Прогнозирование загрузки ресурсов кластера при помощи нейросетевых моделей

Ю.С. Артамонов

Самарский национальный исследовательский университет имени академика С.П. Королева, 443086, Московское шоссе, 34, Самара, Россия

Аннотация

В настоящее время исследователю для проведения вычислений доступен широкий набор высокопроизводительных окружений. Выбрать окружение, в котором вычисления будут завершены как можно раньше, – довольно сложная задача. Для её решения требуется проанализировать загрузку ресурсов окружения, а также спрогнозировать их доступность в будущем.

В работе решена задача прогнозирования загрузки ресурсов кластера с использованием нейросетевых моделей. Рассмотрен процесс настройки архитектуры сети на примере многослойного персептрона: выбор функций активации, алгоритмов инициализации и обновления весов нейронов. Обучение и тестирование проведено на наборе данных загрузки кластера «Сергей Королев» за период с ноября 2013 года по декабрь 2016 года.

Ключевые слова: загрузка ресурсов; кластер; прогнозирование; нейронная сеть; модель;

1. Введение

В последнее время многие исследования посвящены прогнозированию загрузки различных вычислительных ресурсов, таких как ядра CPU [1], отдельные узлы кластера или облака [2]. Задача прогнозирования загрузки вычислительных ресурсов является актуальной, поскольку от эффективного прогнозирования зависит решение задач планирования использования ресурсов, периодов их обслуживания и модернизации. Без прогнозирования доступности разделяемых ресурсов невозможно эффективное использование, например, кластерных систем, где пользователи кластера совместно используют узлы с различными характеристиками, занимая их частично или полностью своими вычислениями.

В предыдущей работе [3] мы решили задачу прогнозирования загрузки вычислительных ресурсов при помощи модели EMMSP и рассмотрели применимость этой модели. Модель показала себя хорошо только на специфических данных и участках истории загрузки, но вместе с тем мы продемонстрировали, что она может быть эффективно использована в простейших моделях адаптивной композиции с другими моделями. В этой работе рассматривается использование нейросетевых моделей для прогнозирования загрузки ресурсов кластера и сравнивается этот подход с продемонстрированным ранее.

2. Нейросетевые модели прогнозирования

Нейросетевые модели прогнозирования базируются на использовании нейронных сетей, которые могут обучаться в задачах регрессии и на основании некоторых входных параметров выдавать значение на выходе, аппроксимируя неизвестную функциональную зависимость выходных данных от входных.

Нейросетевые модели были использованы в работах [4] и [5] для прогнозирования загрузки различных по своей природе ресурсов: CPU серверов и электрических сетей. В обеих задачах нейросетевые модели показали хорошие результаты и были признаны эффективными и адекватными задаче прогнозирования. Учитывая эти результаты, рассмотрим, насколько хорошо нейросетевые модели подойдут для прогнозирования количества загруженных узлов кластера.

Нейросетевые модели были выбраны нами для изучения исходя из особенностей задачи прогнозирования загрузки узлов кластера и исторических данных, собранных нами. А именно:

- временные ряды загрузки ресурсов являются нестационарными,
- в данных есть шаблоны и периодические составляющие, участки высокой и низкой загрузки, соответствующие выходным и рабочим дням,
- временные ряды имеют известные заранее минимальное и максимальное значение.

Для прогнозирования значений временных рядов такой природы можно использовать нейронные сети, решая задачу аппроксимации неизвестной функции. Принимая во внимание работы [6] и [7], применяющие нейронные сети для решения задач прогноза сходных по природе временных рядов (загрузка технических ресурсов кластера/облака), мы

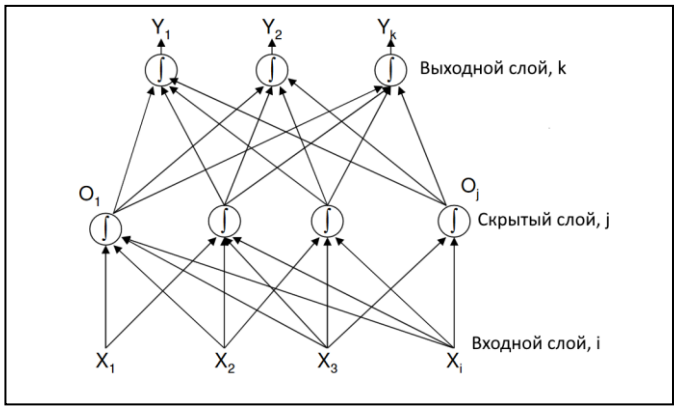
выбрали для исследования модель многослойного персептрона (MLP) с одним (SL MLP) и двумя скрытыми слоями (DL MLP).

