

ВЛИЯНИЕ НА ДЪЛЖИНАТА НА ИСТОРИЯТА И БРОЯ ЕПОХИ ПРИ ПРОГНОЗИРАНЕТО НА ВРЕМЕВИ РЕД С ПОМОЩТА НА НЕВРОННА МРЕЖА С ДЪЛГА КРАТКОСРОЧНА ПАМЕТ

Кирил Копаранов¹, Красин Георгиев¹, Васил Щерев¹
¹ ТУ-София

INFLUENCE OF THE LOOKBACK PERIOD AND THE NUMBER OF EPOCHS IN THE PREDICTION OF TIME SERIES USING A NEURAL NETWORK WITH A LONG SHORT-TERM MEMORY

Kiril Koparanov¹, Krasin Georgiev¹, Vasil Shterev¹
¹ Technical University of Sofia

Abstract

Forecasting the future values of time series is a problem that occurs in various subject areas such as: finance, meteorology, logistics, etc. The techniques and tools of machine learning have wide application for solving such tasks in recent years. However, they have not been sufficiently studied. The article explores the influence of the lookback period (time lag) and number of training epochs in forecasting time series using a deep neural network with long short-term memory. It is monitored how different combinations of the above parameters affect the accuracy of the forecast, with the aim to reach an optimal combination of their values. As a result of a numerical experiment with an example financial time series, it was found that the use of more lags (over two to four) does not improve the results due to problems in model training. Such a study of the parameters of the model is important for their proper selection in solving practical problems.

Keywords: Deep Learning; LSTM; RNN; Stock Price Predicting; Time Series

ВЪВЕДЕНИЕ

Достоверното и надеждно прогнозиране на бъдещи стойности на времеви редове е от особено значение при вземането на ефективни управленски решения обслужващи и направляващи процесите в най-разнообразни предметни области и също така за да се планират следващите действия в съответния икономически субект [1].

Първоначално за качествено предвиждане на времеви редове са се използвали предимно статистически методи, но в последните 5-10 години следствие на два фактора: Технологичният напредък в областта на хардуера, който доведе до бързо повишаване на изчислителната мощ на графичните процесори от една страна и от друга страна натрупването и съхраняването на го-

леми обеми от данни обусловиха широкото навлизане и използване на дълбоките невронни мрежи в областта на машинното самообучение и съответно до разширяване на сферите на тяхното приложение. Една от тези сфери е и прогнозирането на времеви редове.

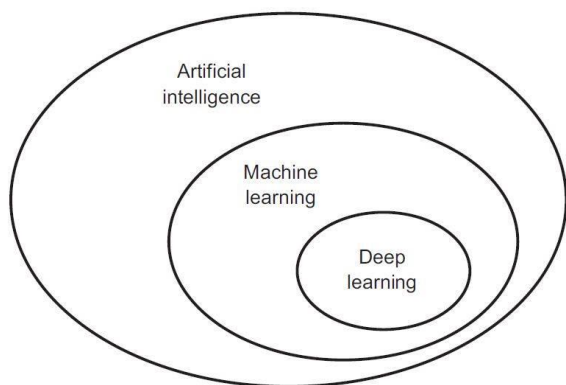
ВРЕМЕВИ РЕДОВЕ

Отличителната черта при времевите редове е, че ако стойностите му са подредени в една колона, то има отделна колона показваща кой ред след кой следва [2]. Метеорологията и борсовата търговия [3] са предметни области, които на пръв поглед нямат нищо общо, но техните показатели като например обем валежи, температура, котировки също могат да бъдат оформени във

времени редове и по един и същи начин да бъдат подложени на обработка и прогнозиране [4].

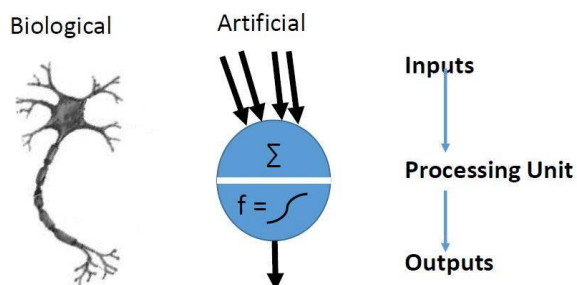
МАШИННО САМООБУЧЕНИЕ

Определение за машинното самообучение въвежда още през 1959 г. Артър Самюъл, според който: “Машинното самообучение е научна област, в която се предоставя на компютрите способността да учат без да бъдат изрично програмирани за тази цел” [5], [6]. Машинното самообучение е подобласт в изкуствения интелект (Фиг. 1), която се отнася до извличането на знания от данните [7].



