

ПРОГНОЗИРОВАНИЕ ФИНАНСОВЫХ ВРЕМЕННЫХ РЯДОВ С ИСПОЛЬЗОВАНИЕМ РЕКУРРЕНТНЫХ НЕЙРОННЫХ СЕТЕЙ LSTM

© 2022 г. Г. Х. МАФУТАЛА

Аннотация. Прогнозирование доходности фондового рынка является важной задачей в управлении портфелем. Успешное предсказание будущей цены акции может принести значительную прибыль. Однако ненормальность финансового рынка не позволяет простым моделям предсказывать будущие цены активов с высокой точностью. В настоящее время научные исследования в этой области сосредоточены на глубоком обучении, которое заключается в соединении нескольких искусственных нейронов в виде слоев для создания нелинейных моделей, способных обучаться выполнять разнообразные и сложные задачи из большого количества примеров. В этой статье мы построим модель на основе рекуррентных нейронных сетей, а точнее, на основе LSTM-сети для прогнозирования будущих цен фондового рынка.

Ключевые слова: прогнозирование временных рядов, рекуррентная нейронная сеть, долгая краткосрочная память, LSTM.

ВВЕДЕНИЕ

В настоящей работе изучается эволюция случайного явления во времени, которое предполагается дискретным. В этом случае мы говорим о временном ряде. Временные ряды — это актуальный инструмент, применимый во множестве решений, от предсказания цен на акции, прогнозов погоды, планирования бизнеса, до распределения ресурсов [1].

Здесь речь идет о прогнозировании биржевого курса валютной пары на рынке Форекс. Таким образом, задача данного исследования состоит в том, чтобы на базе котировки валютной пары в определенном временном интервале — $x_t = (x_1, \dots, x_n)$, прогнозировать будущие котировки валютной пары — \hat{x}_{n+h} еще не наблюдаемое на горизонте h .

Основной целью этой статьи является тренировка алгоритма глубокого обучения для прогнозирования будущих котировок по валютной паре. Архитектура модели будет основана на рекуррентной нейронной сети LSTM, а вызов будет заключаться в том, чтобы получить высокую точность не только на обучающем наборе данных, но и на новой выборке данных. Модель должна хорошо обобщаться.

1. ВРЕМЕННЫЕ РЯДЫ И ИХ КОМПОНЕНТЫ

Временной ряд — это действительная конечная последовательность наблюдений $(x_t)_{1 \leq t \leq n}$ ($n \in \mathbb{N}^*$), где t представляет собой единицу времени (час, день, неделя и т.д.). Последовательность случайных переменных x_1, x_2, \dots, x_t представляют собой различные значения ряда во времени, а x_t временной ряд (см. рис. 1).

При анализе временных рядов принято выделять следующие компоненты [2]:

- тренд (Z_t) — плавно изменяющаяся компонента, описывающая влияние долговременных факторов,

- сезонная компонента (S_t) — циклические колебания изучаемого процесса,
- случайная составляющая (ε_t) — компонента, показывающая влияние случайных факторов. Она состоит из всего того, что не учитывается трендом и сезонным движением.



Рис. 1. Пример временного ряда. Цены открытия курса доллара США по отношению к российскому рублю (USD/RUB) с января 2021 года по апрель 2022 года.

Выделение этих компонентов (см. рис. 2) — один из первых важных этапов анализа. Таким образом модель временного ряда можно описать как $X_t = Z_t + S_t + \varepsilon_t$ (аддитивная модель) и $X_t = Z_t * S_t * \varepsilon_t$ (мультипликативная модель), $\forall t \in 1, \dots, n$.