Структура MLP состоит из нейронов и связей между ними (рис. 1). Нейроны имеют заданную функцию преобразования – активации, а связи – веса. В MLP выходной сигнал некоторого слоя Z можно описать уравнением 1[8]:

$$z_j = f(\sum_{i=1}^{N} w_{ij} u_i) \quad (1)$$

где u_i – входные сигналы слоя Z, w_{ij} – веса связей нейронов между *i-ым* нейроном предыдущего слоя и *j-ым* нейроном слоя Z, f – функция активации, z_j – выходной сигнал нейрона. В работе мы использовали в качестве функции активации нейронов скрытого слоя функцию гиперболического тангенса.

Обучение нейронной сети заключается в изменении весов связей между нейронами, и задача алгоритма обучения состоит в нахождении такой конфигурации весов всех связей, где будет минимизирована функция ошибки. В поставленной задаче прогнозирования используется функция ошибки – MSE (Mean Squared Error) в методе



градиентного спуска, в качестве итоговой ошибки для сравнения моделей используется MAE (Mean Average Error).

Рис. 1. Структура MLP с одним скрытым слоем.

Для обучения и тестирования моделей на основе MLP мы используем библиотеку DeepLearning4j, которая предоставляет инструментарий для работы с нейронными сетями различных конфигураций. В состав этой библиотеки включены наиболее популярные архитектуры нейронных сетей, алгоритмы обучения и оптимизации. Библиотека написана на языке Java и использует нативные расширения для вычислений на CPU и GPU, чтобы обеспечить требуемую производительность [9]. Библиотека DeepLearning4j распространяется на условиях лицензии Apache License 2.0, что позволяет применять её в любых приложениях, в том числе коммерческих, а открытый исходный код позволяет привлечь большое количество исследователей и улучшить качество библиотеки.

3. Настройка архитектуры сети и параметров обучения

Для обучения нейронных сетей семейства MLP используется метод обратного распространения ошибки (Back propagation) с различными модификациями. Метод представляет собой итеративный градиентный алгоритм, который применяется для минимизации ошибки работы MLP и получения желаемых выходных значений. Суть метода заключается в распространении сигналов ошибки от выходов сети к её входам, обратно прямому распространению сигналов в обычном режиме работы [10].

Существенными параметрами метода и его модификаций являются:

- количество эпох обучения,
- коэффициент обучения,
- алгоритм инициализации весов связей,
- алгоритм обновления весов,
- алгоритм оптимизации,
- момент обучения.

В поставленной задаче необходимо спрогнозировать количество занятых узлов кластера в нескольких наиболее интенсивно используемых группах узлов. Целевое время прогноза – 12 часов, требуется спрогнозировать 12 значений временного ряда, по одному среднему значению занятых узлов в группе на один час. Для обучения и прогнозирования были выбраны данные по группам узлов qdr_tmp и ddr_tmp. Их загрузка представляет наибольший интерес в связи с большой регулярной загрузкой.

Для сравнения методов обучения с различными модификациями нами были подобраны параметры обучения, представленные в таблице 1. Мы сравнивали обучение моделей SL MLP и DL MLP на задаче прогнозирования 12 точек загрузки кластера (каждая точка – среднее количество занятых узлов группы кластера за 1 час). При обучении и прогнозировании на вход нейронным сетям подавались только данные временных рядов. Оптимальное количество входов, подобранное экспериментально, равно 6.

В процессе обучения на вход нейронной сети подавались всевозможные наборы последовательных 6 значений ряда, при этом тестовые наборы шли в случайном порядке. Для каждого тестового набора на выходе нейронной сети формировалось 12 значений, которые сравнивались с 12 значениями из тестового набора. Параметры $i = 6$, $k = 12$.