Фиг. 1. Изкуствен интелект (*Artificial intelligence*), машинно самообучение (*Machine learning*), многослойно/ дълбоко самообучение (*Deep learning*) [8]

Съществена част заемат Изкуствените невронни мрежи, в които основен градивен елемент е изчислителния възел представен на Фиг. 2.



Фиг. 2. Биологичен (*Biological*) и изкуствен (*Artificial*) неврон. Изчислителен модел (*Inputs -> Processing Unit-> Outputs*) [9]

Изкуствените невронни мрежи се обособяват като такива следствие на опитите за

математическо представяне на информационните процеси възникващи и протичащи в биологичните организми [10]

ДЪЛБОКИ НЕВРОННИ МРЕЖИ

Дълбоките невронни мрежи се наричат така, защото съдържат в себе си скрити слоеве от клетки. Скритите слоеве се разположени между входния и изходния слой, които са видими за потребителя. Области на приложение за дълбоките невронни мрежи са компютърното зрение (изображения), генериране на естествен език (текст), автоматично разпознаване на говор (звук) [11].

РЕКУРЕНТНИ НЕВРОННИ МРЕЖИ

За обработка на данни от тип последователности, каквито са и времевите редове, при които има логическа зависимост и под някаква форма предпоставеност между настоящите стойности на наборите от данни и предшестващите ги е най-подходящо включването на рекурентен слой. Отличителната му черта е, че при него говорим за съществуването на памет в съставящите го клетки [12].

Или с други думи осъществява се използване на параметрите в различните части на модела [13]. Една от съществените модификации на клетки при рекурентен слой са тези с дълга краткосрочна памет [14].

ЦЕЛ

В досегашния опит в областта срещаме разработки, в които се сравняват различните методи за прогнозиране в това число и хибридни [15]. Инструментариумът може успешно да се прилага както за предвиждане курса на акциите, така и на борсовите индекси [16] тъй като комбинира в себе си линейни и нелинейни компоненти и показатели. Акцентите на авторите са върху превъзходството в точността и предоставената ни от невронните мрежи по-голяма гъвкавост пред класическите линейни методи като цяло [17]. Затова при това изследване се стремим да съсредоточим вниманието си върху ключовите елементи в процеса на обучението на една рекурентна невронна мрежа с дълга краткосрочна памет - това са

нейните хиперпараметри и върху начина за по-прецизен подбор на техните стойности

ЕКСПЕРИМЕНТ И РЕЗУЛТАТИ

Входни данни:

Входните данни са ни дневните котировки на акциите при отварянето на пазара и най-ниската цена в рамките на същия ден на компанията производител на самолети “Боинг” публикувани в Yahoo finance. Продължителността на периода е от 02-01-1962 до 19-06-2019 и включват 14465 реда (записа).

Предварителна обработка:

За улесняване на изчислителния процес и подобряване работата на модела сме ги мащабирали и така всички стойности са поместени в интервала между 0 и 1.

Мащабирането на данните с функцията MinMaxScaler се прилага тъй като алгоритъмът се представя по-добре и дава по-точни резултати, когато между входните параметри разликите не са прекалено големи. Аналитичния вид на преобразованието е показан в долния израз:

$$\tilde{p}_i = \frac{P_i - P_{\min}}{P_{\max} - P_{\min}}, \quad (1)$$

където:

за i от 1 до T (дължината на времеви ред, периода)

\tilde{p}_i е нормализираната цена (котировка)

P_i е текущата цена (котировка)

P_{\min} е минималната цена (котировка) във времеви ред за целия период

P_{\max} е максималната цена (котировка) във времеви ред за целия период

Архитектура на модела:

Използваме двуслойна невронна мрежа с 4 клетки с дългосрочно краткосрочна памет в първия входен слой и 365 в напълносвързания изходен втори слой. Броят на клетки-

те в изходния слой предопределя и дължината на прогнозния хоризонт. В нашия случай сме се спряли на продължителност от 365 дена т.е. една година.

