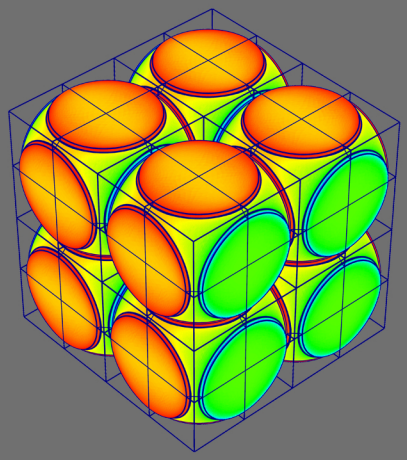
ISSN 2309-3684



Математическое моделирование и численные методы

Облакова Т.В., Алексеев Д.С. Исследование качества прогнозирования временных рядов с помощью реализации модели AT-LSTM на C++. Математическое моделирование и численные методы, 2025, № 1, с. 80–91.

Источник: https://mmcm.bmstu.ru/articles/367/

Параметры загрузки:
IP: 216.73.216.114
23.01.2026 16:01:22

УДК 519.2 DOI: 10.18698/2309-3684-2025-1-8091

Исследование качества прогнозирования временных рядов с помощью реализации модели AT-LSTM на C++

© Т.В. Облакова, Д.С. Алексеев

МГТУ им. Н.Э. Баумана, Москва, 105005, Россия

В настоящей работе рассматривается реализация рекуррентной нейронной сети AT-LSTM (Attention based Long Short Term Memory) на языке программирования C++, разработанная с целью сокращения времени обучения и прямого хода модели. В статье представлены архитектура и примеры работы данной нейросети, описаны подходы к ее обучению и оценке результатов. В ходе исследования были проведены эксперименты по оценке производительности нейросети AT-LSTM при обучении и прямом ходе в сравнении с реализацией на языке Python. Оценка производительности включала измерение времени обучения и времени работы сети при одинаковой длине входных данных, но разных значениях гиперпараметров. Эксперименты показали возможность значительного сокращения времени обучения, снижения ошибки прогнозирования и сохранения высокого качества результатов прогноза при использовании реализации на языке программирования C++. Для того чтобы оценить применимость рассматриваемой реализации AT-LSTM на практике, был проведен анализ качества прогнозирования финансовых временных рядов. В качестве объектов анализа были выбраны курсы валют USD/RUB и EUR/RUB, а также курсы акций компаний Apple (AAPL) и Microsoft (MSFT). Результаты анализа показали, что полученная модель высокоэффективна для прогнозирования временных рядов и может быть успешно применена на практике. На основе проведенных экспериментов и анализа, установлено, что рассматриваемая реализация AT-LSTM на C++ позволяет быстро и качественно обучить модель для дальнейшего прогнозирования временных рядов.

Ключевые слова: *глубокое обучение, рекуррентная нейронная сеть, модель AT-LSTM, модель LSTM, нейросети долгой краткосрочной памяти, механизм внимания, прогнозирование временных рядов, реализация AT-LSTM на C++, исследование качества прогнозирования*

Введение. На сегодняшний день люди все активнее используют нейросети для решения различных задач. Одним из ключевых факторов, влияющих на развитие искусственного интеллекта, является рост вычислительных мощностей современных компьютеров и суперкомпьютеров, что позволяет использовать более сложные архитектуры нейросетей для решения задач.

Обучение рекуррентных нейросетей — это вычислительно трудозатратный процесс, поэтому использование для этих целей языка программирования Python часто приводит к большим временным затратам. Данная проблема может быть решена путем использования более быстрого языка программирования C++.

В работе была рассмотрена задача прогнозирования временных рядов с помощью рекуррентный нейросети AT-LSTM. Архитектура

модели была написана на языке C++ с целью ускорения вычислений и сравнивалась с библиотечной реализацией на языке Python (Keras).

Основные определения и факты. Рекуррентные нейронные сети (Recurrent neural network, RNN) — это класс искусственных нейронных сетей, которые разработаны для обработки последовательностей данных переменной длины. В отличие от классических моделей глубокого обучения, рекуррентные нейросети обладают внутренними циклами или связями, которые позволяют им передавать информацию между последующими временными шагами и хранить информацию о предыдущих состояниях.