Таблица 1. Экспериментально подобранные параметры обучения

Модель	Количество эпох обучения	Коэффициент обучения	Момент обучения	Количество нейронов входного слоя	Количество нейронов скрытых слоёв
SL MLP	300	0.01	0.9	6	15
DL MLP	400	0.01	0.9	6	1ый: 20 2ой: 10

Метод обратного распространения ошибки подвержен следующим проблемам:

- медленная сходимость,
- сходимость к локальным минимума,
- переобучение.

Модификации метода обратного распространения ошибки с моментами и различными алгоритмами обновления весов связей, такими как Adadelta, позволяют бороться с приведёнными выше проблемами, ускорить обучение и уменьшить ошибку работы MLP.

В рамках исследования нейросетевых моделей мы рассмотрели различные конфигурации обучения нейронных сетей методом обратного распространения ошибки. В качестве параметров конфигурации в обучении выступали: алгоритм инициализации весов нейронов, алгоритм обновления весов нейронов и алгоритм оптимизации.

Мы протестировали 2 варианта инициализации весов: равномерным распределением (Uniform) и по методу Xavier . В качестве алгоритма обновления весов были протестированы: алгоритм Nesterov Accelerated Gradient (Nesterovs), адаптивный градиентный спуск (Adagrad), метод адаптивного шага обучения (Adadelta), адаптивная оценка моментов (Adam). Были опробованы 2 алгоритма оптимизации: линейный градиентный спуск (LGD) и стохастический градиентный спуск (SGD). Эти оптимизации и параметры метода градиентного спуска и алгоритма обратного распространения ошибки описаны в работе [11].

В тесте использовалась обучающая выборка длины 6000 точек и тестовая с длиной 1000 точек, данные выборки получены за период с 1 января 2015 года по 1 января 2016 года. Результаты тестирования моделей с различными параметрами обучения для решения задачи прогнозирования загрузки кластера представлены в таблице 2, значения ошибки RMSE (Root Mean Square Error) приведены для оценки разброса прогнозных значений.

В таблице 2 представлены результаты тестирования модификаций метода обратного распространения ошибки, 3 лучших результата для каждой модели выделены подчёркиванием. Из приведённых результатов мы можем сделать вывод, что наиболее эффективными модификациями метода обратного распространения ошибки для задачи прогнозирования загрузки кластера являются:

1. стохастический градиентный спуск с инициализацией весов равномерным распределением и обновлением весов алгоритмом ADAM – как для SL MLP, так и для DL MLP,

2. линейный градиентный спуск с инициализацией весов равномерным распределением и обновлением весов по методу Нестерова с моментами – для SL MLP,

3. стохастический градиентный спуск с инициализацией весов по методу Xavier и обновлением весов алгоритмом ADAM – для DL MLP.

Результат DL и SL MLP отличается незначительно, что, вероятно, обусловлено особенностью тестовых данных.

Таблица 2. Значения ошибок RMSE и MAE в зависимости от конфигурации обучения нейронной сети

Алгоритм инициализации весов	Алгоритм обновления весов	Алгоритм оптимизации	SL MAE	DL MAE	SL RMSE	DL RMSE
UNIFORM	NESTEROVS	LGD	8,03	7,99	9,28	9,24
XAVIER	NESTEROVS	LGD	8,05	8,20	9,30	9,38
UNIFORM	NESTEROVS	SGD	8,02	7,64	9,28	8,94
XAVIER	NESTEROVS	SGD	8,06	7,70	9,32	8,97
UNIFORM	ADADELTA	LGD	9,71	11,15	10,99	12,11
XAVIER	ADADELTA	LGD	9,62	11,77	10,78	12,75
UNIFORM	ADADELTA	SGD	15,09	8,33	16,08	9,86
XAVIER	ADADELTA	SGD	17,44	12,52	18,61	14,64
UNIFORM	ADAGRAD	LGD	13,24	11,52	13,61	12,64
XAVIER	ADAGRAD	LGD	15,24	9,70	16,36	12,13
UNIFORM	ADAGRAD	SGD	19,04	10,60	21,36	19,23
XAVIER	ADAGRAD	SGD	17,21	11,21	18,12	17,22
UNIFORM	ADAM	LGD	8,10	7,83	9,34	9,13
XAVIER	ADAM	LGD	8,14	7,91	9,38	9,18
UNIFORM	ADAM	SGD	7,99	7,54	9,26	8,84
XAVIER	ADAM	SGD	8,09	7,56	9,34	8,85

4. Сравнение ошибок моделей

Пример прогнозирования данных загрузки кластера нейронной сетью с одним скрытым слоем приведен на рис. 2, нейронной сетью с двумя скрытыми слоями – на рис. 3. Пунктиром показаны прогнозные значения ряда. Графики прогнозных значений были получены вычислением прогноза через каждые 12 точек.