Интересува ни да установим влиянието на времеви лаг от минали стойности върху точността и надеждността на нашата прогноза. Определили сме невронната ни мрежа да се обучава в рамките на 100 епохи, а обучителната ни партида (batch_size) включва 512 елемента. Измерителят, който използваме е средно квадратична грешка (MSE). Това е средната аритметична стойност на сумата от квадратите на прогнозираните грешки. MSE е стандартен измерител за оценка на разликите между два времеви реда и се получава, както е показано в долния израз:

$$MSE = \frac{1}{T} \sum_{t=1}^T (p_t - \hat{p}_t)^2, \quad (2)$$

където T е дължината на времеви ред, периода

p_t е истинската цена (котировка)

\hat{p}_t е прогнозната цена (котировка)

1 етап:

Ще увеличаваме броя на лаговете като започваме от 1.

Получените резултати са поместени в Табл. 1

Табл. 1. СТОЙНОСТИ НА ОБУЧЕНОСТТА (TRAIN SCORE С ИЗМЕРИТЕЛ MSE) И ПРОГНОЗИРАЩАТА СПОСОБНОСТ (TEST SCORE С ИЗМЕРИТЕЛ MSE) НА МОДЕЛА ПРИ ПРОМЯНА НА ЛАГОВЕТЕ (LOOK_BACK / TIME STEPS) ОТ 1 ДО 200 И, ЗАПАЗЕНИ ПОСТОЯННИ БРОЙ НА ЕПОХИТЕ = 100 И РАЗМЕР НА ОБУЧИТЕЛНАТА ПАРТИДА (BATCH_SIZE) = 512 ЕЛЕМЕНТА

lookback	Train Score	MSE	RMSE	Test Score	MSE	RMSE
1	0.00047844	0.02187334	0.05361184	0.05361184	0.23154232	0.48130258
10	0.00045769	0.02139379	0.04922100	0.04922100	0.30036810	0.54805000
20	0.00045314	0.02128716	0.04958560	0.04958560	0.30097441	0.54869000
50	0.00043278	0.02080340	0.04639137	0.04639137	0.31046960	0.55728000
100	0.00040101	0.02002531	0.04471409	0.04471409	0.29784910	0.54580000
150	0.00036091	0.01899762	0.04368403	0.04368403	0.32596630	0.57100000
200	0.00034060	0.01845543	0.042972	0.042972	0.30071534	0.54800000

Може да се отбележи, че при увеличаването на лаговете над 20 дена модела ни не отбелязва подобрение на точността и достоверността на прогнозата, въпреки непрекъснатото и устойчиво намаляване на показателя измерващ неговата обученост: от

270

0.00047844 в началото при един лаг през 0.00043278 при 50 лага за да достигне до 0.00034060 при 200 лага. Свидетели сме на overfit (модела ни е преобучен като изглежда да е започнал да “наизустява” входящото множество от данни и на практика се стреми да ги повтори/ възпроизведе). Поради това се насочваме към по детайлно разглеждане на стойностите на лаговете в интервала между 1 и 20 дена.

Целта:

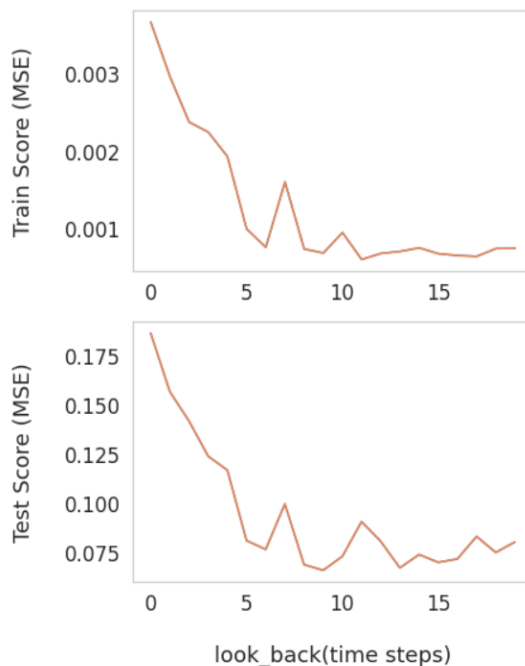
Запазваме размера на обучителната ни партида (batch_size) на 512 елемента. Променяме лаговете от начална стойност 1 ден до крайна стойност 20 дена, като стъпката на нарастването е 1 ден. При брой на епохите равен на 100 преобучеността не се повлиява и не изчезва.