Считалось, что рекуррентные нейронные сети трудно обучаемые, потому что они включают в себя миллионы параметров, но недавние достижения в архитектуре ячеек сети, оптимизационных методах и параллельных вычислениях позволили успешно проводить с ними полное обучение [1]. Рассмотрим простейший пример рекуррентной нейронной сети, схема которой представлена на рис. 1.

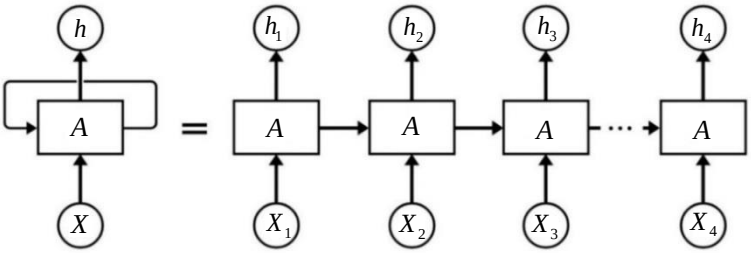


Рис. 1. Схема простейшей однослойной рекуррентной нейронной сети для прогнозирования временных рядов

Математически ее можно записать в следующем виде при $t \in [0,T]$:

$$\begin{gathered} h_t = A\left(W_{xh}x_t + W_{hh}h_{t-1} + b_h\right), \\ y_t = W_{hy}h_t + b_y, \end{gathered} \tag{1}$$

где A — функция активации, x_t — входное значение, h_t — скрытое значение, y_t — прогноз, W_{xh}, W_{hh}, W_{hy} — матрицы весов, b_h, b_y — величины смещения для h и y.

Тем не менее, рекуррентные нейронные сети обладают недостатком, который заключается в том, что нестационарные зависимости, возникающие в течение длительного периода времени, улавливаются хуже. Это явление называется проблемой затухающего градиента.

Таким образом, чем больше элементов в исходной выборке, тем больше шансы, что градиенты обратного распространения либо сойдутся к нулю, либо возрастут к бесконечности. Частично, эту проблему можно решить использованием другой активационной функ-

ции, но наиболее популярным способом решения является применение рекуррентных нейронных сетей, которые строятся на долгой краткосрочной памяти.

Модель LSTM. Одним из наиболее широко используемых типов рекуррентных нейронных сетей для решения задачи прогнозирования финансовых временных рядов является сеть с ячейками долгой кратковременной памяти [2]. Она, способна фиксировать значения как на короткие, так и на длинные интервалы времени. Данная возможность достигается тем, что модель не использует функцию активации внутри своих рекуррентных ячеек.

Таким образом, сохраняемая величина не теряется во времени, и величина штрафа не исчезает при использовании метода обратного распространения ошибки во времени при обучении нейронной сети.

Схематическое представление ячейки LSTM модели отображено на рис. 2.

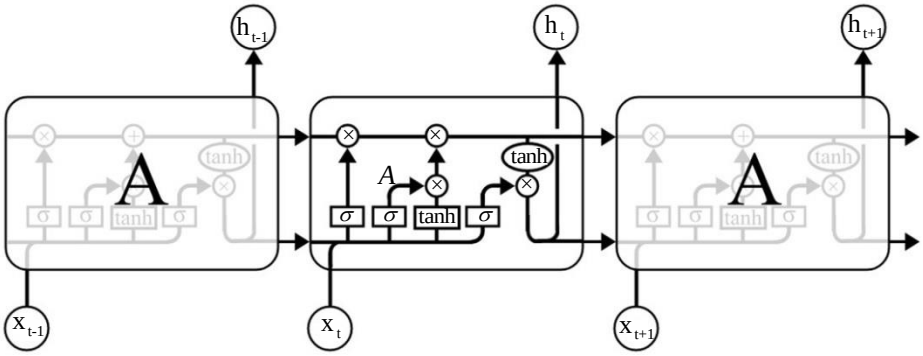


Рис. 2. Схематическое представление LSTM ячейки

Для упрощения описания структуры ячейки LSTM модели объединим вектора x_t и h_{t-1} в матрицу:

$$x_c(t) = [x_t, h_{t-1}]. \tag{2}$$

Тогда мы можем переписать исходную модель ячейки LSTM в виде:

$$\begin{gathered} g(t) = \tanh\left(W_g x_c(t) + b_g\right), \\ i(t) = \sigma\left(W_i x_c(t) + b_i\right), \\ f(t) = \sigma\left(W_f x_c(t) + b_f\right), \\ o(t) = \sigma\left(W_o x_c(t) + b_o\right), \\ s(t) = g(t) \cdot i(t) + s(t-1) \cdot f(t), \\ h(t) = \tanh\left(s(t)\right) \cdot o(t). \end{gathered} \tag{3}$$

Представленный алгоритм также называется вентильным механизмом. Он используется для замены классических функций активации.

