

МАШИННОЕ ОБУЧЕНИЕ ДЛЯ ПРОГНОЗИРОВАНИЯ ТРАФИКА В СЕТИ LTE

Трошин А.В.

Поволжский государственный университет телекоммуникаций и информатики, Самара, РФ

E-mail: a.v.troshin77@yandex.ru

Прогнозирование сетевого трафика играет ключевую роль в организации управления ресурсами и планировании развития современных телекоммуникационных сетей. Традиционно прогнозирование трафика строилось на основе статистических авторегрессионных моделей, которые устанавливали корреляционные связи между текущими объемами трафика и его возможными будущими значениями. Однако данные модели имеют существенные ограничения, в частности, большинство из них применимы только к стационарным статистическим данным, что сужает их область применения до сетей простой структуры со стабильными во времени объемами потребляемого трафика. Большинство современных сетей имеют сложную структуру и подвержены быстрым изменениям потоков трафика. Для преодоления ограничений традиционных моделей и улучшения прогнозирования в последнее время предпринимаются активные усилия по использованию технологий машинного обучения. В статье рассмотрены основные методы машинного обучения на основе искусственных нейронных сетей в сравнении с традиционными статистическими подходами, а также их практическое применение для прогнозирования трафика в LTE-сети.

Ключевые слова: машинное обучение, искусственные нейронные сети, временные ряды, прогнозирование трафика, статистические методы

Введение

Бурный рост современных телекоммуникационных сетей, внедрение новых технологий и видов услуг, постоянное повышение требований к производительности и задержкам в передаче трафика делают необходимым тщательное планирование архитектуры сети и оперативное управление сетевыми ресурсами с целью обеспечения ее функционирования на высоком качественном уровне [1–5].

Прогнозирование сетевого трафика играет ключевую роль в сетевом планировании и управлении, его целью является построение регрессионной модели, позволяющей предсказывать возможные значения объемов трафика на основе отслеживаемых в хронологическом порядке значений. Данные модели, как правило, относятся к области анализа временных рядов и традиционно их строят на основе статистических методов [5; 7; 8].

Такие методы пытаются установить корреляционные связи, а также различного рода неявные закономерности между предыдущими и текущими значениями временных параметров, чтобы на их основе прогнозировать будущие значения. Они широко применяются в экономических прогнозах, при прогнозировании биржевых котировок и уровней цен, а также во множестве других областей, где есть необходимость в прогнозировании отслеживаемых во времени параметров [1].

Существенным недостатком данных методов является их ограниченность в прогнозировании

только стационарных или «слабостационарных» временных последовательностей. Поскольку для современных бурно развивающихся сетей стационарные во времени значения трафика являются исключением, то данные методы для прогнозирования трафика имеют весьма узкое применение [5].

В последние годы машинное обучение находит все большее применение в областях, где ранее основные усилия были направлены на построение аналитических и статистических моделей: компьютерного зрения, автоматического управления, обработки естественных языков и т. д. Методы машинного обучения также являются продуктивными и для анализа временных рядов, поскольку позволяют строить модели, самостоятельно устанавливающие корреляционные связи и закономерности во временных параметрах за счет их обучения на больших объемах данных [2–6].

Статья посвящена практическому анализу методов машинного обучения для прогнозирования трафика на примере LTE-сети.

Статистические методы

К наиболее широко используемым статистическим методам прогнозирования временных рядов относятся [1; 5]:

1. Авторегрессионная модель AR.
2. Модель скользящего среднего MA.
3. Авторегрессионная модель скользящего среднего ARMA.

4. Авторегрессионная интегрированная модель скользящего среднего ARIMA.

В модели AR (AutoRegression) текущее значение величины представляется в виде следующего ряда [1]:

$$z_t = \sum_{i=1}^p \varphi_i z_{t-i} + a_t, \quad (1)$$

где z_{t-i} – значения случайной величины в моменты времени $t-i$; φ_i – весовые коэффициенты модели; a_t – белый шум. Полученный в результате данного представления статистический процесс называют авторегрессионным порядка p и обозначают как AR(p).

В модели MA (Moving Average) используется следующее представление [1; 7]:

$$z_t = a_t - \sum_{j=1}^q \theta_j a_{t-j}, \quad (2)$$

где θ_j – весовые коэффициенты модели; a_{t-j} – значения белого шума в моменты времени $t-j$. Полученный стохастический процесс называют процессом скользящего среднего порядка q и обозначают как MA(q).

