

## Сравнительный анализ нейросетевой и регрессионных моделей прогнозирования временных рядов

**С. В. Шолтанюк**, ассистент кафедры компьютерных технологий и систем ФПМИ

E-mail: SSholtanyuk@bsu.by

ORCID ID: 0000-0003-0266-7135

Белорусский государственный университет, пр. Независимости, д. 4, 220030, г. Минск, Республика Беларусь

**Аннотация.** Рассмотрена и исследована возможность применения нейронных сетей при решении задачи прогнозирования временных рядов. Для этого было осуществлено обучение нейронной сети для различных рядов с предварительным подбором оптимального набора гиперпараметров. Проведён сравнительный анализ полученной нейросетевой прогностической модели с линейной регрессией и авторегрессией, построенными методом наименьших квадратов. Выявлены условия, влияющие на точность и устойчивость результатов нейронной сети.

**Ключевые слова:** нейронная сеть, обучение нейронной сети, гиперпараметры, точность и устойчивость прогнозирования, MAE, линейная регрессия, авторегрессия, метод наименьших квадратов

**Для цитирования:** Шолтанюк, С. В. Сравнительный анализ нейросетевой и регрессионных моделей прогнозирования временных рядов / С. В. Шолтанюк // Цифровая трансформация. – 2019. – № 2 (7). – С. 60–68. <https://doi.org/10.38086/2522-9613-2019-2-60-68>



© Цифровая трансформация, 2019

## Comparative Analysis of Neural Networking and Regression Models for Time Series Forecasting

**S. V. Sholtanyuk**, assistant of the Department of Computer Applications and Systems, FAMCS

E-mail: SSholtanyuk@bsu.by

ORCID ID: 0000-0003-0266-7135

Belarusian State University, 4 Independence Ave., 220030 Minsk, Republic of Belarus

**Abstract.** Applicability of neural nets in time series forecasting has been considered and researched. For this, training of neural network on various time series with preliminary selection of optimal hyperparameters has been performed. Comparative analysis of received neural networking forecasting model with linear regression has been performed. Conditions, affecting on accuracy and stability of results of the neural network, have been revealed.

**Key words:** neural network, training of neural network, hyperparameters, forecasting accuracy and stability, MAE, linear regression, autoregression, ordinary least squares

**For citation:** Sholtanyuk S. V. Comparative Analysis of Neural Networking and Regression Models for Time Series Forecasting. *Cifrovaja transformacija* [Digital transformation], 2019, 2 (7), pp. 60–68 (in Russian). <https://doi.org/10.38086/2522-9613-2019-2-60-68>

© Digital Transformation, 2019

**Введение.** В настоящее время разработаны и исследованы различные классы моделей и методов прогнозирования временных рядов. Прогностические модели можно разделить на две категории. В статистических моделях зависимость значения временного ряда в тот или иной момент времени от его же значений в предыдущие моменты выражается формулой. В структурных мо-

делях зависимость значения временного ряда от предыдущих задаётся в виде некоторой структуры [1]. Такие модели используются, как правило, в тех случаях, когда вывести эту зависимость в аналитическом виде представляется затруднительным.

К структурным прогностическим моделям относятся, в частности, нейросетевые модели про-

гнозирования. В таких моделях, в отличие от статистических, на точность и устойчивость результатов прогнозирования влияют не только исходные данные, являющиеся объектом прогнозирования, но и некоторые структурные характеристики нейронных сетей (количество нейронов и связи между ними, функции активации, длительность обучения и др.). Это усложняет использование нейронных сетей и является их недостатком. Поэтому интерес представляет сравнение статистического и структурного подходов к прогнозированию различных временных рядов.

**Особенности нейросетевой модели анализа и прогнозирования временных рядов.** Цель данной работы заключается в сравнении структурного (прогностическая нейронная сеть) и статистического (регрессионные модели) подходов к прогнозированию временных рядов, а также в последующем выборе оптимальной модели прогнозирования для того или иного временного ряда. В связи с этой целью были поставлены следующие задачи:

- подбор оптимальных значений гиперпараметров нейросети;
- сравнение результатов по точности нейронной сети и регрессионных моделей для временных рядов с линейными и нелинейными трендами.