Модель AT-LSTM. Модель AT-LSTM [3][4] является разновидностью рекуррентной нейронной сети, которая дополняет стандартную архитектуру LSTM механизмом внимания (AT), чтобы лучше распределять важность исходной информации. Алгоритм внимания может адаптивно выбирать наиболее релевантные входные признаки и придавать больший вес соответствующей исходной последовательности признаков.

Затем выходные данные механизма внимания используются в качестве входных данных модели глубокого обучения LSTM для прогнозирования временного ряда [5]. Как итог, модель AT-LSTM может лучше адаптироваться к разнообразию исходных данных и дает возможность лучше улавливать контекстуальную информацию и релевантные детали. Процесс получения новой взвешенной последовательности описан на рис. 3.

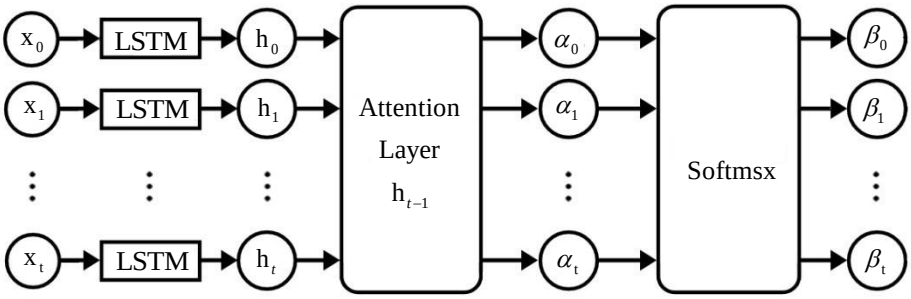


Рис. 3. Схема модели внимания

Первым слоем модели внимания выступают ячейки LSTM для того, чтобы избежать проблемы долгосрочной зависимости. Полученные значения h_t поступают на вход слою:

$$\alpha_k = v^{\mathrm{T}} \tanh\left(W_1 \cdot \left[h_{t-1}, s_{t-1}\right] + W_2 x_k\right), \tag{4}$$

где вектор v^{T} и матрицы W_1, W_2 — обучаемые параметры модели.

Результат модели внимания можно оформить в виде вектора:

$$z_t = \left[\beta_0 x_0, \beta_1 x_1, \ldots, \beta_t x_t\right]^{\mathrm{T}}, \tag{5}$$

где $\beta_k = \mathrm{softmax}\left(\alpha_k\right) = \dfrac{\exp\left(\alpha_k\right)}{\sum_{i=1}^{n} \exp\left(\alpha_i\right)}$.

Окончательно взвешенная последовательность z_t поступает на вход модели глубокого обучения (6) для дальнейшего обучения:

$$\begin{gathered} g(t) = \tanh\left(W_g z_c(t) + b_g\right), \\ i(t) = \sigma\left(W_i z_c(t) + b_i\right), \\ f(t) = \sigma\left(W_f z_c(t) + b_f\right), \\ o(t) = \sigma\left(W_o z_c(t) + b_o\right), \\ s(t) = g(t) \cdot i(t) + s(t-1) \cdot f(t), \\ h(t) = \tanh\left(s(t)\right) \cdot o(t), \end{gathered} \tag{6}$$

где $z_c(t) = \left[z_t, h_{t-1}\right]$.

Рассматриваемая модель благодаря своей структуре менее чувствительна к пропущенным данным последовательности. Также, поскольку AT-LSTM может учиться на большем контексте данных, она может ослабить влияние выбросов на прогноз и быть более устойчивой к шумам.