С целью более гибкого представления временных статистических процессов на практике модели AR и MA объединяют на основе выражений (1), (2), получая следующее представление [1; 7]:

$$z_t = \sum_{i=1}^p \varphi_i z_{t-i} + a_t - \sum_{j=1}^q \theta_j a_{t-j}. \quad (3)$$

Данная модель получила название ARMA (AutoRegressive Moving Average) порядков p и q с обозначением ARMA(p, q).

Модели AR, MA и ARMA позволяют представлять только стационарные стохастические процессы [1; 7]. Однако для практического использования наибольший интерес вызывают нестационарные процессы с изменяющимися параметрами во времени. Для представления таких процессов наибольшее распространение получила модель ARIMA (AutoRegressive Integrated Moving Average), основанная на том, что некоторые нестационарные процессы можно привести к стационарному виду путем дифференцирования и использования для их представления модели ARMA [1; 7–8]:

$$\nabla^d z_t = \sum_{i=1}^p \varphi_i z_{t-i} + a_t - \sum_{j=1}^q \theta_j a_{t-j}, \quad (4)$$

где d – степень дифференцирования.

Представление для исходного процесса получается путем интегрирования выражения (4) d раз. Таким образом порядок модели обозначается как ARIMA(p, d, q).

Сложность использования ARIMA на практике связана с необходимостью вначале определить необходимую степень дифференцирования для обеспечения стационарности процесса, а затем установить необходимый порядок модели. Для этого используется методология Бокса – Дженкинса или решетчатый поиск. После этого требуется подобрать $p + d + q + 2$ параметров на основе имеющихся данных [1; 8].

Машинное обучение на основе нейронных сетей

Машинное обучение является одной из парадигм области искусственного интеллекта, в основе которой лежит получение алгоритмов решения какой-либо задачи непосредственно из представленных данных. Методы машинного обучения можно разделить на две основные группы [2–4]:

- обучение с учителем (supervised learning), когда обучение производится на предварительно размеченных данных;
- обучение без учителя (unsupervised learning), когда алгоритм самостоятельно выявляет полезные качества в необработанных данных.

Для задач прогнозирования (линейной регрессии) используются методы первой группы [2–3].

В основе большинства современных методов машинного обучения лежит использование искусственных нейронных сетей ANN (Artificial Neural Networks) для аппроксимации некоторой функции f , связывающей входные аргументы x с прогнозируемыми значениями на ее выходе y [2–4]. Наиболее распространенными типами ANN являются [2–4]:

- многослойный перцептрон MLP (MultiLayer Perceptron);
- сверточная нейросеть CNN (Convolutional Neural Network);
- рекуррентная нейросеть RNN (Recurrent Neural Network).

Сеть MLP состоит из нескольких слоев искусственных нейронов: входного слоя, одного или нескольких скрытых (внутренних) слоев, выходного слоя, см. рисунок 1 [2–4].

Ключевой особенностью MLP является то, что каждый нейрон последующего слоя связан со всеми нейронами предыдущего слоя. Нейроны скрытого слоя $\mathbf{h} = \{h_1, h_2, \dots, h_n\}$ могут быть описаны как [2]:

$$h = g(W^T x + b), \quad (5)$$

где x – вектор входных нейронов; W – матрица весов между слоями; b – вектор смещений данного слоя; g – функция активации нейронов.

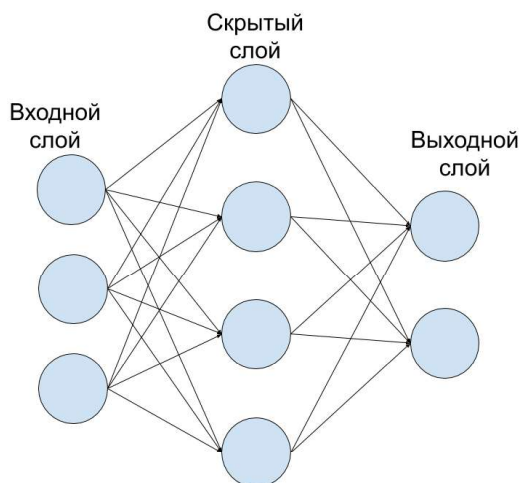


Рисунок 1. Пример MLP

Для активации нейронов наиболее часто применяются следующие функции: выпрямляющий линейный узел (ReLU), гиперболический тангенс (tanh), сигмоида (sigmoid) [2–4].