Объектом исследования является нейронная сеть, состоящая из трёх последовательных слоёв. Выбор такой архитектуры связан, с одной стороны, с тем, что нейронные сети с одним скрытым слоем являются универсальными аппроксиматорами функций [2], а с другой – использование глубоких нейронных сетей не приводит к улучшению результатов по сравнению с нейросетями с 1-2 скрытыми слоями [3]. Входной слой содержит  $p$  нейронов, на которые подаются  $p$  последовательно идущих элементов рассматриваемого ряда. После входного следует скрытый слой, который может содержать произвольное количество нейронов. На этом слое действует функция активации ReLU, которая задаётся формулой

$$f(x) = \max(0, x).$$

Выходной слой содержит только один нейрон, на котором вычисляется результат предсказания значения временного ряда по  $p$  предыдущим значениям (лагам). Эта нейронная сеть является полносвязной, то есть все нейроны каждого слоя (кроме выходного) соединены со всеми нейронами следующего слоя направленными связями

(синапсами), каждая из которых обладает своим весом. Кроме того, входные данные предварительно нормируются по формуле

$$x' = \frac{x - x_{min}}{x_{max} - x_{min}}, \quad (1)$$

где  $x$  – значение до нормировки,  $x' \in [0, 1]$  – после нормировки,  $x_{min}$  и  $x_{max}$  – соответственно минимальное и максимальное значения временного ряда. К значению, полученному на выходе нейронной сети, применяется преобразование, обратное преобразованию (1), в результате чего получается прогнозное значение временного ряда.

**Схема оценивания гиперпараметров и анализ работы нейронной сети.** Алгоритм, реализующий нейронную сеть, описанную в предыдущем пункте, реализован на языке R с использованием библиотеки Keras [4; 5]. Построение нейросетевой модели для каждого ряда осуществляется путём обучения нейронной сети.

Обучение нейронной сети происходит следующим образом. Вначале формируется  $n - p$  обучающих примеров, где  $n$  – это длина рассматриваемого временного ряда, каждый из которых представляет собой вектор из  $p$  последовательных элементов временного ряда, которому поставлен в соответствие следующий,  $p + 1$ -ый элемент этого же ряда. После чего нейронная сеть обучается на этих примерах определённое количество эпох. Таким образом, нейронная сеть обладает следующими гиперпараметрами, которые следует оценить:

- количество нейронов на входном слое  $p$  (которое также указывает максимальный порядок лага, по которому будет осуществляться прогнозирование ряда);
- количество нейронов на скрытом слое;
- количество эпох обучения.

Осуществляется подбор алгоритма оптимизации нейросети, т. е. алгоритм перерасчёта весов синапсов на каждой эпохе, на примере смоделированного ряда с детерминированным квадратичным трендом:

$$y_t = a + bt + ct^2 + \xi_t, \quad t = \overline{1, n}, \quad (2)$$

где  $y_t$  – значение временного ряда в момент времени  $t$ ,  $a$ ,  $b$  и  $c$  – параметры, задающие тренд,  $\xi_t$ ,  $t = \overline{1, n}$  – случайные величины с нормальным распределением с нулевым математическим ожиданием и заданной дисперсией  $\sigma^2$ .

После выбора алгоритма оптимизации производилось оценивание гиперпараметров

Таблица 1. Сравнение результатов работы нейронной сети на примере ряда  $y_t = a + bt + ct^2 + \xi_t$ ,  $t = \overline{1, n}$ ,  $\xi_t \in N(0, \sigma^2)$ . Приведены результаты при  $a = 21,4$ ,  $b = 3,2$ ,  $c = 0,07$ ,  $n = 1109$ ,  $\sigma = 15$ ,  $Q = 25$   
 Table 1. Comparison of the neural network performance on an example of series  $y_t = a + bt + ct^2 + \xi_t$ ,  $t = \overline{1, n}$ ,  $\xi_t \in N(0, \sigma^2)$ . Results for  $a = 21,4$ ,  $b = 3,2$ ,  $c = 0,07$ ,  $n = 1109$ ,  $\sigma = 15$ ,  $Q = 25$  are given

Название функции-оптимизатора	Среднее значение MAE	Среднее квадратическое отклонение MAE
adadelata	191,79	265,55
adagrad	19,98	5,63
adam	107,89	154,34
adamax	184,50	349,35
nadam	83,21	85,46
rmsprop	462,29	498,85
sgd	26,20	9,97

на примере ряда (2). Для этого осуществлялось обучение нейронной сети на этом ряде с различными наборами значений гиперпараметров. Для каждого из этих наборов обучение повторялось  $Q$  раз. После каждого обучения вычислялось среднее абсолютное отклонение, или MAE (mean absolute error), по следующей формуле:

$$e_q = \frac{1}{n - p + 1} \sum_{t=p+1}^n |y_t - \hat{y}_{q,t}|, \quad q = \overline{1, Q}, \quad (3)$$

где  $\hat{y}_{q,t}$ ,  $t = \overline{p+1, n}$ ,  $q = \overline{1, Q}$ , – прогнозные значения временного ряда  $y_t$ . Для полученных значений  $e_q$ ,  $q = \overline{1, Q}$ , вычислялось затем их среднее значение и среднее квадратическое отклонение. После сравнения этих характеристик происходил выбор оптимальных значений гиперпараметров.