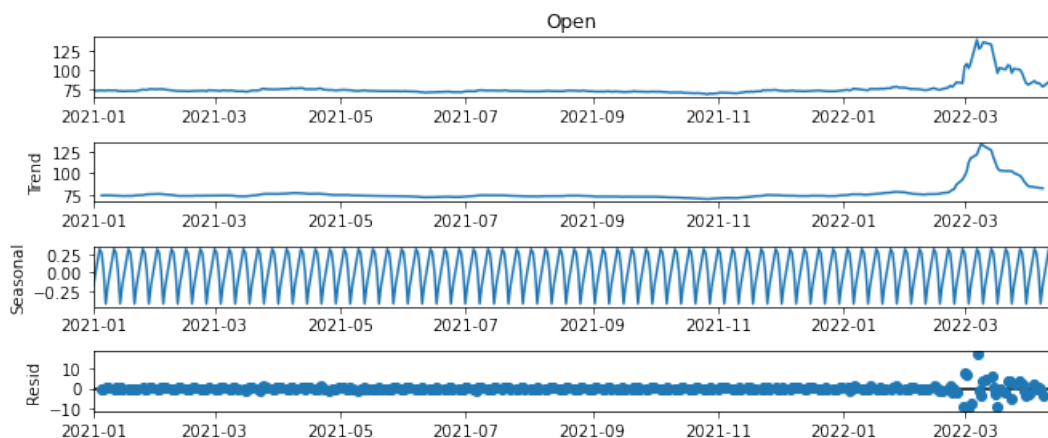


Рис. 2. Разложение временного ряда на его 3 соответствующих компонента (аддитивная модель).

Из всех компонентов временного ряда *тренд* остается самым важным. Это не только самый простой индикатор для принятия решения о покупке или продаже, но и наличие тенденций на различных финансовых рынках (акции, курсы валют, фьючерсы и т.д.) является важным элементом при распределении активов в портфеле [6]. Говорят, что ряд имеет тенденцию, если можно написать: $x_t = f(t) + \varepsilon_t$, при f — фиксированная функция, а ε_t — случайные шумы [4].

Когда $x_t = \alpha_t + \beta + \varepsilon_t$, это означает, что тренд линейный, а полиномиальная тенденция приведет к $x_t = \alpha_t t^p + \alpha_{t-1} t^{p-1} + \dots + \alpha_{p+1} + \varepsilon_t$ и так далее. Цель здесь состоит в том, чтобы сделать предсказания относительно эволюции ряда x_t , другими словами: оценить неизвестные параметры α_i , при $i = 1, \dots, n$.

2. РЕКУРРЕНТНАЯ НЕЙРОННАЯ СЕТЬ

Рекуррентная нейронная сеть (РНС) является моделью нейронной сети, предложенной в 80-х годах для моделирования временных рядов [9, 5, 10]. Структура сети похожа на структуру стандартного многослойного перцептрона, с тем отличием, что мы допускаем связи между скрытыми слоями, связанными с временной задержкой (см. рис. 3). Благодаря этим связям модель может сохранять информацию о прошлом, что позволяет ей обнаруживать временные корреляции между событиями, которые находятся на большом расстоянии друг от друга в данных [7].

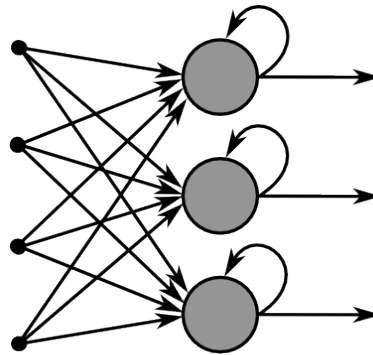


Рис. 3. Схема рекуррентной нейронной сети. Благодаря связям между скрытыми слоями модель может сохранять информацию о прошлом.

2.1. Архитектура и обучение традиционной РНС. Рекуррентные нейронные сети позволяют использовать предыдущие выходы в качестве входов, имея при этом скрытые состояния. Поэтому они идеально подходят для последовательной обработки данных. Обычно они имеют структуру, показанную на рис. 4.

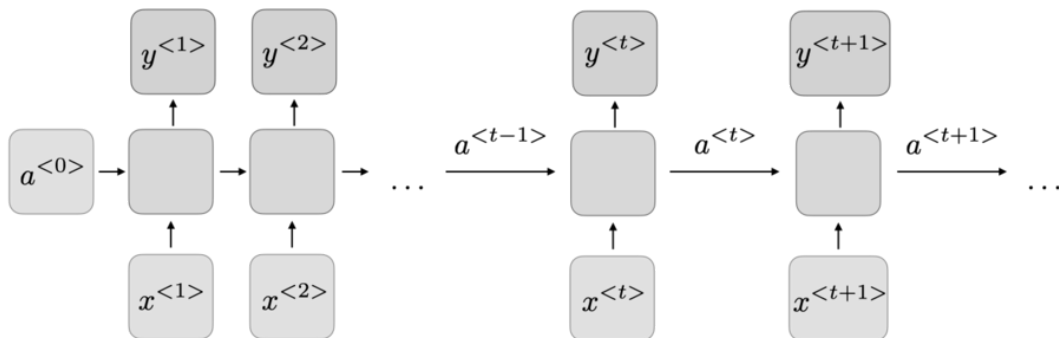


Рис. 4. Традиционная архитектура РНС.