Извършваме намаляване на броя на епохите постъпково и забелязваме непрекъснато устойчиво подобрение на прогнозния резултат, което продължава и след достигането на 20-ата епоха, но ако след това продължим да намаляваме броя на епохите, то след 15-ата имаме влошаване и то стръмно.

При 5 епохи с размер на обучителната ни партида (batch_size) от 512 елемента и нарастване на лаговете (look_back/time steps) от 1 до 20 Train Score (обучаемостта на модела) с измерител MSE варира от 0.00367102 (най-високата стойност се получава, когато лага приема стойност = 1) до 0.00059474 (най-ниската стойност се получава, когато лага приема стойност = 12), а Test Score (прогнозиращата способност на модела) с измерител MSE варира от 0.18636835 (най-високата стойност се получава, когато лага приема стойност = 1) до 0.06605206 (най-ниската стойност се получава, когато лага приема стойност = 10)

Тези резултати са графично илюстрирани на Фиг. 3:

Impact of look_back to precision of the model



Фиг. 3. Влияние на лаговете върху обучаемостта и прогнозиращата способност на модела при зададен брой на епохите = 5 и размер на обучителната партида (batch_size) от 512 елемента

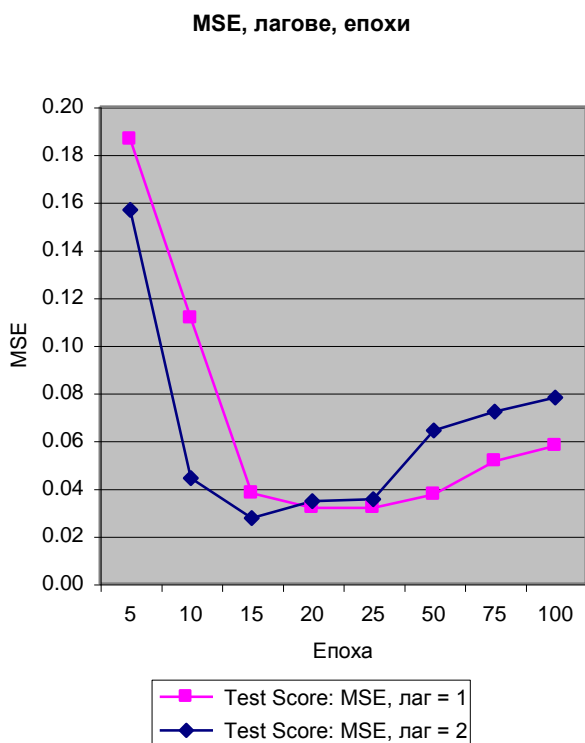
Краен резултат:

Най-добрия резултат, който получаваме за прогнозиращата способност на модела е при 2 лага и 15 епохи -

Test Score: MSE = 0.02752614 е поместен в Табл. 2 и графично илюстриран на Фиг. 4

Табл. 2. НАЙ-ДОБРО СЪЧЕТАНИЕ НА СТОЙНОСТИТЕ НА ЛАГОВЕТЕ И ЕПОХИТЕ ЗА ТЕСТОВИТЕ ДАННИ

епохи	MSE за лаг = 1	MSE за лаг = 2
5	0.1863684	0.15664777
10	0.1112703	0.04423844
15	0.0379302	0.02752614
20	0.0317413	0.03454246
25	0.0317758	0.03543815
50	0.0373919	0.06426059
75	0.0513779	0.07205827
100	0.0578380	0.07803351



Фиг. 4. Зависимост на MSE от лаговете и броя на епохите

ЗАКЛЮЧЕНИЕ

Фокусът на изследването върху компонентите на рекурентната невронна мрежа с дълга краткосрочна памет използвана за едновременното прогнозиране на няколко-стотин стойности в бъдещето ни показва, че увеличаването на броя на лаговете не води до подобряване точността на модела. Не особено високите стойности на лаговете и епохите в най-доброто им съчетание:

Брой лагове = 2

Брой епохи = 15

могат да ни наведат на извода, че е напълно възможно потенциала и обхвата на така приложени инструментариум да не се използва пълноценно и че не е избран достатъчно сполучливо и удачно за да извлече максимално достоверно зависимостите и закономерностите от входните данни на съответната предметна област.