В таблице 1 представлены результаты работы нейронной сети на примере ряда (2) при использовании различных функций-оптимизаторов (эксперимент для каждого оптимизатора повторялся  $Q$  раз).

Как видно из таблицы 1, для большинства алгоритмов оптимизации, реализованных в библиотеке Keras, среднее квадратическое отклонение MAE превышает их среднее значение, что свидетельствует о неустойчивости нейронной сети в этих случаях. Поэтому в качестве функции-оптимизатора во время обучения использовался метод адаптивного градиента, или AdaGrad, который показал лучшую устойчивость нейронной сети по сравнению с другими оптимизаторами.

Данный метод является модификацией метода стохастического градиентного спуска (SGD), широко применяющегося при обучении нейронных сетей. В отличие от SGD, где длина вектора-градиента, вдоль которого изменяются веса

синапсов нейросети, не зависит от входных данных, в AdaGrad учитывается повторяемость входных данных для того, чтобы принять во внимание редко встречающиеся значения временного ряда, которые могут повлиять на искомую модель [6]. Так, если при стохастическом градиентном спуске параметры  $\theta_i$ ,  $i = \overline{1, \nu}$ , функции оптимизации  $J(\theta)$  пересчитываются по формуле

$$\theta_{\tau+1, i} = \theta_{\tau, i} - \eta \cdot \nabla_{\theta} J(\theta_{\tau, i}),$$

где  $\theta_{\tau+1, i}$  – значение параметра  $\theta_i$  в начале  $\tau$ -ой эпохи,  $\eta$  – гиперпараметр, называемый скоростью обучения, то при использовании адаптивного градиента имеет место преобразование вида

$$\theta_{\tau+1, i} = \theta_{\tau, i} - \frac{\eta}{\sqrt{G_{\tau, ii} + \varepsilon}} \cdot \nabla_{\theta} J(\theta_{\tau, i}),$$

где  $G_{\tau}$  – диагональная матрица порядка  $\nu$ , в которой  $i$ -ый диагональный элемент  $G_{\tau, ii}$  равен сумме квадратов частных производных функции  $J$  по параметру  $\theta_i$ , полученных на первых  $\tau$  эпохах,  $\varepsilon$  – сглаживающая переменная, предотвращающая деление на ноль (обычно используются значения порядка  $10^{-8}$ ) [7].

После оценивания гиперпараметров нейронная сеть применялась для построения прогнозов рядов с детерминированными трендами следующего вида:

$$\begin{aligned} y_t &= a + bt + \xi_t, \\ y_t &= a + bt + csint + \xi_t, \\ y_t &= a + bt + ct^2 + \xi_t, \\ y_t &= asint + bt^2 + \xi_t, \\ y_t &= a + bsint + \xi_t, \end{aligned}$$

где  $a$ ,  $b$ ,  $c$ , – параметры тренда,  $\xi_t \in N(0, \sigma^2)$ . При этом инициализация весов синапсов для данных

рядов проводилась случайным образом. После обучения нейронной сети определялось MAE по формуле (3). Эта процедура повторялась  $Q$  раз для каждого временного ряда.

Для каждого рассматриваемого ряда также строилась линейная регрессионная модель вида

$$y_t = \alpha + \beta t + \xi_t, t = \overline{1, n}, \quad (4)$$

а также авторегрессионная модель по формуле

$$y_t = \beta_1 y_{t-1} + \beta_2 y_{t-2} + \dots + \beta_p y_{t-p} + \xi_t, t = \overline{p+1, n}, \quad (5)$$

методом наименьших квадратов. Для обеих моделей затем вычислялись средние абсолютные отклонения прогнозных значений от заданных по формуле (3) (для модели (4) полагалось  $p = 0$ ). Эти

средние абсолютные отклонения сравнивались со статистическими характеристиками MAE, полученными для нейронной сети.