Для каждого временного интервала t вектор активации $a^{<t>}$ и выход $y^{<t>}$ выражаются уравнениями 1 и 2 [8].

$$a^{<t>} = g_1(W_{aa}a^{<t-1>} + W_{ax}x^{<t>} + b_a), \tag{1}$$

$$y^{<t>} = g_2(W_{ya}a^{<t>} + b_y). \tag{2}$$

Где x_t и a_t – соответственно вектор входа сети и вектор активации в момент времени t , g_1 и g_2 – функции активации, W_{aa} – рекуррентная весовая матрица, W_{ax}, W_{ya} – входные весовые матрицы, а b коэффициенты смещения. Таким образом, тренировка модели будет заключаться в обучении весов W , а также коэффициентов смещений b .

Выходное значение $y^{<t>}$ в момент времени t рассчитывается по уравнению 2 в соответствии со значением активации $a^{<t>}$, рассчитанным по уравнению 1. При увеличении масштаба внутри активации $a^{<t>}$, видно, что $W_{ax}, W_{aa}, W_{ya}, b_a, b_y$ – это коэффициенты, которые разделяются во времени, а также функции активации g_1, g_2 (см. рис. 5).

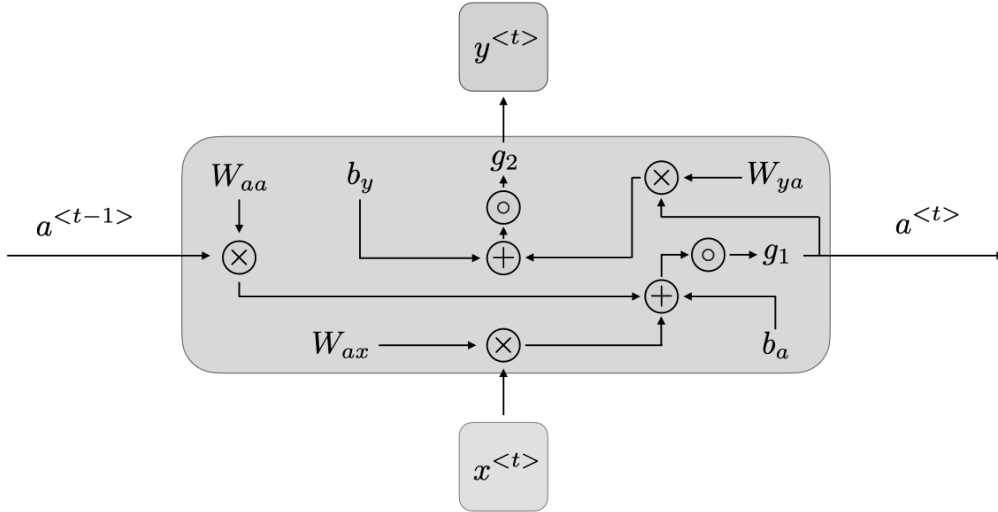


Рис. 5. Коэффициенты весов и смещения разделяются внутри скрытого слоя.

В этих расчетах можно ясно видеть *рекуррентный аспект* (расчет в момент времени t основывается на информации, предоставленной в момент времени $t - 1$, сам по себе рассчитывается на основе информации, предоставленной в момент времени $t - 2$ и т.д.), в отличие от классической ИНС (искусственной нейронной сети), где выход зависит только от входных значений.