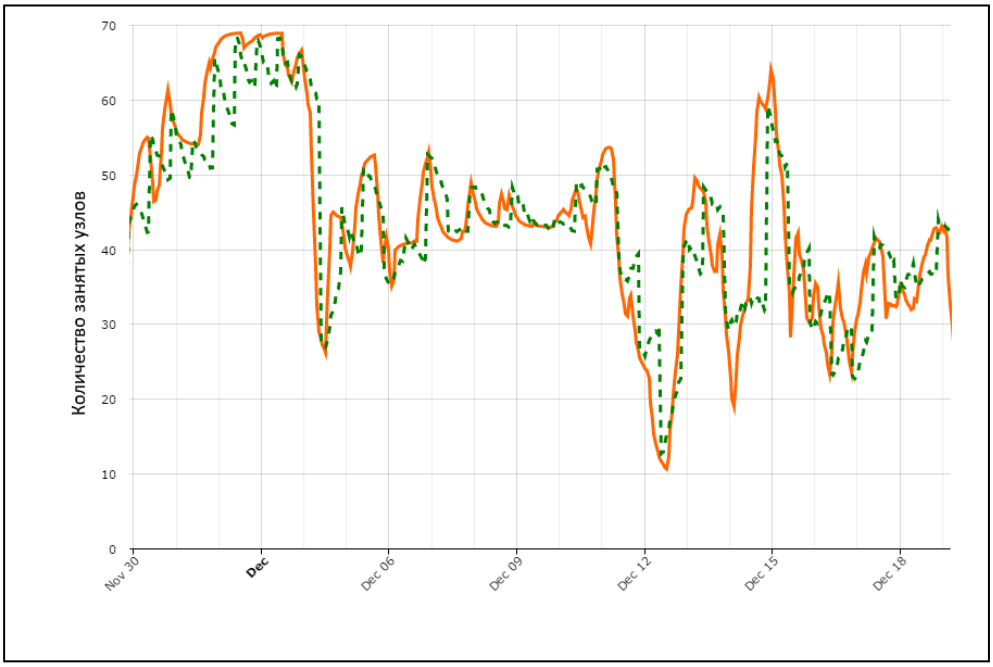


Рис. 2. Прогноз загрузки ресурсов кластера «Сергей Королёв» при помощи SL MLP.

В качестве итоговой метрики ошибок выбрана средняя абсолютная ошибка (MAE), поскольку относительная ошибка прогноза (MAPE) не может быть использована в рядах, включающих значения близкие или равные нулю. Распределение ошибок MAE моделей SL MLP и DL MLP представлено на рис. 4, распределение ошибок обеих моделей близко к нормальному.

Ранее, задача прогнозирования 12 точек загрузки кластера решалась методом прогнозирования временных рядов по выборке максимального подобия (EMMSP) [3]. Ошибки MAE прогнозирования для сравнения методов приведены в таблице 3.

Кроме прямого сравнения моделей мы попробовали задействовать все три модели (EMMSP, SL MLP, DL MLP) вместе. Для этого, была выдвинута гипотеза: каждая из моделей является лучшей (показывает наименьшую ошибку MAE) на некотором участке данных длинной L > M, где M – количество точек прогнозирования. Мы проверили эту гипотезу для данных, на которых производилось тестирование моделей MLP. Каждая из моделей сохраняет своё лидерство в среднем на участке длиной 24 – 36 точек, что соответствует временному отрезку 1 – 1.5 дня.