**Результаты исследования.** Примеры результатов работы нейронной сети при различных значениях гиперпараметров приведены в таблице 2.

При десяти нейронах на скрытом слое нейронная сеть является менее точной и устойчивой, о чем свидетельствуют показатели среднего значения и среднего квадратического отклонения MAE, которые превышают аналогичные результаты при других количествах нейронов на скрытом слое. В остальных случаях связь между количеством нейронов на входном и скрытом слоях явно не прослеживается. Из всех рассмотренных результатов наилучшие получены при

Таблица 2. Точность работы нейронной сети на примере ряда  $y_t = a + bt + ct^2 + \xi_t, t = \overline{1, n}$   
Table 2. Accuracy of the neural network performance on an example of the series  $y_t = a + bt + ct^2 + \xi_t, t = \overline{1, n}$

Количество нейронов на входном слое $p$	Количество нейронов на скрытом слое	Среднее значение MAE	Среднее квадратическое отклонение MAE
5	10	70,33	90,39
5	50	26,79	17,75
5	100	19,26	4,51
5	250	22,32	17,37
5	500	19,11	8,24
5	1000	21,71	6,91
10	10	155,01	493,26
10	50	30,98	23,24
10	100	25,18	13,10
10	250	22,94	11,12
10	500	21,92	10,56
10	1000	23,07	14,31

Таблица 3. Точность работы нейронной сети на примере рядов  $y_t = a + bt + ct^2 + \xi_t, t = \overline{1, n}$   
Table 3. Accuracy of the neural network performance on an example of the series  $y_t = a + bt + ct^2 + \xi_t, t = \overline{1, n}$

Длина ряда	Количество нейронов на скрытом слое	Среднее значение MAE	Среднее квадратическое отклонение MAE
1000	100	21,72	6,95
1000	1000	24,37	16,29
1000	5000	19,82	9,00
1000	20000	24,70	15,14
1000	40000	23,25	15,69
5000	100	92,73	40,47
5000	1000	36,11	12,79
5000	5000	30,26	15,85
5000	20000	47,51	48,85
5000	40000	50,73	39,68

количестве нейронов на скрытом слое, равном 100, и  $p \in \{3,5,7\}$  (были рассмотрены значения  $p \in \{3,5,7,10,15\}$ , количество нейронов на скрытом слое из множества  $\{10,50,100,250,500,1000\}$ ).

Была также рассмотрена зависимость точности работы нейронной сети от количества нейронов скрытого слоя на примере рядов различной длины. Примеры результатов приведены в таблице 3.

Наилучшие результаты по точности и устойчивости были получены при следующих парах значений  $n$  и количества нейронов: (1000, 5000), (1000, 10000), (2500, 5000), (2500, 10000), (5000, 5000), (5000, 10000), (10000, 2500), (10000, 5000) (рассматривались ряды длиной  $n \in \{1000,2500,5000,10000\}$  и количество нейронов из множества  $\{100, 500, 1000, 5000, 10000, 20000, 25000, 40000\}$ ). При  $n = 1000$  получились примерно равные результаты для различного количества нейронов на скрытом слое, для других показателей  $n$  лучшую точность и устойчивость нейронная сеть продемонстрировала при 2500–10000 нейронах.

На компьютере с процессором Intel® Core™ i5-3230M CPU 2.60 GHz среднее время обучения нейронной сети для ряда длиной  $n = 1000$  – 9 секунд за одну эпоху, при  $n = 1000$  – 82 секунды и не зависит от количества нейронов. Обучение проводилось в течение 5-10 эпох, так как при дальнейшем обучении точность результатов нейронной сети изменяется незначительно (рис. 1–2).

На рисунках 1 и 2 видно, что значительное улучшение результатов для различных временных рядов достигается после второй эпохи, например, для ряда с квадратичным трендом (рис. 1) MAE нормированных данных уменьшилось в 10 раз по сравнению с первой эпохой. При дальнейшем обучении среднее абсолютное отклонение либо остаётся практически неизменным (рис. 1), либо медленно уменьшается (например, для ряда  $y_t = a + bt + csint + \xi_t$  (рисунок 2) MAE после 12-ой эпохи составило  $2,3 \cdot 10^{-3}$ , а после 24-ой –  $2,2 \cdot 10^{-3}$ , т. е. при увеличении количества эпох в 2 раза результаты улучшились в  $23/22 \approx 1,045$  раза).