Однако стоит отметить, что модель также имеет свои слабые стороны. Например, AT-LSTM затрачивает больше времени на обучение и является более требовательной к вычислительным ресурсам по сравнению с более простыми методами, такими как ARIMA или простыми нейронными сетями.

Пример работы AT-LSTM. Рассмотрим полностью обученную однослойную нейросеть AT-LSTM для прогнозирования временных рядов. После полного цикла обучения можно внимательно рассмотреть значения скрытого слоя $h(t)$, который по своей сути содержит в себе «память» о значениях входного ряда, и результат работы модели внимания (рис. 4).

Таким образом, модель глубокого обучения на основе ячеек LSTM при анализе входного временного ряда запоминает информацию о каждом его элементе. В дальнейшем эта информация используется для построения прогноза [6] согласно формуле (1). Заметим, что модель внимания выстраивает новый временной ряд согласно формуле (5).

Ключевым отличием от исходной последовательности является то, что некоторые элементы увеличили или уменьшили свое значение. Это обуславливается тем, что механизм внимания в контексте прогнозирования временных рядов с помощью нейронных сетей позволяет модели динамически выбирать, на какие элементы входной последовательности нужно обратить больше внимания, т.е. модель выделяет определенные элементы ряда, которые оказывают наиболее

весомое влияние на дальнейший прогноз. Это касается различных перепадов, скачков и других изменений временного ряда. [1]

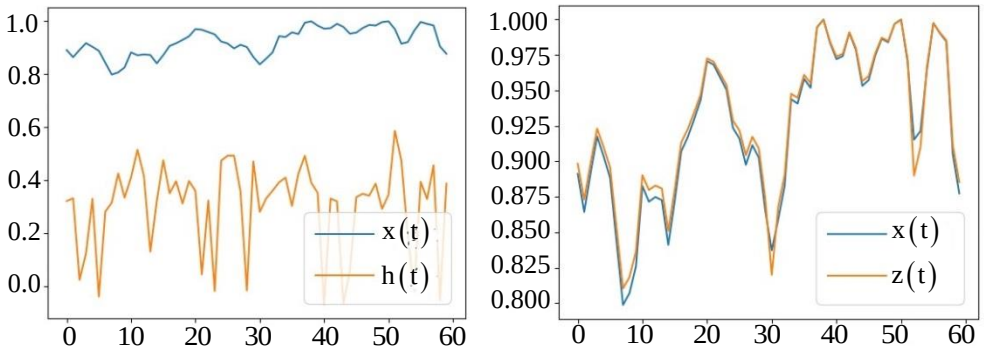


Рис. 4. Значения скрытого слоя $h(t)$ для полностью обученной нейросети LSTM и результат работы модели внимания для входных значений $x(t)$

Также стоит отметить, что модель AT-LSTM предоставляет хорошие результаты прогнозирования временных рядов в сравнении с известными алгоритмами, такими как ARIMA(3, 1, 0) [7], Holt-Winters(5), FBM(0.65) [8], а также с современной нейростью NNETAR [9]. Результаты прогнозирования рассматриваемых моделей изобразим на рис. 5.

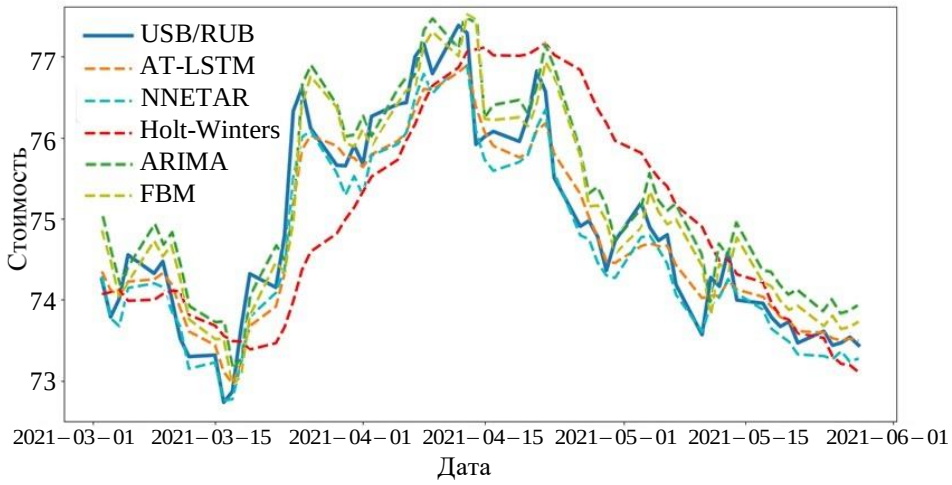


Рис. 5. Совмещенный график прогноза на 1 значение для моделей AT-LSTM, NNETAR, Holt-Winters, ARIMA, FBM для курса доллара USB/RUB