В случае нескольких скрытых слоев в MLP для каждого последующего скрытого слоя входной вектор в выражении (5) будет представлять собой вектор нейронов предыдущего скрытого слоя. Нейроны выходного слоя определяются выражением, аналогичным (5), за исключением иной функции активации, определяемой типом решаемой задачи. В задачах линейной регрессии, как правило, активация выходного слоя нейронов не используется [2–4].

Обучение MLP заключается в подборе параметров W , b для каждого слоя нейронов с целью минимизации некоторой функции ошибки. В задачах регрессии наиболее часто используемой функцией ошибки является среднеквадратическая ошибка [2–4]:

$$MSE = \frac{1}{m} \sum_{i=1}^m (\hat{y}_i - y_i)^2, \quad (6)$$

где \hat{y}_i – спрогнозированное значение, y_i – истинное значение. Обучение MLP, как правило, производится на специально выделенных тренировочных данных методом стохастического градиентного спуска SGD (Stochastic Gradient Descent).

В SGD для группы входных векторов методом обратного прохода ошибки определяется средний градиент ошибки в каждом слое, в соответствии с которым рассчитываются изменения параметров W и b для минимизации ошибки. Данные изменения производятся достаточное количество раз в течение нескольких временных интервалов (эпох), достаточных для поиска минимума функции ошибки. Однако окончательная

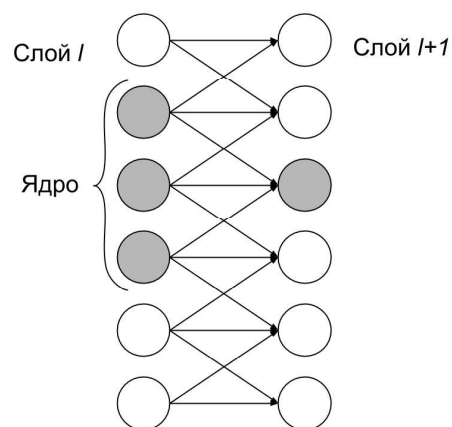


Рисунок 2. Сверточный слой CNN

оценка качества обучения MLP производится на специально выделенных валидационных данных, которые не использовались для обучения MLP [2–4].

Существенным недостатком MLP является очень большое количество параметров в связи с тем, что каждый нейрон последующего слоя связан со всеми нейронами предыдущего слоя. При большом количестве скрытых слоев это приводит к необходимости выделения значительного количества памяти под хранение параметров и требует большой вычислительной мощности при обучении. Для устранения этого недостатка на практике, особенно в системах компьютерного зрения, в качестве замены MLP используют CNN. Нейроны сверточных слоев связаны только с небольшой частью нейронов предыдущего слоя, называемых ядром (kernel), (см. рисунок 2), что многократно снижает число параметров нейронной сети [2–4].

Таким образом, операция матричного умножения в выражении (5) заменяется на операцию свертки матриц. Для дополнительного уменьшения параметров CNN используется операция пулинга, при которой из нескольких соседних нейронов сверточного слоя отбирают для взаимодействия с последующими слоями только один нейрон, как правило, с максимальной активацией. Однако использование CNN связано с необходимостью подбора большого числа параметров: размеров ядра для каждого сверточного слоя и параметров пулинга [2; 3].

Системы MLP и CNN не позволяют учитывать длительные временные связи в хронологически последовательных данных, например таких, как текст и речь. Для обработки таких данных были разработаны рекуррентные нейронные сети, параметры которых адаптируются под длительные изменения входных состояний. Текущее состояние на выходе RNN зависит не только от вход-

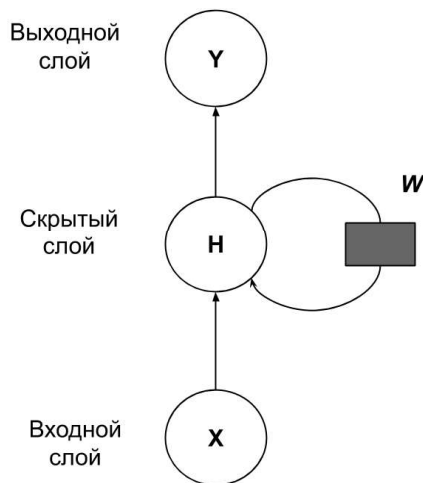


Рисунок 3. Вид простой RNN

ных нейронов, но и учитывает все предыдущие входные состояния, см. рисунок 3 [2]. Скрытый слой нейронов RNN в момент t рассчитывается по формуле [2]:

$$h^{(t)} = g(W h^{(t-1)} + U x^{(t)} + b), \quad (7)$$

где W и U – матрицы весов для скрытого слоя и входного слоя, соответственно.