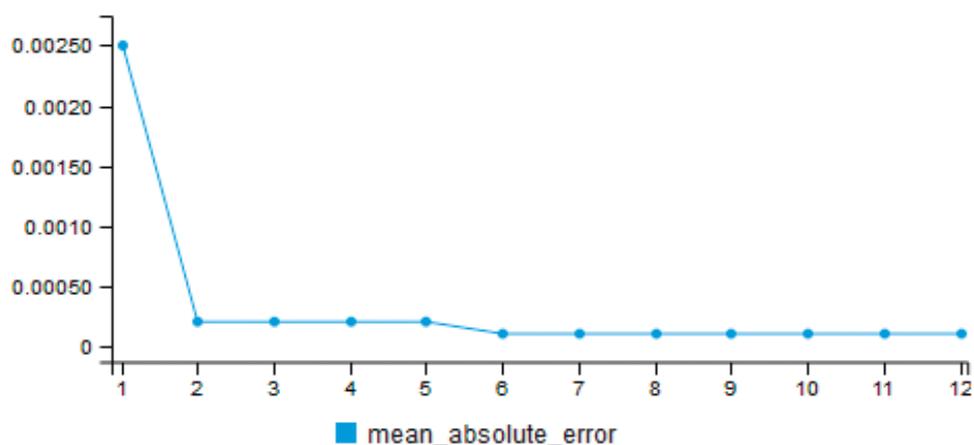


Рис. 1. Динамика значений MAE для нормированных данных после каждой эпохи в процессе обучения нейронной сети при прогнозировании ряда  $y_t = a + bt + ct^2 + \xi_t$

Fig. 1. Dynamics of mean absolute errors for normalized data after each epoch during the neural network training for forecasting of series  $y_t = a + bt + ct^2 + \xi_t$

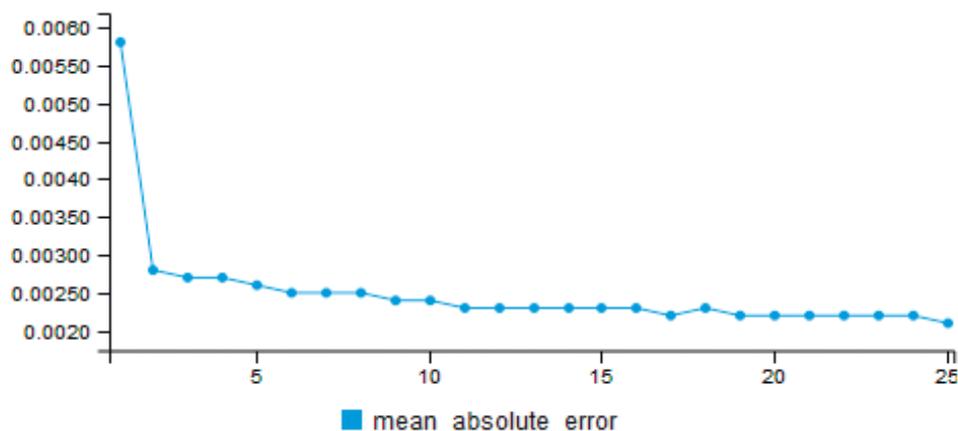


Рис. 2. Динамика значений MAE для нормированных данных при прогнозировании ряда  $y_t = a + bt + csint + \xi_t$

Fig. 2. Dynamics of mean absolute errors for normalized data, when forecasting series  $y_t = a + bt + csint + \xi_t$

Таблица 4. Сравнение точности нейросетевой модели с регрессионными моделями для рядов вида  $y_t = a + bt + \xi_t$ ,  $t = \overline{1, n}$ ,  $\xi_t \in N(0, \sigma^2)$

Table 4. Comparison of accuracy of neural networking model with regression models for series  $y_t = a + bt + \xi_t$ ,  $t = \overline{1, n}$ ,  $\xi_t \in N(0, \sigma^2)$

a	b	σ	Нейронная сеть		Значение MAE (линейная регрессия)	Значение MAE (авторегрессия)
			Среднее значение MAE	Среднее квадратическое отклонение MAE		
5	1	10	10,06	6,00	8,07	10,59
5	5	10	10,84	11,32	8,13	12,41
5	10	10	10,93	13,64	7,94	15,78
5	50	10	13,42	19,43	8,01	55,51
5	100	10	16,39	35,33	7,56	108,95
10	1	10	9,51	5,03	7,89	10,68
10	5	10	11,09	12,38	8,26	12,26
10	10	10	12,31	20,22	8,38	15,98
10	50	10	16,05	31,38	8,16	55,41
10	100	10	19,35	61,89	7,84	109,02
1	10	1	1,88	2,09	0,76	10,90
1	10	5	6,25	1,78	3,98	12,19
1	10	10	12,59	5,60	8,12	15,87
1	10	50	53,59	9,64	42,12	61,86
1	10	100	101,13	12,89	82,10	101,14