Согласно совмещенному графику точечного прогноза видно (рис. 5), что модель AT-LSTM не уступает по качеству таким моделям, как ARIMA, FBM, NNETAR. При этом модель AT-LSTM предоставляет

результат лучше, чем модель Хольта-Винтерса. Стоит отметить, что прогнозы, полученные с помощью AT-LSTM и NNETAR, достаточно похожи друг на друга. Для более детальной оценки приведем значения метрик RMSE и MAPE в таблице 1.

Таблица 1

Метрики RMSE и MAPE для моделей AT-LSTM, NNETAR, Holt-Winters, ARIMA, FBM при прогнозировании 1 значение

Модель	RMSE	MAPE
ARIMA	0.176	0.005
Holt-Winters	0.338	0.011
NNETAR	0.497	0.006
FBM	0.439	0.005
AT-LSTM	0.330	0.004

Полученные метрики (табл. 1) для прогнозирования рассматриваемых моделей подтверждают факт того, что модель AT-LSTM позволяет получить удовлетворительный прогноз. Ошибка полученного прогноза с помощью данной модели составила 0.4%, что сопоставимо с моделями ARIMA, FBM, NNETAR. Следовательно, можно сделать вывод о том, что модель AT-LSTM является надежной в рамках задачи точечного прогнозирования.

Реализация AT-LSTM на C++. Модель AT-LSTM была разработана под компилятор clang16 на win10 с использованием стандарта C++20. В качестве библиотеки линейной алгебры была выбрана шаблонная библиотека «Eigen» версии 3.4.0. Программная реализация состоит из 4 частей:

1) Класс LSTM, который представляет ячейки LSTM. Содержит состояние ячейки и скрытое состояние, которые обновляются на каждом временном шаге после обработки входных данных.

2) Класс Attention, представляющий слой механизма внимания.

3) Класс Dense, который содержит методы для инициализации параметров и обработки входных данных полносвязного слоя.

4) Класс AT-LSTM, который содержит в себе функции обучения и прямого хода нейросети.

Все весовые матрицы и векторы смещения слоев инициализируются случайно из стандартного нормального распределения с помощью модуля «random». Для более удобного обучения нейронной сети была добавлена возможность работы с батчами (небольшой набор исходных данных). В качестве оптимизационного метода был выбран и реализован алгоритм «Adam», ввиду его адаптивной скорости обучения с использованием скользящих средних градиента.

Оценка производительности AT-LSTM. В этом разделе представлены результаты по измерению времени обучения (рис. 6) и пря-

мого хода (рис. 7) модели AT-LSTM, реализованной в 3 окружениях на 2 языках программирования: C++ (CPU, на центральном процессоре), Python (CPU, на центральном процессоре) и Python (GPU, на графическом процессоре с использованием CUDA). Эксперименты проводились с целью оценить эффективность реализации AT-LSTM на C++ в сравнении с готовыми решениями на языке Python (Keras).

В качестве тестовых данных возьмем произвольный временной ряд на 5000 элементов и разобьем его на обучающую выборку (4900 элементов) и валидационную выборку (100 элементов). Чтобы в полной мере оценить работу алгоритмов, обучение нейронной сети будем производить на 50 эпохах и на разных размерах батча.