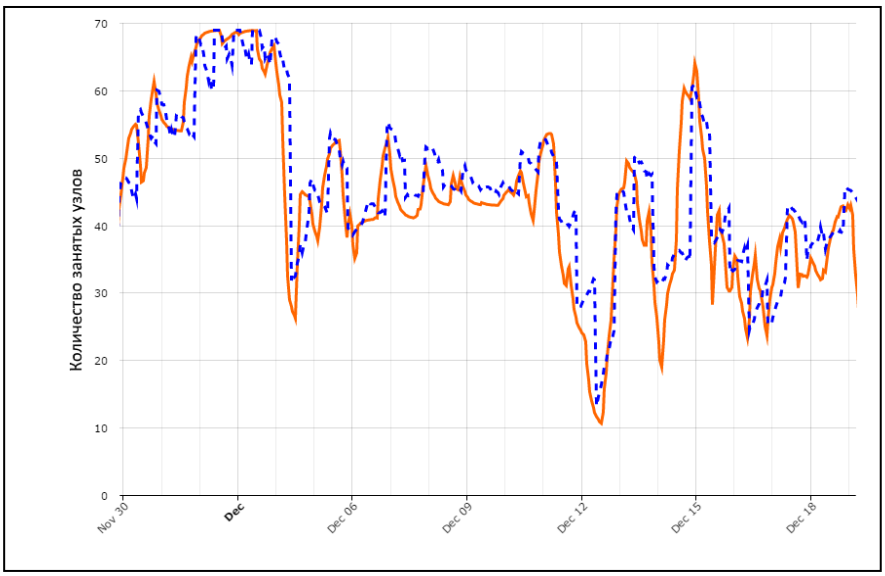


Рис. 3. Прогноз загрузки ресурсов кластера «Сергей Королёв» при помощи DL MLP.

Таблица 3. Значение ошибки MAE для различных моделей прогнозирования

Модель	EMMSP	SL MLP	DL MLP	Простая адаптивная селективная модель
MAE	8.7	8.02	7.54	6.8

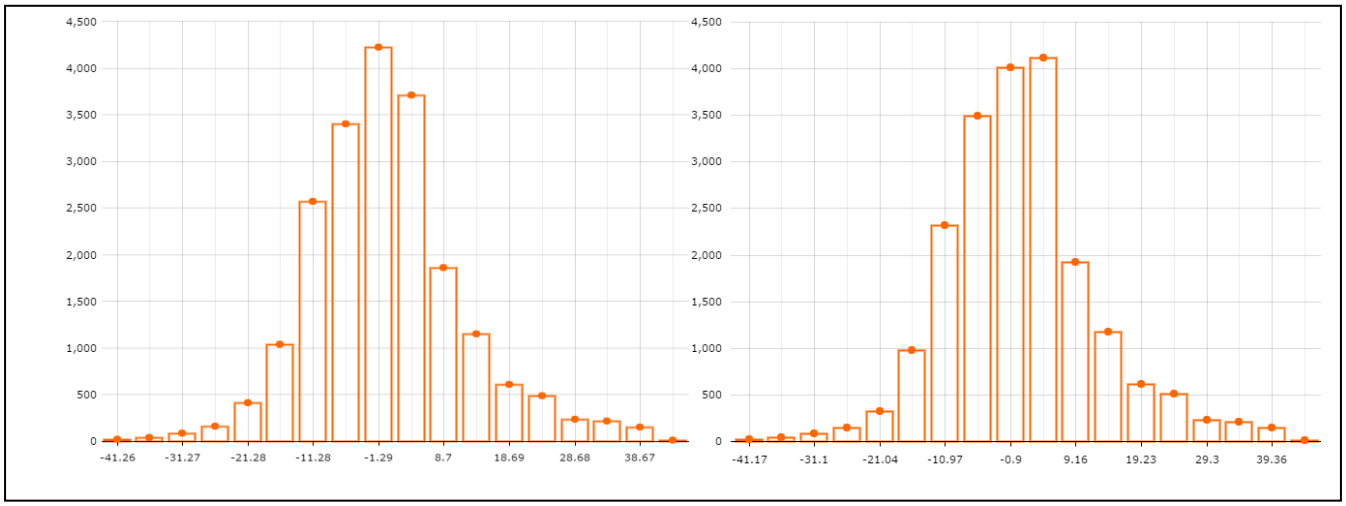


Рис. 4. Распределение средней абсолютной ошибки моделей с одним скрытым слоем (слева) и двумя скрытыми слоями (справа).

