприятий и физических лиц, оценки рисков, задачи классификации и экспертного оценивания предметов и т.д. При этом экспертная информация

в большинстве случаев носит нечеткий, лингвистический характер. Это обусловлено тем, что человеку более свойственно характеризовать понятия на естественном языке.

В качестве примера можно привести оценку роста человека, для определения которого люди используют нечеткие переменные «высокий», «средний» и др. Таким образом, при экспертном оценивании наиболее