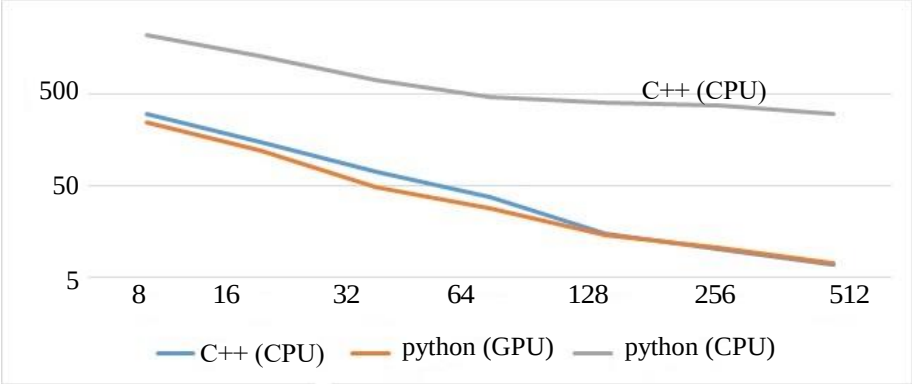


Рис. 6. Логарифм времени обучения модели AT-LSTM в зависимости от размера батча для C++ (CPU), Python (GPU), Python (CPU)

Было получено, что для минимального размера батча (8 элементов последовательности) необходимо затратить в 7 раз меньше времени на обучение нейронной сети на C++ (CPU), чем на Python (CPU). При этом, для максимального размера батча (512 элементов последовательности) необходимо затратить в 44 раза меньше времени. Также, из Рис. 6 видно, что с увеличением размера батча время обучения AT-LSTM на C++ (CPU) приближается к времени обучения на python (GPU). Это можно объяснить более явной специализацией модели на C++, чем на Python.

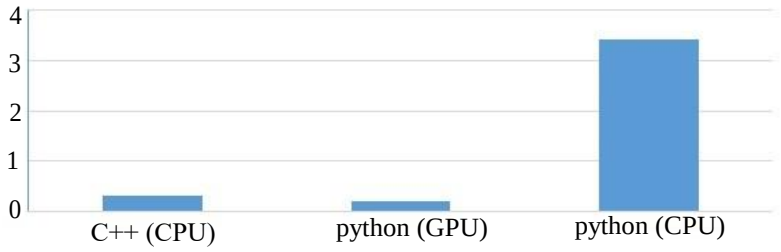


Рис. 7. Время прямого хода нейросети AT-LSTM в секундах для реализаций на C++ (CPU), Python (GPU), Python (CPU)

По рис. 7 видно, что прямой ход реализации нейросети AT-LSTM на C++ (CPU) работает почти в 11 раз быстрее, чем на Python (CPU). При этом скорость выполнения прямого хода на C++ практически приблизился к скорости работы на Python (GPU).

Сравнительный анализ реализаций. Для оценки качества прогнозирования были выбраны 4 временных ряда: USD/RUB, EUR/RUB, AAPL, MSFT [3]. Детальная оценка качества прогноза приведена в табл. 2 [4].

Таблица 2

Метрики RMSE, MAPE и коэффициент корреляции Пирсона для прогноза моделей AT-LSTM (Python/C++)

	Python			C++		
	RMSE	MAPE	Коэф. корр. Пирсона	RMSE	MAPE	Коэф. корр. Пирсона
USD/RUB	0.211	0.008	0.874	0.058	0.008	0.912
EUR/RUB	0.497	0.009	0.853	0.195	0.007	0.885
MSFT	2.306	0.023	0.731	0.648	0.021	0.772
AAPL	2.088	0.029	0.813	2.082	0.023	0.886

По результатам обучения моделей были получены графики прогноза (рис. 8).