Функция потерь \mathcal{L} для всех временных шагов определяется на основе потерь на каждом временном шаге следующим образом:

$$\mathcal{L}(\hat{y}, y) = \sum_{t=1}^{T_y} \mathcal{L}(\hat{y}^{<t>}, y^{<t>}) \quad (3)$$

Обратное распространение выполняется в каждый момент времени. На временном шаге T производная потерь \mathcal{L} по отношению к весовой матрице W выражается следующим образом:

$$\frac{\partial \mathcal{L}^{(T)}}{\partial W} = \sum_{t=1}^T \frac{\partial \mathcal{L}^{(T)}}{\partial W} \Big|_{(t)} \quad (4)$$

Хотя в принципе рекуррентная сеть является простой и мощной моделью, на практике ее трудно правильно обучить. Причина их возникновения заключается в том, что трудно уловить долгосрочные зависимости из-за мультипликативного градиента, который может экспоненциально уменьшаться или увеличиваться по отношению к количеству слоев. Это известно как проблема *исчезающего* или *взрывающегося градиента* [7]. Чтобы преодолеть

эти недостатки, с тех пор были разработаны новые варианты РНС. В нашем случае мы будем использовать LSTM-сеть.

2.2. LSTM-сеть. Долгая краткосрочная память (англ. Long short-term memory; LSTM), обычно называемая *LSTM-сетью* — это особый вид рекуррентных нейронных сетей, способных к запоминанию долговременных зависимостей. Они были введены Сеппом Хохрайтером и Юргеном Шмидхубером в 1997 году и были использованы и развиты многими исследователями в своих работах.

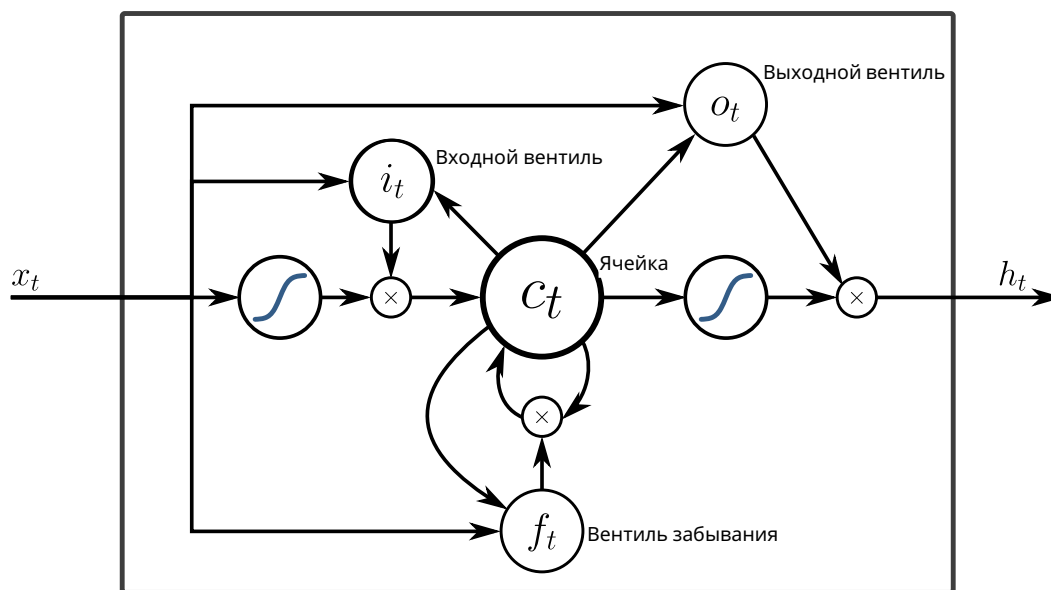


Рис. 6. Схема работы базового блока LSTM.

Блок LSTM, показанный на рис. 6 является основным компонентом архитектуры LSTM. Это ряд вентиляей и ячеек, которые взаимодействуют для получения конечного результата. Скрытое состояние LSTM-сети h_t в момент времени t по состоянию $h_{(t-1)}$ на предыдущем шаге находится по формулам от ур. 5 до ур. 10 [3]:

$$f_t = \sigma(W_f x_t + U_f h_{t-1} + b_f), \tag{5}$$

$$i_t = \sigma(W_i x_t + U_i h_{t-1} + b_i), \tag{6}$$

$$\tilde{c}_t = \tanh(W_c x_t + U_c h_{t-1} + b_{\tilde{c}}), \tag{7}$$

$$c_t = (f_t \otimes c_{t-1}) \oplus (i_t \otimes \tilde{c}_t), \tag{8}$$

$$o_t = \sigma(W_o x_t + U_o h_{t-1} + b_o), \tag{9}$$

$$h_t = o_t \otimes \tanh(c_t). \tag{10}$$

Где σ – сигмовидная функция, f_t – вектор активации вентиля забывания, i_t – вектор активации вентиля входа, o_t – вектор активации вентиля выхода, \tilde{c}_t – вектор активации входа ячейки, c_t – состояние ячейки, а h_t – выходной вектор блока LSTM. Таким образом, при тренировке модели, основанной на LSTM-сети, необходимо научиться всем весам W и U , а также всем векторам смещения b .

