УДК 004.032.26
DOI: 10.25206/1813-8225-2018-162-244-246

Е. О. ВИКУЛОВ

Омский государственный технический университет, г. Омск

ИССЛЕДОВАНИЕ РАСПРЕДЕЛЕНИЯ ДАННЫХ ВЫСОКОНАГРУЖЕННЫХ ВЕБ-ПРИЛОЖЕНИЙ С ПРИМЕНЕНИЕМ НЕЙРОСЕТЕВЫХ ТЕХНОЛОГИЙ

В статье предложен подход к оптимизации распределения данных при проектировании высоконагруженных веб-приложений. Разработана модель, описывающая состояния серверных станций на текущий момент времени, пригодная для осуществления последующей оптимизации распределения данных. Проведен эксперимент по распределению тестовой выборки данных с применением нейросетевых технологий, в частности, с использованием сетей Кохонена; приведены результаты распределения данных по серверным станциям. На основании результатов эксперимента сделаны выводы о возможности применения сетей Кохонена при распределении данных высоконагруженных веб-приложений.

Ключевые слова: распределение данных, высоконагруженные веб-приложения, сервер-балансир нейросетевые технологии, сеть Кохонена, кластеризация.

Введение. В настоящее время одной из актуальных проблем в сфере информационных технологий является передача больших объемов данных по сетям ЭВМ [1]. Для увеличения скорости доставки данных пользователю используют следующие технологии: сети доставки и дистрибуции данных, системы балансировки (распределения) данных [1, 2].

Сети доставки и дистрибуции данных CDN (от англ. *Content Delivery Network*) [2] состоят из географически распределенных многофункциональных платформ, взаимодействие которых позволяет максимально эффективно обрабатывать данные и удовлетворять запросы пользователей при получении данных. Системы балансировки данных — это решения, распределяющие данные с использованием сервера-балансира, который получает заявку на загрузку данных и производит выбор подходящего для хранения данных сервера [3]. Следует отметить, что приведенные технологии не учитывают состояние аппаратного обеспечения серверных станций, его загруженность и скорость работы [4], что сказывается на скорости доставки данных пользователю.

В работе рассмотрен метод обработки и анализа данных о состоянии серверных станций позволяющий увеличить скорость доставки данных, посредствам применения нейросетевых технологий. Нейронные сети традиционно используются при выполнении процедур кластеризации, классификации, прогнозирования [5]. Искусственная нейронная сеть представляет собой математическую модель, построенную по принципу организации и функционирования биологических нейронных сетей — сетей нервных клеток живого организма [5].

В данной работе в качестве основного инструмента моделирования использована нейронная сеть Кохонена [6]. Нейронные сети Кохонена являются одной из разновидностей нейронных сетей, однако принципиально отличаются от остальных нейронных сетей, так как используют неконтролируемое обучение. При таком обучении обучающее множество состоит лишь из значений входных переменных, в процессе обучения нет сравнивания выходов нейронов с эталонными значениями. Такая сеть учится понимать структуру данных. Основной принцип работы сетей — введение в правило обучения нейрона информации относительно его расположения [7].

Постановка задачи оптимизации распределения данных. Для решения задачи оптимизации выбора серверной станции необходимо определить параметры, которые позволят провести оптимальное распределение данных по серверным станциям. В качестве таких параметров принимаются следующие: расстояние от клиента до серверной станции, загруженность жесткого диска, время работы без отказов. На основании этих данных разрабатываемая система должна делать вывод о том, на какую из серверных станций передать загружаемые данные.