В дальнейшем анализировалась зависимость точности работы нейросети от параметров  $a, b, c, \sigma$  при  $p = 7$  и 100 нейронах на скрытом слое. Полученные характеристики средних абсолютных отклонений и их сравнение с MAE у регрессионных моделей для рядов с линейным трендом представлены в таблице 4.

Для всех рядов с линейным трендом нейросетевая модель оказалась более точной, чем авторегрессионная модель (5), но менее точной по сравнению с линейной регрессией (4). Кроме того, среднее абсолютное отклонение при использовании модели (4) не зависит от свободного члена и коэффициента, которыми задаётся линейный тренд рядов, в то время как в нейросетевой модели при увеличении коэффициента  $b$  точность и устойчивость результатов ухудшаются. Дисперсия  $\sigma^2$  также влияет на точность и устойчивость результатов.

В таблицах 5–8 представлены результаты по точности и сравнение моделей для рядов с нелинейным трендом.

При прогнозировании рядов с линейным трендом и синусоидальными колебаниями важную роль играет дисперсия  $\sigma^2$  случайных величин  $\xi_t$ , увеличение которой влечёт рост значений MAE при использовании рассмотренных моделей.

При прогнозировании рядов с квадратичным трендом влияние на работу моделей оказывает также трендовый параметр  $c$ , при увеличении которого наблюдается ухудшение точности и устойчивости нейросетевой модели, а также ухудшение точности регрессионных моделей.

Для рядов вида  $y_t = asint + bt^2 + \xi_t$  при увеличении коэффициента  $b$  увеличиваются MAE у нейронной сети и у авторегрессии, причём при использовании авторегрессии изменение результатов более существенно. Влияние параметров  $a$  и  $\sigma$  менее значительно.

Для рядов с синусоидальными колебаниями, не имеющих тренда, точность прогнозирования зависит исключительно от дисперсии  $\sigma^2$ .

В целом из таблиц 5–8 можно сделать вывод, что для рядов с нелинейным трендом лучше подходит нейросетевая модель. Исключение составили ряды вида  $y_t = 1 + t + csint + \xi_t$  при  $c = 1$  и  $c = 10$  (таблица 5). В этом случае нелинейный компонент  $csint$ , значение которого варьируется от  $-c$  до  $c$ , оказывает меньшее влияние на поведение ряда, нежели случайные величины  $\xi_t$ , значения которых варьируются в пределах  $(-3\sigma, 3\sigma)$ , т. е. от  $-3$  до  $3$ , от  $-30$  до  $30$  и от  $-150$  до  $150$  для рассмотренных примеров. Поэтому здесь результаты нейрон-

Таблица 5. Сравнение точности нейросетевой модели с регрессионными моделями для рядов вида  $y_t = 1 + t + csint + \xi_t$ ,  $t = \overline{1, n}$ ,  $\xi_t \in N(0, \sigma^2)$

Table 5. Comparison of neural networking model with regression models for series  $y_t = 1 + t + csint + \xi_t$ ,  $t = \overline{1, n}$ ,  $\xi_t \in N(0, \sigma^2)$

с	$\sigma$	Нейронная сеть		Значение MAE (линейная регрессия)	Значение MAE (авторегрессия)
		Среднее значение MAE	Среднее квадратическое отклонение MAE		
1	1	1,35	2,26	0,97	1,65
1	10	9,46	4,63	7,94	10,48
1	50	42,48	2,92	38,68	43,28
10	1	3,84	5,38	6,41	5,11
10	10	10,88	7,27	9,74	11,75
10	50	42,23	7,95	38,59	43,21
50	1	1,32	1,19	31,81	4,64
50	10	11,47	8,37	32,99	12,96
50	50	47,17	8,28	50,72	48,03

Таблица 6. Сравнение точности нейросетевой модели с регрессионными моделями для рядов вида  $y_t = 1 + t + ct^2 + \xi_t$ ,  $t = \overline{1, n}$ ,  $\xi_t \in N(0, \sigma^2)$