Затем для вычисления функции потерь, в нашем случае мы будем использовать средне-квадратичную ошибку (англ. Mean Squared Error; *MSE*), заданную следующей формулой:

$$\text{MSE} = \frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2 \quad (11)$$

3. МЕТОДЫ И ИСХОДНЫЕ ДАННЫЕ

Данные, используемые в этой статье, представляют собой ежедневные цены открытия валютной пары USD/RUB (доллар США по отношению к российскому рублю), взятые с сайта *Yahoo! Finance*. Набор данных для обучения модели охватывает период с 01.01.2014 по 31.12.2021. Наша модель использует 80% данных для обучения и остальные 20% данных для валидации.

Оба два набора данных структурированы на 60 временных шагов для 1 выхода. Это значит, что для каждого момента времени t нейронная сеть будет использовать цены последних 60 дней: $x_{t-1}, x_{t-2}, \dots, x_{t-60}$ и на основе этих значений предскажет цену на следующий день x_{t+1} .

Для обучения мы используем среднюю квадратичную ошибку (уравнение 11) для оптимизации нашей модели. Также мы использовали различные эпохи для обучения данных (12 эпох, 25 эпох, 50 эпох и 100 эпох). Наша модель будет иметь следующую структуру:

ТАБЛИЦА 1. Структура модели LSTM

Слой (тип)	Выходная форма	Параметры
lstm (LSTM)	(None, 60, 60)	14880
dropout (Dropout)	(None, 60, 60)	0
lstm_1 (LSTM)	(None, 64, 120)	86880
dropout_1 (Dropout)	(None, 60, 120)	0
lstm_2 (LSTM)	(None, 60, 120)	115680
dropout_2 (Dropout)	(None, 60, 120)	0
lstm_3 (LSTM)	(None, 60)	43440
dropout_3 (Dropout)	(None, 60)	0
dense (Dense)	(None, 1)	61
Количество параметров:		260,941
Обучаемые параметры:		260,941
Необучаемые параметры:		0

4. РЕЗУЛЬТАТЫ И ОБСУЖДЕНИЕ

После обучения модели в течение нескольких эпох мы обнаружили, что ошибка предсказания уменьшается по мере продвижения, как показано на рис. 7. Мы также видим, что модель еще не переобучилась на обучающем наборе данных, демонстрируя сравнимые навыки на обоих наборах данных.

Мы также протестировали модель на новой выборке данных, чтобы увидеть, насколько хорошо она обобщается. Эта выборка содержит котировки цен открытия валютной пары USD/RUB за период с 01 января по 13 апреля 2022 года. И мы получили разные результаты в зависимости от количества эпох, в течение которых обучалась модель, как показано на рис. 8.

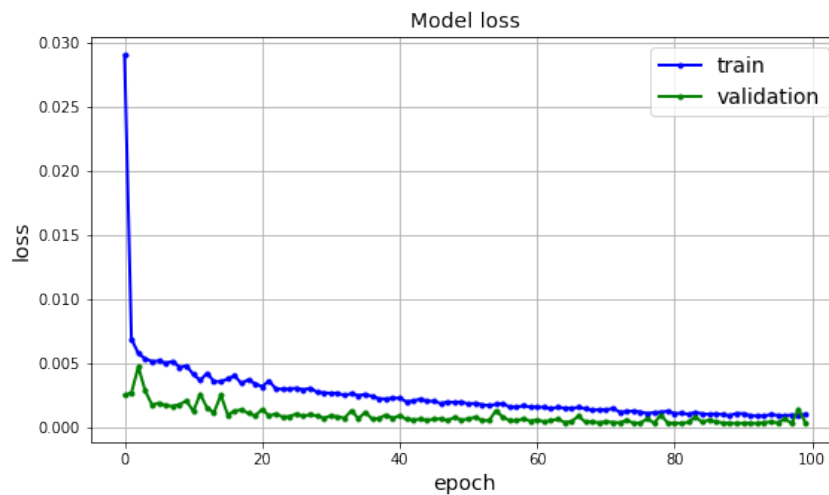


Рис. 7. График, показывающий, как ошибка уменьшается с течением времени на обучающей и проверочной наборах данных.