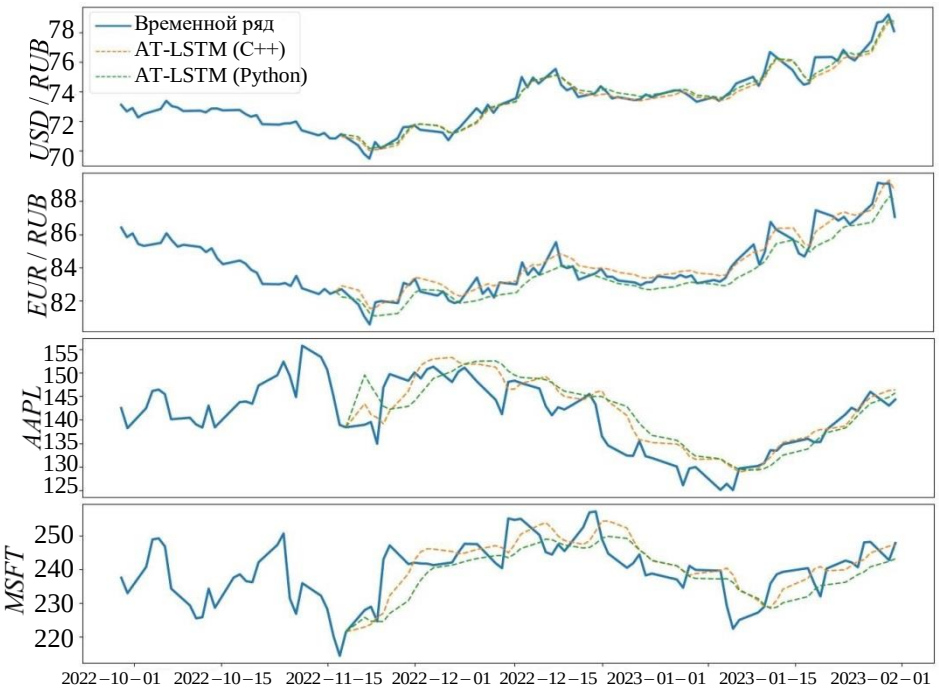


Рис. 8. Совмещенные графики прогноза для моделей AT-LSTM (C++) и AT-LSTM (Python) для USB/RUB, EUR/RUB, AAPL, MSFT

Из графиков видно, что модель AT-LSTM (C++) позволяет получить схожий прогноз, что и модель AT-LSTM (Python). Стоит также заметить, что повысилась точность прогноза за счет использования двойной точности на языке C++.

Выводы. Таким образом, была реализована рекуррентная нейронная сеть AT-LSTM на языке программирования C++. Результаты обучения показали, что модель AT-LSTM (C++) позволяет ускорить процесс обучения нейросети до 40 раз по сравнению с AT-LSTM (Python), а прямой ход реализации на C++ (CPU) работает почти в 11 раз быстрее, чем на Python (CPU). Отметим, что скорость выполнения прямого хода на C++ практически приблизился к скорости работы на Python (GPU). При этом качество прогнозируемых значений не ухудшилось.

ЛИТЕРАТУРА

[1] Макаренко А. В. Глубокие нейронные сети: зарождение, становление, современное состояние. *Проблемы управления*, 2020, № 2, с. 3–19.

[2] Лабусов М.В. Нейронные сети долгой краткосрочной памяти и их использование для моделирования финансовых временных рядов. *Инновации и инвестиции*, 2020, № 3, с. 167–171.

[3] Ширяев А.Н. Вероятностно-статистические модели эволюции финансовых индексов. Обозрение прикладной и промышленной математики. Москва, *ТВП*, 1995, Т. 2, № 4, с. 527–555.

[4] Булинский А.В., Ширяев А.Н. *Теория случайных процессов*. Москва, Физматлит, 2005, 408 с.

[5] Ширяев А.Н. *Основы стохастической финансовой математики. Том 1. Факты. Модели.* Москва, МЦНМО, 2016, 440 с.

[6] Ехлаков Р.С., Судаков В.А. Прогнозирование стоимости котировок при помощи LSTM и GRU сетей. *Препринты ИПМ им. М.В. Келдыша*, 2022, № 17, 13 с.

[7] Алжеев А.В., Кочкаров Р.А. Сравнительный анализ прогнозных моделей ARIMA и LSTM на примере акций российских компаний. *Финансы: теория и практика*, 2020, 24(1), с. 14–23.

[8] Облакова Т.В., Алексеев Д.С. Сравнительный анализ методов моделирования и прогнозирования временных рядов на основе теории фрактального броуновского движения. *Математическое моделирование и численные методы*, 2022, № 4, с. 48–62.

[9] As'ad Mohamad, Sujito Sujito, Setyowibowo Sigit. Neural Network Autoregressive For Predicting Daily Gold Price. *Jurnal INFORM*, 2020, vol. 5, no. 69, 10.25139/inform.v0i1.2715.

Статья поступила в редакцию 06.03.2024

Ссылку на эту статью просим оформлять следующим образом:

Облакова Т.В., Алексеев Д.С. Исследование качества прогнозирования временных рядов с помощью реализации модели AT-LSTM на C++. *Математическое моделирование и численные методы,* 2025, № 1, с. 80–91.