Значение ошибки для простой адаптивной селективной модели [12] было получено для модели, которая выбирает наилучшую модель для прогнозирования будущих значений по простому эвристическому правилу: если какая-то из моделей была лучше на предыдущем участке данных, то её следует использовать для прогнозирования снова.

Данные для тестирования были собраны в период с ноября 2013 года по декабрь 2016 года. Открытые данные мониторинга загрузки кластера «Сергей Королев» доступны в машиночитаемом формате JSON по адресу: http://templet.ssau.ru/wiki/открытые_данные

5. Заключение

Алгоритмы прогнозирования на основе нейросетевых моделей с одним и двумя скрытыми слоями интегрированы в сервис Templet Web, что позволяет пользователям оценить время запуска задачи. Графики прогноза и истории загрузки кластера доступны зарегистрированным пользователям системы. В будущем планируется предоставить пользователям интерактивную подсказку о количестве доступных ресурсов и оценке времени запуска задачи на основе требований к кластеру (узлов, групп, лицензий на ПО), указанных в задаче на момент запуска.

Результаты прогнозирования загрузки кластера могут быть применены для решения нескольких типов задач:

- повышение эффективности использования кластера (энергоэффективность, повышение загрузки),
- выбор оптимальных окружений и параметров для расчётов,
- планирование развития кластера и периодов его обслуживания.

Методы прогнозирования загрузки вычислительных ресурсов наиболее востребованы сейчас в облачных окружениях, где они могут позволить коммерческим компаниям снизить затраты на обслуживание серверов или же наоборот эффективно приспосабливаться к растущим требованиям клиентов.

Литература

[1] Naseera, S. Host CPU Load Prediction Using Statistical Algorithms a comparative study / S. Naseera, G.K. Rajini, P. Sunil Kumar Reddy // International Journal of Computer Technology and Applications – 2016. – 9(12). – P. 5577-5582.

[2] Di, S. Host load prediction in a Google compute cloud with a Bayesian model / S. Di, D. Kondo, W. Cirne // Proceedings of the International Conference on High Performance Computing, Networking, Storage and Analysis. – IEEE Computer Society Press, 2012. – P. 21.

[3] Артамонов, Ю. С. Применение модели EMMSP для прогнозирования доступных вычислительных ресурсов в кластерных системах // Известия Самарского научного центра РАН. – 2016. – том 18, № 4 (4). – С. 681-687.

[4] Naseera, S. A comparative study on CPU load predictions in a computational grid using artificial neural network algorithms / S. Naseera, G.K. Rajini, N. Amutha Prabha, G. Abhishek // Indian Journal of Science and Technology. – 2015. – Т. 8. – №. 35.

[5] Kalaitzakis, K. Short-term load forecasting based on artificial neural networks parallel implementation / K. Kalaitzakis, G. Stavrakakis, E.M. Anagnostakis // Electric Power Systems Research. – 2002. – Т. 63. – №. 3. – P. 185-196.

[6] Chandini, M. A Brief study on Prediction of load in Cloud Environment / M. Chandini, R. Pushpalatha, R. Boraia // International Journal of Advanced Research in Computer and Communication Engineering. – 2016. – 5(5). – P. 157-162.

[7] Engelbrecht, H. A. Forecasting methods for cloud hosted resources, a comparison / H.A. Engelbrecht, M. van Greunen // Network and Service Management (CNSM), 2015 11th International Conference on. – IEEE, 2015. – P. 29-35.

[8] Хайкин, С. Нейронные сети. – М.: Вильямс, 2006. – 1104 с.

[9] Deeplearningj: Open-source distributed deep learning for the JVM [Электронный ресурс]. – Режим доступа: http://deeplearning4j.org (01.01.2017)

[10] Осовский, С. Нейронные сети для обработки информации. – М.: Финансы и статистика, 2002. – 344 с.

[11] Ruder, S. An overview of gradient descent optimization algorithms //arXiv preprint arXiv: 1609.04747. – 2016.

[12] Лукашин, Ю.П. Адаптивные методы краткосрочного прогнозирования временных рядов. – М.: Финансы и статистика, 2003. – 415 с.