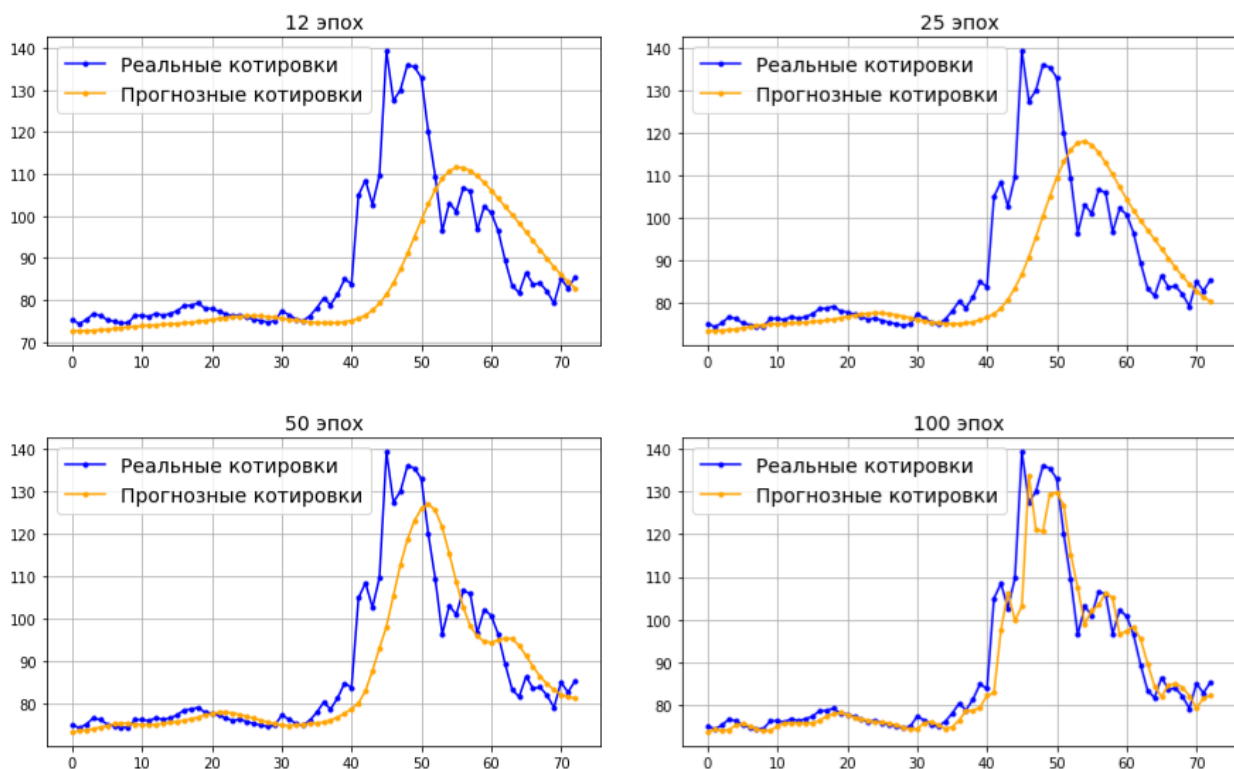


Рис. 8. Результаты обучения модели прогнозирования курса валютной пары USD/RUB с различным количеством эпох.

После обучения модели в течение 100 эпох видно, что она предсказывает с хорошей точностью цены валютной пары, на которой она была обучена, поэтому можно сказать, что она хорошо обобщается на новом наборе данных.

Также заметно, что модель немного запаздывает, когда происходит резкое изменение тренда из-за высокой волатильности. Но мы не хотели обучать ее на более чем 100 эпох,

чтобы избежать риска переобучения. И видно, что наша модель не переобучилась обучающим данным, так как на графике она не следует строго за линией реального курса.

В таблице 2 показано уменьшение ошибки на обучающем и проверочном наборах данных для различного количества эпох.