Потвърждава се необходимостта от детайлни, конкретни и комплексни наблюдения и изследвания на хиперпараметрите,

които ще помогнат за техния по-ефективен подбор при конструирането и работата на моделите. Бъдещите стъпки за развитие биха могли да се насочат именно в тази посока: пререструктуриране и модифициране на така подобрения модел за постигане подобрене на резултатите и минимизиране стойността на показателя на средно квадратичната грешка. Допълнителни резерви могат да се търсят и в прилагането на още методи за предварителна обработка на данните.

БЛАГОДАРНОСТИ /ACKNOWLEDGMENT

Този доклад е изготвен в съответствие с проект на Технически Университет - София "Интелигентно управление на трафика" No 201ПП0025-07

This work is supported by the Technical University of Sofia Grant No 201ПП0025-07 "Intelligent traffic management"

REFERENCE

- [1] A. Nielsen, Practical Time Series Analysis. Prediction with Statistics & Machine Learning. Sebastopol, CA: O'Reilly Media, 2020.
- [2] E. Stevens, L. Antiga, and T. Viehmann, Deep learning with PyTorch. Shelter Island, NY: Manning Publications, 2020.
- [3] Akhter Mohiuddin Rather, "A Hybrid Intelligent Method of Predicting Stock Returns", Advances in Artificial Neural Systems, vol. 2014, Article ID 246487, 7 pages, 2014. <https://doi.org/10.1155/2014/246487> Last accessed on 16.09.2020
- [4] Abbot, John & Marohasy, Jennifer. (2015). Using lagged and forecast climate indices with artificial intelligence to predict monthly rainfall in the Brisbane catchment, Queensland, Australia. International Journal of Sustainable Development and Planning. Vol. 10. pp. 29-41. 10.2495/SDP-V10-N1-29-41.
- [5] Samuel, A.L. (1959). Some Studies in Machine Learning Using the Game of Checkers. IBM Journal of Research and Development., vol 3, pp. 210-229.
- [6] A. Geron, Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow. Concepts, Tools, and Techniques to Build Intelligent Systems. Second Edition. Sebastopol, CA: O'Reilly Media, 2019.

- [7] A. C. Müller, and S. Guido, Introduction to Machine Learning with Python. A Guide for Data Scientists. Sebastopol, CA: O'Reilly Media, 2016
- [8] F. Chollet, Deep Learning with Python. Shelter Island, NY: Manning Publications, 2018.
- [9] N. D. Lewis, Deep Time Series Forecasting with Python. An Intuitive Introduction to Deep Learning for Applied Time Series Modeling. USA, 2016.
- [10] C. M. Bishop, Pattern Recognition and Machine Learning (Information Science and Statistics). Berlin, Heidelberg: Springer-Verlag, 2006.
- [11] A. W. Trask, Grokking Deep Learning. Shelter Island, NY: Manning Publications, 2019.
- [12] J. Moolayil, Learn Keras for Deep Neural Networks. A Fast-Track Approach to Modern Deep Learning with Python. Vancouver, BC, Canada: Apress, 2019.
- [13] I. Goodfellow, Y. Bengio, and A. Courville, Deep Learning. Cambridge, MA: MIT Press, 2016.
- [14] S. Hochreiter, and J. Schmidhuber, "Long short-term memory" Neural Computation Vol. 9(8): pp 1735-1780, 1997
- [15] M. Khashei, and M. Bijari, "Which Methodology is Better for Combining Linear and Nonlinear Models for Time Series Forecasting?" Journal of Industrial and Systems Engineering 4 (2011): 265-285.
- [16] M. García, A. Jalal, L. Garzón, and J. Lopez. (2013). Methods for Predicting Stock Indexes. Ecos de Economía. 17. 51-82.
- [17] A. A. Adebisi, A. Adewumi, and C. Ayo, (2014). Comparison of ARIMA and Artificial Neural Networks Models for Stock Price Prediction. Journal of Applied Mathematics. 2014. 1-7. 10.1155/2014/614342.