Недостатком простых RNN является нестабильность градиента при обучении на длительных временных последовательностях. С целью повышения стабильности обучения в RNN используют элементы долгой краткосрочной памяти LSTM (Long Short Time Memory) для определения состояния скрытых слоев. Элементы LSTM позволяют запоминать наиболее характерные хронологические взаимосвязи в последовательностях, однако их использование значительно повышает требования к вычислительным ресурсам в процессе обучения [2; 4].

Исходные данные и постановка задачи

Данные для исследования были взяты с ресурса Kaggle [9]. Набор данных представляет собой сведения об общей загрузке 57 сот 4G сети в течение года, которые собирались с интервалом в час. Например: к соте с номером 039872 подключено 50 пользователей, каждый пользователь имеет среднюю скорость подключения 10 Мбит/с, таким образом, средняя загрузка соты в течение часа составит 500 Мбит/с. Общий размер данных составляет примерно 8738×57 (для некоторых сот небольшая часть данных отсутствует). Величина трафика сильно варьируется в течение суток: высокая загрузка в рабочие часы, от 10 до 19, и низкая – от 0 до 6. Также есть зависимость в течение недели для сот, обслуживающих офисные здания: высокая загрузка с понедельника по пятницу и низкая в субботу-воскресенье. Кроме

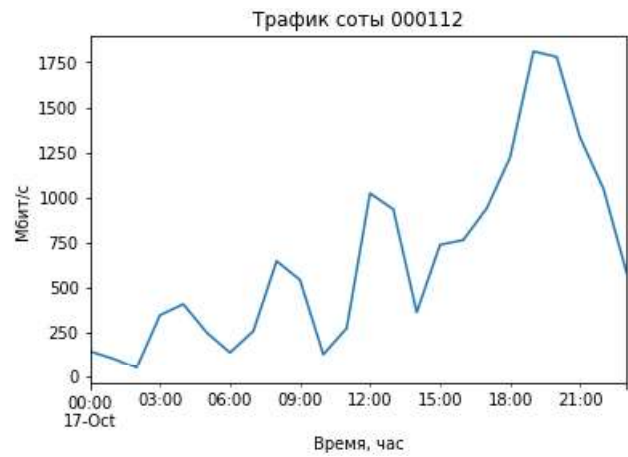


Рисунок 4. Суточный трафик соты

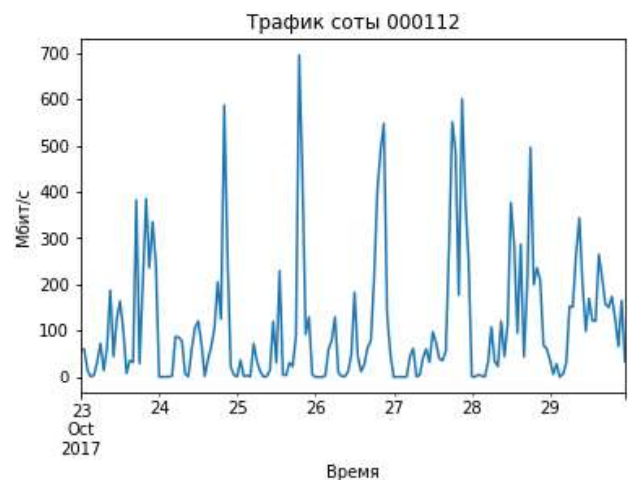


Рисунок 5. Недельный трафик соты

того, могут быть заметные колебания в течение года, связанные со специальными событиями, например, фестивалями, праздниками и т.п.

Пример трафика соты 000112 в течение суток 25 октября 2017 г. представлен на рисунке 4. Как видно на рисунке 4, в течение суток имеется несколько пиков загрузки, которые постепенно нарастают до 18.00–21.00. Аналогичная картина просматривается и в течение недели с 23 по 29 октября 2017 г., см. рисунок 5.

Схожие паттерны трафика наблюдаются и для других сот данной LTE-сети. Несмотря на это, общие временные тренды трафика для большинства сот носят выраженный нестационарный характер. В качестве иллюстрации на рисунке 6 представлена автокорреляционная функция для соты 000821.