Table 6. Comparison of neural networking model with regression models for series  $y_t = 1 + t + ct^2 + \xi_t$ ,  $t = \overline{1, n}$ ,  $\xi_t \in N(0, \sigma^2)$

с	$\sigma$	Нейронная сеть		Значение MAE (линейная регрессия)	Значение MAE (авторегрессия)
		Среднее значение MAE	Среднее квадратическое отклонение MAE		
0,01	1	1,80	4,43	641,55	14,98
0,01	10	12,31	14,69	641,38	18,89
0,01	50	47,86	26,46	643,20	57,56
0,1	1	13,00	46,00	6415,04	140,45
0,1	10	16,08	30,37	6414,37	140,97
0,1	50	59,73	76,45	6416,26	152,50
1	1	135,08	490,01	64149,95	1395,65
1	10	127,08	495,31	64150,18	1395,72
1	50	109,20	154,98	64150,21	1396,38

ной сети схожи с результатами по рядам с линейным трендом, и она работает хуже, чем линейная регрессионная модель (4).

Кроме того, результаты работы нейронной сети оказались несколько хуже, чем у авторегрессионной модели (5), для рядов  $y_t = asint + bt^2 + \xi_t$  при  $a = 50$ ,  $b = 0,02$  и  $y_t = asint + bsint + \xi_t$  (таблицы 7 и 8). Связано это с тем, что данным рядам присуща ярко выраженная периодичность, в силу которой имеет место явная зависимость значения временного ряда в тот или иной момент времени от его же значений в прошлом, которую можно оценить авторегрессионной моделью.

При увеличении дисперсии случайных величин  $\xi_t$  точность нейронной сети уменьшается,

а на устойчивость влияние дисперсии не прослеживается. Параметры, задающие тренд временного ряда, оказывают влияние преимущественно на точность модели, а её устойчивость не всегда зависит от них. Например, в рядах  $y_t = 1 + t + ct^2 + \xi_t$  при увеличении параметра с среднее квадратическое отклонение MAE тоже значительно увеличивается (таблица 6), в остальных случаях такой явной зависимости не наблюдается.

**Заключение.** Из результатов исследования можно сделать вывод, что последовательные трёхслойные полносвязные нейронные сети можно использовать в задачах прогнозирования временных рядов. При этом важную роль играет выбор значений гиперпараметров, который влияет на точность

Таблица 7. Сравнение точности нейросетевой модели с регрессионными моделями для рядов вида  $y_t = asint + bt^2 + \xi_t$ ,  $t = 1, n$ ,  $\xi_t \in N(0, \sigma^2)$

Table 7. Comparison of neural networking model with regression models for series  $y_t = asint + bt^2 + \xi_t$ ,  $t = 1, n$ ,  $\xi_t \in N(0, \sigma^2)$

a	b	σ	Нейронная сеть		Значение MAE (линейная регрессия)	Значение MAE (авторегрессия)
			Среднее значение MAE	Среднее квадратическое отклонение MAE		
10	0,02	1	27,18	16,07	14229,98	28,52
10	0,02	5	24,73	13,86	14229,93	29,13
10	0,02	20	35,21	10,54	14229,50	36,79
10	0,1	1	31,44	20,96	14229,96	139,59
10	0,1	5	28,25	16,36	14229,90	139,71
10	0,1	20	35,24	8,68	14230,65	141,59
50	0,02	1	44,54	11,53	14230,10	40,37
50	0,02	5	49,78	12,01	14229,98	40,57
50	0,02	20	51,37	17,44	14229,41	45,16
50	0,1	1	55,62	51,46	14230,07	142,36
50	0,1	5	49,84	22,22	14230,00	142,49
50	0,1	20	54,02	14,69	14230,02	144,33

Таблица 8. Сравнение точности нейросетевой модели с регрессионными моделями для рядов вида  $y_t = a + bsint + \xi_t$ ,  $t = 1, n$ ,  $\xi_t \in N(0, \sigma^2)$

Table 8. Comparison of neural networking model with regression models for series  $y_t = a + bsint + \xi_t$ ,  $t = 1, n$ ,  $\xi_t \in N(0, \sigma^2)$

a	b	σ	Нейронная сеть		Значение MAE (линейная регрессия)	Значение MAE (авторегрессия)
			Среднее значение MAE	Среднее квадратическое отклонение MAE		
-100	10	1	1,14	0,068	19,16	0,92
-100	10	5	5,18	0,097	19,65	4,40
-100	10	20	18,72	0,274	25,86	16,90
-100	50	1	1,11	0,109	19,16	1,04
-100	50	5	5,26	0,117	19,60	4,48
-100	50	20	20,13	0,166	26,19	17,25
100	10	1	1,06	0,046	19,08	0,90
100	10	5	4,90	0,083	19,59	4,45
100	10	20	19,08	0,194	25,54	16,33
100	50	1	1,07	0,076	19,16	1,10
100	50	5	5,17	0,085	19,77	4,79
100	50	20	19,19	0,310	26,13	17,90

и устойчивость нейронной сети. Было также показано влияние выбора метода оптимизации при обучении нейронной сети на её устойчивость.