Сформируем массив параметров серверных станций. Полученные параметры представим в виде многомерной матрицы (1) [8].

$$X = \begin{bmatrix} \begin{bmatrix} X^1_{11} & X^1_{12} & \cdots & X^1_{1m} \\ X^1_{21} & X^1_{22} & X^1_{2j} & \cdots \\ \cdots & \cdots & X^1_{ij} & \cdots \\ X^1_{n1} & X^1_{n2} & \cdots & X^1_{nm} \end{bmatrix} \cdots \\ \cdots \begin{bmatrix} X^l_{11} & X^l_{12} & \cdots & X^l_{1m} \\ X^l_{21} & X^l_{22} & X^l_{2j} & \cdots \\ \cdots & \cdots & X^l_{ij} & \cdots \\ X^l_{n1} & X^l_{n2} & \cdots & X^l_{nm} \end{bmatrix} \cdots \begin{bmatrix} X^q_{11} & X^q_{12} & \cdots & X^q_{1m} \\ X^q_{21} & X^q_{22} & X^q_{2j} & \cdots \\ \cdots & \cdots & X^q_{ij} & \cdots \\ X^q_{n1} & X^q_{n2} & \cdots & X^q_{nm} \end{bmatrix} \end{bmatrix}, \quad (1)$$

где верхний индекс $l = \overline{1,q}$ — номер заявки на загрузку файла, q — количество заявок. Нижние индексы i, j характеризуют состояние серверов, $i = \overline{1,n}$ — номер серверной станции, n — количество серверных станций, равное количеству кластеров, j — номер параметра функционирования, m — количество параметров. Параметрами системы являются: x^l_{i1} — загруженность жесткого диска i-ой серверной станции, x^l_{i2} — расстояние от клиента до i-ой серверной станции, x^l_{i3} — время работы без отказов i-ой серверной станции.

Каждый из показателей нормируем от 0 до 1. Для того чтобы облегчить процедуру кластеризации, представим вектор входных данных S^l_n в виде суммы нормированных параметров функционирования $S^l_n = \sum_{j=1}^{m} X^l_{nj}$.

Таким образом, входными данными для нейронной сети будут являться:

— вектор входных значений, переданный системе при получении новой заявки на распределение данных, S^l_n;

— количество серверных станций, n.

Рассмотрим алгоритм кластеризации данных при помощи нейронной сети. Этот алгоритм принимает на входе n-мерный вектор входных значений S^l_n, представляющих собой параметры серверных станций. Значения этих параметров будем трактовать как величины импульсов, поступающих на вход нейрона через n входных синапсов. Поступающие в нейрон импульсы складываются со своими весовыми коэффициентами (весами w_0, w_1, …, w_n). Если вес положительный, то соответствующий синапс возбуждающий, если отрицательный — то тормозящий. Если суммарный импульс превышает заданный порог активации w_0, то нейрон возбуждается и выдаёт на выходе 1, иначе выдаётся 0. Таким образом, нейрон вычисляет булеву функцию вида

$$a(s) = \varphi\left(\sum_{i=1}^{n} s^l_i w_i - w_0\right).$$

Здесь φ — ступенчатая функция Хэвисайда [8], которую можно описать как:

$$\varphi = \begin{cases} 0, \text{ если } \sum_{i=1}^{n} s^l_i w_i - w_0 \le \sigma; \\ 1, \text{ если } \sum_{i=1}^{n} s^l_i w_i - w_0 > \sigma, \end{cases}$$

где σ — начальное расстояние между соседними нейронами [9].

В теории нейронных сетей функцию φ, преобразующую значение суммарного импульса в выходное значение нейрона, принято называть функцией активации [9]. Рассмотрим алгоритм последовательного обучения нейронной сети Кохонена. Обучение состоит из последовательности коррекций векторов, представляющих собой нейроны. На каждом шаге обучения выбирается нейрон, который наиболее похож на вектор входов. Под похожестью в данной задаче понимается расстояние между векторами.

После того, как найден наиболее похожий нейрон, производится корректировка весов нейросети в соответствии с выражением:

$$w_i(t+1) = w_i(t) + \eta(t)\, h_{ci}(t)\left[S^l_i(t) - w_i(t)\right],$$

где η — скорость обучения, h_{ci} — функция расстояния от i-го нейрона до текущего центра кластера, $S^l_n(t)$ — вектор входных значений. При этом вектор, описывающий центр кластера, и вектора, описывающие его соседей, перемещаются в направлении входного вектора.