Постановка задачи имеет в виду прогнозирование трафика сот LTE-сети на основе данных за предыдущие периоды с помощью базовых методов машинного обучения и сравнения полученных данных с результатами традиционных статистических методов.

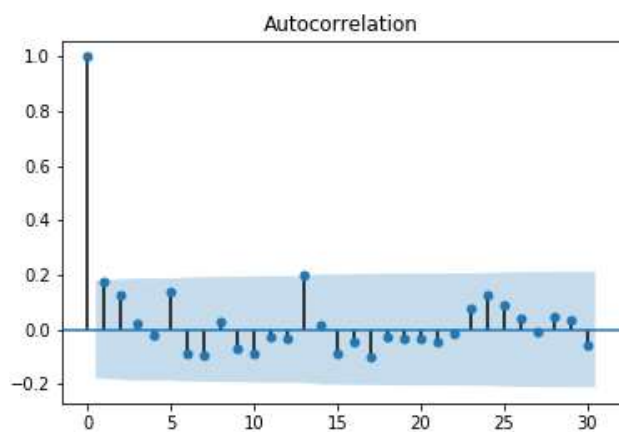


Рисунок 6. Пример автокорреляции

Характеристики и результаты моделирования

Моделирование методов машинного обучения проводилось при помощи библиотеки Tensor Flow, разработанной компанией Google [10]. Для оценки методов ARIMA использовался пакет Statsmodels [11].

Исходные данные за годовой период были разделены на две части:

- обучающая часть (80 %);
- часть для валидации (19 %).

Отсутствующая часть данных (1 %) была замещена линейной интерполяцией на основе ближайших значений. Прогнозирование проводилось на основе предварительно центрированных и нормированных данных относительно среднего и СКО σ за предшествующий трехдневный период следующими методами:

- SMA при усреднении данных за предыдущий период;
- ARIMA, где параметры подбирались для каждой соты индивидуально по критерию наименьшего MSE;
- обучением трехслойной MLP с 16, 8 и 1 нейронами в слоях, активация скрытых слоев – ReLU, выходной слой без активации;
- обучением CNN: первый слой сверточный с 32 нейронами и ядром 4, активация – ReLU; слой пулинга по максимуму из двух нейронов; второй слой – сверточный с 16 нейронами и ядром 3, активация – ReLU, слой пулинга по максимуму из двух нейронов, выходной слой из одного нейрона без активации;
- обучением простой RNN: 32 нейронами в скрытом слое, активация – tanh, один нейрон в выходном слое без активации;
- обучением RNN с LSTM: 32 нейронами – скрытый слой, активация – tanh, один выходной нейрон без активации.