Был проведён сравнительный анализ нейросетевой модели с линейной регрессией и авторегрессией, в результате которого выяснилось, что линейная регрессия лучше аппроксимирует ряды с линейным трендом, авторегрессия – ряды с выраженной пери-

одичностью, а нейронная сеть – непериодические ряды с трендом, отличным от линейного. Время прогнозирования нейронной сетью зависит только от количества эпох и длины прогнозируемого ряда. Таким образом, при помощи рассмотренной нейросетевой модели можно за ограниченное время построить прогноз для произвольного временного ряда с детерминированным трендом.

## Список литературы

1. Чучуева, И. А. Модель прогнозирования временных рядов по выборке максимального подобия : дис. ... канд. тех. наук : 05.13.18 / И. А. Чучуева ; Московский государственный технический университет им. Н. Э. Баумана. – М., 2012. – 155 л.
2. Hornik, K. Multilayer feedforward networks are universal approximators / K. Hornik, M. Stinchcombe, H. White // *Neural Networks* – 1989. – Vol. 2, iss. 5. – Pp. 359–366.
3. Bengio, Y. Scaling learning algorithms towards AI / Y. Bengio, Y. LeCun // *Large-Scale Kernel Machines* / L. Bottou [et al.] – Cambridge, MA : MIT Press, 2007. – Pp. 323–362.
4. R Core Team (2018). R: A language and environment for statistical computing. R Foundation for Statistical Computing, Vienna, Austria. URL: <https://www.R-project.org/>. – Date of access: 19.01.2019.
5. JJ Allaire and Francois Chollet (2018). keras: R Interface to 'Keras'. R package version 2.2.0. <https://CRAN.R-project.org/package=keras>. – Date of access: 19.01.2019.
6. Duchi, J. Adaptive Subgradient Methods for Online Learning and Stochastic Optimization / J. Duchi, E. Hazan, Y. Singer // *Journal of Machine Learning Research* – 2011. – Vol. 12 – P. 2121–2159.
7. Ruder, S. An overview of gradient descent optimisation algorithms [Electronic resource] / S. Ruder // *arXiv.org e-Print archive* – Mode of access: <https://arxiv.org/abs/1609.04747>. – Date of access: 19.01.2019. – (Preprint / arXiv:1609.04747v2).

## References

1. Chuchueva I. A. Model' prognozirovaniya vremennykh ryadov po vyborke maksimal'nogo pravdopodobiya [The time series forecast model based on maximum likeness set] Candidate of science dissertation. M.: Bauman Moscow State Technical University, 2012. 155 p. (In Russian).
2. Hornik K., Stinchcombe M., White H. Multilayer feedforward networks are universal approximators, *Neural Networks* 2 (5) (1989). pp. 359-366.
3. Bengio Y., LeCun Y. Scaling learning algorithms towards AI. In L. Bottou et al. (Eds.), *Large-Scale Kernel Machines*. Cambridge, MA, MIT Press, 2007. pp. 323-362.
4. R Core Team (2018). R: A language and environment for statistical computing. R Foundation for Statistical Computing, Vienna, Austria. Available at: <https://www.R-project.org/> (accessed: 19.01.2019)
5. JJ Allaire and Francois Chollet (2018). keras: R Interface to 'Keras'. R package version 2.2.0. Available at: <https://CRAN.R-project.org/package=keras>. (accessed: 19.01.2019)
6. J. Duchi. E. Hazan. Y. Singer. Adaptive Subgradient Methods for Online Learning and Stochastic Optimization, *Journal of Machine Learning Research*, 12 (2011). pp. 2121-2159.
7. Ruder S. An overview of gradient descent optimisation algorithms. arXiv preprint arXiv:1609.04747v2. Available at: <https://arxiv.org/abs/1609.04747>. (accessed: 19.01.2019)

*Received: 14.05.2019*

*Поступила: 14.05.2019*