Решение задачи оптимизации распределения данных на основе методов кластеризации. На начальном этапе существования системы распределения данных все серверные станции имеют одинаковые показатели загруженности и безотказной работы, поэтому распределять данные в этом случае уместно, оценивая лишь показатель расстояния до серверной станции.

После загрузки определенного количества данных в систему показатели серверных станций начнут отличаться друг от друга. Промоделируем такую ситуацию при помощи программной среды MATLAB [10]. Рассмотрим тестовую выборку, состоящую из 150 случаев загрузки данных. Разделим тестовую выборку на 3 группы:

1-я группа: серверная станция слабо загружена, имеет среднее время безотказной работы и находится на среднем расстоянии от пользователя (кластер G_1);

2-я группа: серверная станция имеет большую загрузку, низкое время безотказной работы и находится на большом расстоянии от пользователя (кластер G_2);

3-я группа: серверная станция имеет средний уровень загрузки, высокое время безотказной работы и находится на близком расстоянии от серверной станции (кластер G_3).

На рис. 1 представлен результат кластеризации тестовой выборки по каждому из критериев. Здесь обозначены различные степени принадлежности

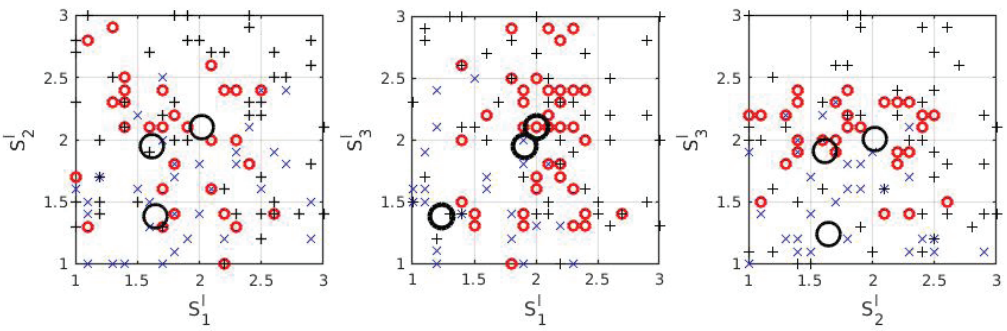

Рис. 1. Кластеризация тестовой выборки

Таблица 1

Результаты кластеризации

№ эксперимента	Обобщенные показатели функционирования серверов			Результат распределения по кластерам	
	S_1^l	S_2^l	S_3^l	Нейронная сеть	Экспертная оценка
1	1,2	0,8	2,6	G_2	G_2
2	2,1	0,9	1,1	G_2	G_2
3	0,7	0,8	1,9	G_2	G_2
4	1,1	0,9	1	G_1	G_3
5	1,2	1,2	1,1	G_1	G_3
6	2,6	2,8	2,7	G_3	G_3
7	2,6	2,8	1,3	G_3	G_3
8	0,8	2,6	2,2	G_1	G_1
9	1,2	2,2	2	G_1	G_1
10	1,1	1	1	G_1	G_1

каждой заявки к одному из кластеров. Так же отражены центры каждого из кластеров.

Используем полученную после кластеризации тестовой выборки нейронную сеть для распределения новых заявок на загрузку данных. Предположим, что в определенный момент времени в систему поступает десять заявок на загрузку данных и необходимо принять решение об их распределении. Для принятия решения проведем кластерный анализ отправленных заявок. Полученные результаты представим в виде таблицы (табл. 1). Также в таблице представлена экспертная оценка. Для двух наборов данных (4 и 5 из табл. 1) результат выбора серверной станции с помощью кластеризации отличается от экспертной оценки. Это говорит о том, что для рассматриваемых наборов данных границы между кластерами достаточно малы и оба кластера G_1 и G_3 подходят для выбора.