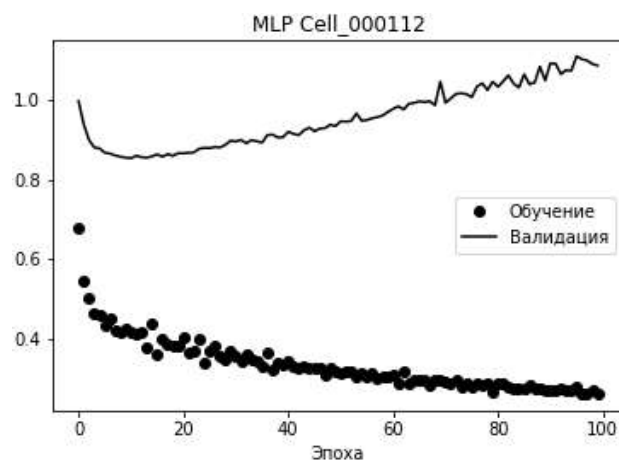


Рисунок 7. Обучение MLP

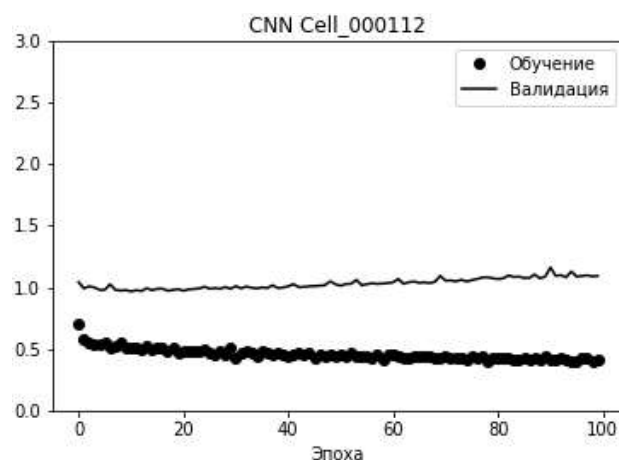


Рисунок 8. Обучение CNN

Ошибка прогнозирования и оценивалась с помощью MSE. В качестве примера на рисунке 7 показано изменение ошибки при обучении MLP соты 000112. Данный пример демонстрирует начальное снижение ошибки валидации до 10 эпохи, а затем ее рост из-за переобучения, при этом ошибка обучения продолжает снижаться. Такой эффект характерен для множества методов машинного обучения. Поэтому с целью минимизации ошибки используется ограничение обучения по эпохам или различные способы регуляризации нейронных сетей [2–4].

На рисунке 8 показан пример обучения CNN для данной соты. Для CNN эффект переобучения менее выражен, однако также имеет место после примерно 10 эпох. На рисунке 9 показан пример обучения простой RNN. Для RNN ошибка валидации перестает снижаться после 40 эпох.

Для RNN с LSTM характеристика обучения имеет сходную картину, что и для простой RNN. Полученные минимальные средние ошибки прогнозирования для некоторых сот и LTE-сети в целом представлены в таблице.

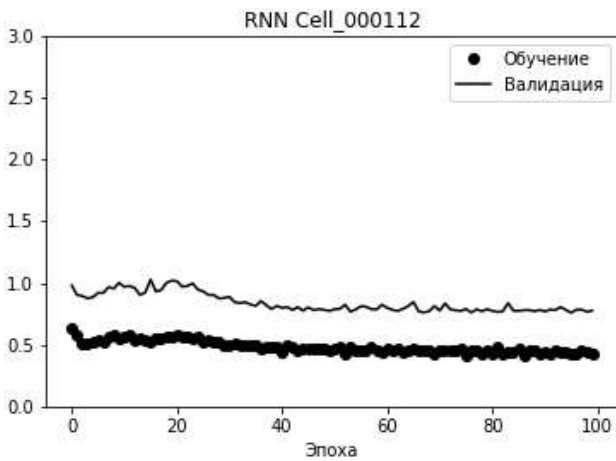


Рисунок 9. Обучение простой RNN

Для прогнозирования трафика полной LTE-сети использовались ANN со следующими параметрами.

1. MLP с четырьмя слоями с 128, 64, 32 и 57 нейронами, активацией ReLU в скрытых слоях.

2. Четырехслойная CNN с 128, 64, 32 нейронами и ядрами в четыре нейрона в сверточных слоях, после каждого сверточного слоя использовалась слои пулинга по максимуму из двух нейронов, выходной слой 57 нейронов без активации.

3. Простой RNN с 128 нейронами и активацией tanh в скрытом слое, 57 нейронами в выходном слое.

4. RNN с LSTM с 128 нейронами и активацией tanh в скрытом слое, 57 нейронами в выходном слое.

Заключение

По результатам прогнозирования трафика LTE-методами машинного обучения и сравнения их с авторегрессионными методами можно сделать следующие выводы:

– методы машинного обучения на основе ANN обеспечивают ошибку прогнозирования на уровне авторегрессионных методов (в среднем порядка 1σ для данного примера) для слабостационарных случаев;

– методы машинного обучения возможно использовать при явно выраженной нестационарности трафика во времени;

– при использовании машинного обучения на основе ANN отсутствует необходимость жесткого определения порядка модели, в отличие от авторегрессионных методов;

– ANN на основе MLP, CNN, простой RNN и RNN с LSTM имеют незначительное расхождение в ошибке прогнозирования;

– машинное обучение с использованием ANN позволяет производить прогнозирование трафика

Таблица. Ошибки прогнозирования

Метод	MSE прогнозирования соты, σ			
	000112	000461	039873	Все
SMA	1,73	1,92	0,81	2,1
ARIMA	0,84	0,85	0,55	–
MLP	0,85	1,06	0,48	1,29
CNN	0,97	1,17	0,48	1,4
RNN	0,76	0,97	0,46	1,14
LSTM	0,75	0,94	0,46	1,12

в сети в целом, что позволяет учитывать скрытые зависимости в распределении трафика между сотами сети.