ТАБЛИЦА 2. Потеря ошибки в прогнозировании курса валютной пары USD/RUB для различного количества эпох.

	Потеря ошибки на наборе данных	
	Обучающих	Проверочных
12 эпох	0.0047	0.0016
25 эпох	0.0029	8.7060e-04
50 эпох	0.0019	6.8230e-04
100 эпох	0.0011	3.9634e-04

ЗАКЛЮЧЕНИЕ

В этой статье мы моделируем проблему прогнозирования финансовых временных рядов. Мы построили модель на основе LSTM-сети для прогнозирования будущих значений котировок валютной пары USD/RUB. После обучения в течение 100 эпох модель показала довольно многообещающие результаты, поскольку она смогла отследить эволюцию цен открытия для этой валютной пары.

Конечно, модель может быть усовершенствована для повышения ее точности, например, путем добавления финансовых индикаторов (акции, биржевые индексы, курсы других валютных пар и т.д.), динамика которых сильно коррелирует с валютной парой USD/RUB, путем изменения структуры LSTM-сети или путем увеличения выборок данных для обучения модели.

СПИСОК ЛИТЕРАТУРЫ

1. Анализ временных рядов [Электронный ресурс] / Викиконспекты, университет ИТМО — Режим доступа: https://neerc.ifmo.ru/wiki/index.php?title=Анализ_временных_рядов
2. Валейко М. *Прогнозирование временных рядов* / М. Валейко, Н. Г. Графеева // Курсовая работа. — Санкт-Петербург : СПбГУ, 2014. — 10 с. — Режим доступа: https://math.spbu.ru/SD_AIS/documents/2014-12-341/2014-12-tw-13.pdf
3. Верзунов С. Н. *Применение глубоких нейронных сетей для краткосрочного прогноза дальности видимости* / С. Н. Верзунов // Проблемы автоматизации и управления — 2019, № 1(36), с. 118–130. — Режим доступа: <https://doi.org/10.5281/zenodo.3253019>
4. Charpentier A. *Cours de séries temporelles : théorie et applications. Volume 1 : Introduction à la théorie des processus en temps discret Modèles ARIMA et méthode Box & Jenkins* / A. Charpentier. — Paris : Université Paris Dauphine-PSL, 2005. — 177 p.
5. Elman J. L. *Finding structure in time* / J. L. Elman // Cognitive Science — 1990, № 14(2), pp.179–211. — Режим доступа: [https://doi.org/10.1016/0364-0213\(90\)90002-E](https://doi.org/10.1016/0364-0213(90)90002-E)
6. MIFDAL R. *Application des techniques d'apprentissage automatique pour la prédiction de la tendance des titres financiers : Mémoire de maîtrise* / R. MIFDAL, E. T. Miresco. — Montréal : Ecole de technologie supérieure, 2019. — 196 p.
7. Pascanu R., Mikolov T. & Bengio Y. *On the difficulty of training recurrent neural networks* / R. Pascanu, T. Mikolov & Y. Bengio // Proceedings of the 30th International Conference on International Conference on Machine Learning — 2013, № 28, pp.III-1310–III-1318. — Режим доступа: <https://dl.acm.org/doi/10.5555/3042817.3043083>
8. Recurrent Neural Networks Cheatsheet [Электронный ресурс] / Amidi A. & Amidi S. — Режим доступа: <https://stanford.edu/~shervine/teaching/cs-230/cheatsheet-recurrent-neural-networks>

9. Rumelhart D. E., Hinton G. E. & Williams R. J. *Learning representations by backpropagating errors* / D. E. Rumelhart, G. E. Hinton & R. J. Williams // *Nature* — 1986, № 323(6088), pp.533–536. — Режим доступа: <https://doi.org/10.1038/323533a0>
10. Werbos P. J. *Generalization of backpropagation with application to a recurrent gas market model* / P. J. Werbos // *Neural Networks* — 1988, № 1(4), pp.339–356. — Режим доступа: [https://doi.org/10.1016/0893-6080\(88\)90007-X](https://doi.org/10.1016/0893-6080(88)90007-X)

Мафутала Грэй Хусен

Воронежский государственный университет, факультет прикладной математики и механики, кафедра математического обеспечения и администрирования информационных систем, студент (ВГУ, ПММ, МОиАИС)

E-mail: graymafutala@gmail.com