Решение, принятое с помощью нейронной сети, в данном случае имеет преимущество перед решением, основанным на экспертной оценке, так как базируется на результатах анализа тестовой выборки. Кроме того, преимущество использования нейронной сети особенно видно при обработке больших объемов данных, где не обойтись без автоматизированной оценки.

Заключение. Рассмотрен подход к обработке информации при проектировании высоконагруженных систем распределения данных. Разработана модель, позволяющая описывать состояния серверных станций на текущий момент времени с помощью наборов характерных параметров. Предложена структура представления параметров текущего состояния системы, пригодная для осуществления оптимизации выбора серверной станции. Проведен эксперимент кластеризации данных для распределения по серверам при помощи нейронной сети Кохонена.

Анализ показал, что результаты, полученные с помощью предлагаемого алгоритма кластеризации (при помощи нейронной сети Кохонена), отличаются от основанных на экспертной оценке (не учитывающей информации о состоянии серверов). Это свидетельствует об эффективности разработанного алгоритма кластеризации, позволяющего принять решение на основе данных о текущих параметрах системы. Следует подчеркнуть, что даже на небольших объемах тестовой выборки интеллектуальные автоматизированные системы позволяют получать более точные результаты. При увеличении объемов обрабатываемых данных в реальных условиях работы приложения использование предлагаемого метода позволит повысить точность результатов.

Библиографический список

1. What is Load Balancing? How Load Balancing Work // NGINX. URL: https://www.nginx.com/resources/glossary/load-balancing/ (дата обращения: 12.09.2018).

2. Load Balancing Techniques Algorithms // KEMP Applications Delivery. URL: https://kemptechnologies.com/load-balancer/load-balancing-algorithms-techniques/ (дата обращения: 12.09.2018).

3. Server Load Balancing // Akamai. URL: https://www.akamai.com /us/en/resources/server-load-balancing.jsp (дата обращения: 23.09.2018).

4. Basic Load Balancing // IBM Cloud. URL: https://console.bluemix.net/docs/infrastructure/loadbalancer-service/basic-load-balancing.html (дата обращения: 23.09.2018).

5. Хайкин С. Нейронные сети. Полный курс. Изд. 2-е, испр. М.: Вильямс, 2017. 1103 с. ISBN 978-5-8459-2069-0.

6. Уоссермен Ф. Нейрокомпьютерная техника: Теория и практика / пер. с англ. Ю. А. Зуева, В. А. Точенова. М.: Мир, 1992. 184 с. ISBN 5-06-004094-1.

7. Каллан Р. Основные концепции нейронных сетей: пер. с англ. А. Г. Сивака. М.: Вильямс, 2001. 287 с. ISBN 5-8459-0210-X.

8. Vikulov E. O., Denisov O. V., Denisova L. A. Data distribution system: preparation of server stations data // IOP Conf. Series: Journal of Physics. Conf. Ser. 2018. Vol. 1050. 012097. DOI: 10.1088/1742-6596/1050/1/012097.

9. Круглов В. В., Борисов В. В. Искусственные нейронные сети. Теория и практика. М.: Горячая линия — Телеком, 2001. 382 с. ISBN 5-93517-031-0.

10. Штовба С. Д. Проектирование нечетких систем средствами MATLAB. М.: Телеком, 2007. 288 с. ISBN 5-93517-359-X.

ВИКУЛОВ Егор Олегович, аспирант кафедры «Автоматизированные системы обработки информации и управления».
SPIN-код: 1835-5804
AuthorID (РИНЦ): 905246
Адрес для переписки: vikuloveo@gmail.com

Для цитирования

Викулов Е. О. Исследование распределения данных высоконагруженных веб-приложений с применением нейросетевых технологий // Омский научный вестник. 2018. № 6 (162). С. 244—246. DOI: 10.25206/1813-8225-2018-162-244-246.

Статья поступила в редакцию 28.10.2018 г.