Кроме того, следует отметить, что в проведенном исследовании рассматривались ANN простой структуры. Для повышения точности прогнозирования возможно использование более сложных ANN, которые, однако, требуют большого объема данных для обучения [2–4].

Литература

1. Time Series Analysis: Forecasting and Control. 5th ed. / G.E.P. Box [et al.]. Hoboken: Wiley, 2015. 712 p.
2. Goodfellow I., Bengio Y., Courville A. Deep Learning. Cambridge: MIT Press, 2016. 800 p.
3. Nielsen M. Neural Networks and Deep Learning. URL: <http://neuralnetworksanddeeplearning.com> (дата обращения: 25.10.2019).
4. Chollet F. Deep Learning with Python. N.-Y.: Manning Publications, 2017. 384 p.
5. A comprehensive survey on machine learning for networking: evolution, applications and research opportunities / R. Boutaba [et al.] // Journal of Internet Services and Applications. 2018. Vol. 9. P. 6-1–99. DOI: 10.1186/s13174-018-0087-2.
6. Chabaa S., Zeroua A., Antari J. Identification and prediction of internet traffic using artificial neural networks // Journal of Intelligent Learning Systems and Applications. 2010. Vol. 2. № 3. P. 147–155. DOI: 10.4236/jilsa.2010.23018.
7. Овчинников К.А., Бушманов В.С. Прогнозирование сетевого трафика при помощи авторегрессионных моделей // Первая МНПК «Проблемы инфокоммуникаций. Наука и технологии». Украина, Харьков, 2013. С. 177–179.
8. Гребенников А.В., Крюков Ю.А., Чернягин Д.В. Моделирование сетевого трафика и прогнозирование с помощью модели ARIMA // Системный анализ в науке и образовании.

2011. № 1. С. 7–17. URL: <https://www.sanse.ru/download/79> (дата обращения: 25.10.2019).
9. Naebolo K.L. Predict traffic of LTE network. URL: <https://www.kaggle.com/naebolo/predict-traffic-of-lte-network> (дата обращения: 25.10.2019).
10. TensorFlow. URL: <https://www.tensorflow.org> (дата обращения: 25.10.2019).
11. Statsmodels. URL: <https://www.statsmodels.org> (дата обращения: 25.10.2019).

Получено 01.11.2019

Трошин Александр Викторович, к.т.н., доцент кафедры сетей и систем связи Поволжского государственного университета телекоммуникаций и информатики. 443010, Российская Федерация, г. Самара, ул. Л. Толстого, 23. Тел. +7 846 339-11-26. E-mail: a.v.troshin77@yandex.ru

MACHINE LEARNING FOR LTE NETWORK TRAFFIC PREDICTION

Troshin A.V.

*Povolzhskiy State University of Telecommunications and Informatics, Samara, Russian Federation
E-mail: a.v.troshin77@yandex.ru*

Predicting network traffic is key to resource management and next-generation network planning. Conventional traffic prediction was based on statistical autoregressive models, which found correlations between current and projected traffic levels. Such models have severe limitations, however; specifically, most of them only work on stationary statistics, which limits them to basic networks and slowly changing levels of incoming traffic. Most of the state-of-the-art networks, however, have a sophisticated structure and rapidly fluctuating traffic flows. Energetic efforts have recently been made to bring machine learning to bear on the limitations of conventional models and to improve predictions. The paper examines key machine learning solutions based on artificial neural networks versus conventional statistical approaches, as well as their practical applications for LTE network traffic prediction.

Keywords: *machine learning, artificial neural networks, time series, traffic prediction, statistical methods*

DOI: 10.18469/ikt.2019.17.4.06

Troshin Alexander Victotovich, Povolzhskiy State University of Telecommunications and Informatics, 23, L. Tolstoy Street, Samara, 443010, Russian Federation; Assistant Professor of Department of Networks and Systems of Telecommunication, PhD in Technical Science. Tel. +7 846 339-11-26. E-mail: a.v.troshin77@yandex.ru