Облакова Татьяна Васильевна – канд. физ.-мат. наук, доцент кафедры «Вычислительная математика и математическая физика» МГТУ им. Н.Э. Баумана. e-mail: obltvu@bmstu.ru

Алексеев Даниил Сергеевич – студент кафедры «Вычислительная математика и математическая физика» МГТУ им. Н.Э. Баумана. e-mail: dreamofskill@gmail.com

Investigation of time series forecasting quality using C++ implementation of AT-LSTM model

© T.V. Oblakova, D.S. Alekseev

Bauman Moscow State Technical University, Moscow, 105005, Russia

The implementation of recurrent neural network AT-LSTM (Attention based Long Short Term Memory) in C++ programming language is considered. This model was developed to reduce the training and inference times. The paper presents the architecture of this neural network and working examples, also describes the training procedure and results estimation. In the research work, calculations were carried out to estimate the performance of the AT-LSTM neural network for training and forward passes in comparison with the Python implementation. The performance analysis included the training time and the running time estimations of the neural network with the same length of input data but different values of hyperparameters. The calculations showed that it is possible to significantly decrease the training time, reduce the prediction error and maintain the high quality of the prediction results when using the C++ programming language implementation. In order to estimate the applicability of the considered AT-LSTM implementation in real cases, the precision of financial time series forecasting was analysed. USD/RUB and EUR/RUB currency rates, as well as Apple (AAPL) and Microsoft (MSFT) share prices were chosen as the objects of current research. The results showed that the obtained model is highly effective for time series forecasting and can be successfully applied in real cases. Finally, based on the experiments and analyses, it was concluded that the considered implementation of AT-LSTM in C++ allows for fast and high-quality training of the model for further time series forecasting.

Keywords*: deep learning, recurrent neural network, AT-LSTM model, LSTM model, long short-term memory neural networks, attention mechanism, time series forecasting, AT-LSTM implementation in C++, forecasting quality research*

REFERENCES

[1] Makarenko A.V. Deep neural networks: origin, formation, current state. *Management problems*, 2020, no. 2, pp. 3–19.

[2] Labusov M.V. Neural networks of long short-term memory and their use for modeling financial time series. *Innovations and investments*, 2020, no. 3, pp. 167–171.

[3] Shiryaev A.N. *Probabilistic and statistical models of the evolution of financial indices* [Review of applied and industrial mathematics], Moscow, TVP, 1995, vol. 2, no. 4, pp. 527–555.

[4] Bulinsky A.V., Shiryaev A.N. *Teoriya sluchajnyh processov* [Theory of random processes]. Moscow, Fizmatlit Publ., 2005, 408 p.

[5] Shiryaev A.N. *Osnovy stohasticheskoj finansovoj matematiki. Tom 1. Fakty. Modeli* [Fundamentals of stochastic financial mathematics. Volume 1. Facts. Models]. Moscow, MCCME Publ., 2016, 440 p.

[6] Ekhlakov R.S., Sudakov V.A. Forecasting the cost of quotes using LSTM and GRU networks. *Keldysh Institute PREPRINTS*, 2022, no. 17, 13 p.

[7] Alzheev A.V., Kochkarov R.A. Comparative Analysis of ARIMA and lSTM Predictive Models: Evidence from Russian Stocks. *Finance: Theory and Practice*, 2020, 24(1), pp. 14–23.

[8] Oblakova T.V., Alekseev D.S. Comparative analysis of methods for modeling and forecasting time series based on the theory of fractal Brownian motion. *Mathematical modeling and numerical methods*, 2022, no. 4, p. 48–62.

[9] As'ad Mohamad, Sujito Sujito, Setyowibowo Sigit. Neural Network Autoregressive For Predicting Daily Gold Price. *Jurnal INFORM*, 2020, vol. 5, no. 69, 10.25139/inform.v0i1.2715.

Oblakova T.V., Cand. Sc. (Phys.-Math.), Assistant Professor of the Computational Mathematics and Mathematical Physics Department, Bauman Moscow State Technical University. e-mail: obltvu@bmstu.ru

Alekseev D.S., Student of the Computational Mathematics and Mathematical Physics Department, Bauman Moscow State Technical University. e-mail: dreamofskill@gmail.com