References

1. Box G.E.P. et al. *Time Series Analysis: Forecasting and Control*. 5th ed. Hoboken: Wiley, 2015, 712 p.
2. Goodfellow I., Bengio Y., Courville A. *Deep Learning*. Cambridge: MIT Press, 2016, 800 p.
3. Nielsen M. *Neural Networks and Deep Learning*. Available at: <http://neuralnetworksanddeeplearning.com> (accessed: 21.10.2019).
4. Chollet F. *Deep Learning with Python*. N.-Y.: Manning Publications, 2017, 384 p.
5. Boutaba R. et al. A comprehensive survey on machine learning for networking: evolution, applications and research opportunities. *Journal of Internet Services and Applications*, 2018, vol. 9, pp. 6–1–99. DOI: 10.1186/s13174-018-0087-2.
6. Chabaa S., Zeroua A., Antari J. Identification and prediction of internet traffic using artificial neural networks. *Journal of Intelligent Learning Systems and Applications*, 2010, vol. 2, no. 3, pp. 147–155. DOI: 10.4236/jilsa.2010.23018.
7. Ovchinnikov K., Bushmanov V. Prognozirovanie setevogo trafika pri pomoshchi avtoregressionnyh modelej [Prediction of network traffic using autoregressive models]. *Pervaya MNPК «Prob-*

- lemь infokommunikacij. Nauka i tekhnologii»* [First MNPК «Problems of infocommunications. Science and technology»], 2013, pp. 177-179.
8. Grebennikov A., Krukov Y., Chernyagin D. *Modeling Network Traffic and Forecasting with Model ARIMA*. Available at: <https://www.sanse.ru/download/79> [accessed: 21.10.2019].
 9. Naebolo K. *Predict Traffic of LTE Network*. Available at: <https://www.kaggle.com/naebolo/predict-traffic-of-lte-network> [accessed: 25.10.2019].
 10. *TensorFlow*. Available at: <https://www.tensorflow.org> [accessed: 25.10.2019].
 11. *StatsModels: Statistics in Python*. Available at: <https://www.statsmodels.org> [accessed: 25.10.2019].

Received 01.11.2019

УДК 791.4

МЕТОД ОЦЕНКИ ФУНКЦИОНАЛЬНОСТИ СРЕДСТВ, ИСПОЛЬЗУЕМЫХ В СИСТЕМАХ ВИРТУАЛЬНОЙ РЕАЛЬНОСТИ

Нестерова Е.И.

*Санкт-Петербургский государственный институт кино и телевидения, Санкт-Петербург, РФ
E-mail: nesterova.el.i@yandex.ru*

Технологии и системы виртуальной реальности в настоящее время вызывают большой интерес пользователей в различных областях. На рынке мультимедийной аппаратуры представлен широкий спектр разнообразных моделей устройств и аксессуаров, используемых в системах виртуальной реальности. Предложен метод оценки уровня функциональности средств, используемых в системах виртуальной реальности, основанный на результатах субъективного экспертного опроса пользователей. Особенностью метода является возможность формирования оценочной системы (перечня исходных требований, учета пользовательских предпочтений, субъективных оценочных шкал). Возможность изменения элементов оценочной системы в соответствии с требованиями пользователя позволяет использовать данный метод оценки вне зависимости от существующего на данный момент уровня технологий и конструктивно-параметрических особенностей средств, используемых в системах виртуальной реальности.

Ключевые слова: *функциональность, аппаратные средства, функциональный параметр, системы виртуальной реальности*

Введение

В условиях существующего в настоящее время интереса пользователей к технологиям виртуальной реальности (Virtual Reality, VR), и к поддерживающим данные технологии системам виртуальной реальности, наличия на рынке широкого спектра моделей аппаратных средств, используемых в данной области, необходимы инструменты, предоставляющие возможность пользователям обоснованного выбора определенной номенклатуры элементов при формировании систем виртуальной реальности.

Обоснованный выбор элементной структуры системы виртуальной реальности в общем случае должен быть основан на результатах подробного функционально-параметрического и модельно-параметрического анализа аппаратных средств, используемых в данных системах [1].

В настоящее время на рынке аппаратных средств, используемых в системах виртуальной

реальности, представлен широкий спектр моделей устройств, воспроизводящих изображение, аудиосистем, элементов трекинга, элементов, обеспечивающих взаимодействие пользователя с виртуальной средой, что затрудняет выбор конкретной модели. Поэтому при решении практических задач, связанных с обоснованным выбором конкретной модели определенного мультимедийного средства, выполняющего определенный перечень функций, может быть использован метод оценки, основанный на результатах сравнительного анализа уровня функциональности, предложенный в данной статье.

Метод субъективной экспертной оценки уровня функциональности аппаратных средств, используемых в системах виртуальной реальности

Метод оценки функциональности аппаратных средств, поддерживающих технологии виртуальной реальности, включает несколько этапов [2]: