

Д. И. Милосердов, аспирант, e-mail: dmmil94@yandex.ru,
Санкт-Петербургский институт информатики и автоматизации
Российской академии наук (СПИИРАН)

Архитектурные особенности программных систем нейросетевого прогнозирования с непрерывным обучением

Рассматривается задача поиска целесообразной программной архитектуры, реализующей перспективный метод нейросетевого прогнозирования с непрерывным обучением. Предложен усовершенствованный метод такого прогнозирования, а также процедура ускоренного расчета весов синапсов нейронной сети. Показано, что предложенные архитектурные решения многократно снижают необходимые объемы памяти и вычислений в программных реализациях.

Ключевые слова: рекуррентная нейронная сеть, прогнозирование, программная реализация, оптимизация затрат, аппаратные ресурсы

Введение

В настоящее время идет бурное развитие технологий искусственного интеллекта, направленное на наделение машин широкими когнитивными возможностями [1]. Среди задач, возлагаемых на когнитивные машины, большое внимание уделяют прогнозированию ими различных событий.

Известно большое число методов прогнозирования, использующих как традиционные подходы, так и машинное обучение [2–7]. Основным преимуществом подходов на основе нейронных сетей является их способность к обучению, которая позволяет выявлять скрытые факторы исследуемой среды. Архитектуры нейронных сетей прямого распространения [9–15] просты в реализации и обучении. Они демонстрируют достаточно хорошие результаты в краткосрочном прогнозировании, но по мере увеличения горизонта прогноза их точность снижается. Такие сети не могут учитывать изменений факторов внешней среды, что приводит к необходимости их переобучения. В отличие от них рекуррентные нейронные сети (РНС) [16–22] обладают более высоким потенциалом, но они остаются во многом не исследованными. Не отработаны до конца механизмы ассоциатив-

ного вызова ранее сохраненной информации из памяти таких сетей. Одним из путей преодоления указанных недостатков выступает объединение различных архитектур в гибридные модели [23–26]. Это позволяет несколько расширить возможности систем прогнозирования, но не решает принципиальных проблем, характерных как для сетей прямого распространения, так и для РНС. Кроме того, усложнение общей архитектуры затрудняет управление системой прогнозирования.

К перспективным инструментам прогнозирования следует отнести решения на основе рекуррентных нейронных сетей с управляемыми элементами [27, 28]. В работах [29–31] раскрыт перспективный метод нейросетевого прогнозирования с непрерывным обучением этих РНС. Для полноценной системы, реализующей его, нужна целесообразная программная реализация. С одной стороны, требуется минимизировать объем оперативной памяти, затрачиваемый на функционирование таких систем. С другой стороны, необходимо обеспечить их работу в режиме реального времени. При этом важно не допустить снижения точности прогнозирования. Необходим поиск целесообразных программных решений по реализации перспективного метода нейросетевого прогнозирования событий с непрерывным обучением.

Постановка задачи

Известна [31] обобщенная структура программной системы (рис. 1), реализующей метод нейросетевого прогнозирования событий с непрерывным обучением когнитивных машин.

В этой системе используются две идентичные РНС с управляемыми элементами [27, 28], обозначенные РНС-1 и РНС-2. Сигналы на вход РНС-1 подаются после предобработки. Предобработка предусматривает преобразование исходных сигналов в последовательность совокупностей единичных образов. РНС-1 функционирует непрерывно в режиме обучения. В ней формируется и постоянно обновляется модель воспринимаемых сетью сигналов. Когда необходимо выполнить прогноз, блок управления копирует состояние РНС-1 в РНС-2. После этого в ускоренном времени РНС-2 выполняет прогнозирование. Полученные на выходе РНС-2 последовательные совокупности единичных образов в блоке постобработки преобразуются обратно в соответствующие им исходные сигналы — результаты прогнозирования. Такой подход позволяет, во-первых, одновременно обучаться и выполнять прогнозы, а во-вторых, исключает искажение обучаемой модели из-за переключения между режимами обучения и прогнозирования.

К особенностям РНС с управляемыми элементами относятся следующие их свойства. Эти двухслойные импульсные рекуррентные нейронные сети наделены прозрачными логическими структурами. Такое наделение осуществляется за счет реализации пространственных сдвигов сигналов при передаче их от слоя к слою. В результате сигналы продвигаются вдоль слоев от входа к выходу по заданной (линейной, спиральной, петлевой или другой) схеме. Однозначность между входом и выходом в РНС обеспечивается за счет приоритетности коротких связей между нейронами. Нейронные сети могут находиться в трех состояниях:



Рис. 1. Структура программной системы нейросетевого прогнозирования с непрерывным обучением

ожидание, возбуждение и невосприимчивость. Каждый нейрон одного слоя в общем случае связан со всеми нейронами другого слоя. Связи между нейронами одного и того же слоя отсутствуют. Время задержки сигналов в образуемых двухслойных контурах сети меньше времени невосприимчивости нейронов после их возбуждения. В РНС также предусмотрено управление вызовом сигналов из памяти за счет изменения направлений их ассоциативного взаимодействия в зависимости от текущих состояний слоев сети.

На эмуляцию РНС с N нейронами в каждом слое требуется хранить как минимум один массив длиной $2N^2$. В то же время, дублирование нейросетевого ядра требует почти двукратного увеличения объема памяти. При этом расчет остальных параметров "на ходу" снижает скорость реализации предусмотренных функций нейросетевой обработки сигналов. При ограниченных возможностях применяемых вычислительных средств и больших размерах РНС система может перестать удовлетворять требованиям реального времени, что недопустимо. Необходимо конкретизировать данную структуру таким образом, чтобы оптимизировать затраты памяти и вычислений при сохранении точности прогнозирования.

Усовершенствованный метод нейросетевого прогнозирования

Рассмотренный в предыдущем разделе метод предусматривает копирование состояния элементов РНС-1 в РНС-2 для выполнения прогнозов без прерывания обучения. Согласно ему необходимо дублировать как нейроны РНС-1, так и ее синапсы, что требует увеличения объема памяти почти в два раза.

В целях снижения объема занимаемой оперативной памяти предлагается усовершенствованный метод нейросетевого прогнозирования (рис. 2). Этот метод позволяет отказаться от выделения памяти на хранение синапсов РНС-2.

Согласно методу РНС-2 фактически моделируется РНС-1, память выделяется только под запоминание состояний нейронов для РНС-2. Предлагается как обучение, так и прогнозирование рекуррентной нейронной сети осуществлять в ускоренном времени, а входные сигналы накапливать в буфере. В режиме обучения сети рекомендуется по командам с блока управления быстро считывать и обрабатывать накопленные сигналы. Когда необходимо вы-

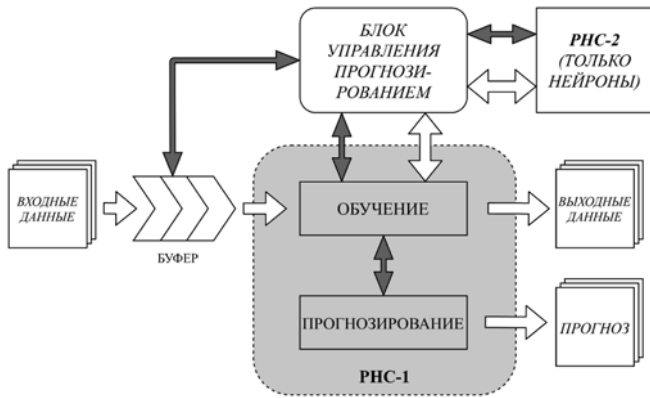


Рис. 2. Усовершенствованный метод нейросетевого прогнозирования с непрерывным обучением

полнить прогноз, подача сигналов с буфера в РНС-1 прерывается, веса ее синапсов замораживаются, а состояния нейронов копируются для последующего использования. Далее РНС-1 выполняет прогноз таким же образом, как РНС-2 в методе-прототипе. После выполнения прогноза сеть инициализируется запомненными на момент прерывания обучения состояниями нейронов. Веса синапсов размораживаются, начинается подача на вход сети сигналов с буфера. РНС-1 снова начинает функционировать в режиме обучения.

В идеале прогнозирование должно осуществляться быстрее, чем время, за которое в РНС-1 поступит следующая последовательность совокупностей единичных образов. Однако теоретически возможны ситуации, когда интервал поступления сигналов невелик, а горизонт прогноза значителен. В этом случае важно не допустить "разрывов" в обучающей выборке — этим оправдано наличие буфера.

Ускоренный расчет весов синапсов

В предлагаемом методе, как и в прототипе, веса синапсов $w_{ij}(t)$, связывающих i -е нейроны с j -ми нейронами, зависят от весового коэффициента $k_{ij}(t)$, определяющего историю обучения, а также от функции $\beta(r_{ij}(t))$ ослабления расходящихся и функции $\eta(r_{ij}(t))$ ослабления сходящихся единичных образов:

$$\begin{aligned} w_{ij}(t) &= k_{ij}(t)\beta(r_{ij}(t))\eta(r_{ij}(t)), \\ k_{ij}(t) &= \text{th}(\gamma g_{ij}(t)/2), \\ g_{ij}(t) &= g_{ij}(t - \Delta t) + \Delta g_{ij}(t), \\ g_{ij}(0) &= 0, \\ \beta(r_{ij}(t)) &= 1/(1 + \alpha(r_{ij}(t))^{1/h}), \end{aligned}$$

где $r_{ij}(t)$ — расстояние между взаимодействующими нейронами; $g_{ij}(t)$, $g_{ij}(t - \Delta t)$ — история

обучения на моменты времени t и $t - \Delta t$ соответственно; γ , α , h — некоторые положительные коэффициенты. Величина $\Delta g_{ij}(t)$ определяется в зависимости от состояний i -го и j -го нейронов в моменты времени t и $t - \Delta t$. Если i -й нейрон сгенерировал единичный образ, и после этого возбудился j -й нейрон, то $\Delta g_{ij}(t)$ положительна. В случае, когда возбуждение j -го нейрона произошло при отсутствии возбуждения на i -го нейрона, $\Delta g_{ij}(t)$ отрицательна. В остальных случаях $\Delta g_{ij}(t) = 0$.

Для ускоренного расчета весов синапсов предлагается учитывать следующее обстоятельство. Весовые коэффициенты $k_{ij}(t)$ являются функциями гиперболического тангенса. Их особенность заключается в том, что при выходе параметра $g_{ij}(t)$ за пределы некоторого интервала $[G^-, G^+]$ его изменения становятся пренебрежимо малы.

Одновременно, поскольку $\Delta g_{ij}(t)$ может принимать определенное положительное (Δg^+) либо отрицательное (Δg^-) значение, существует дискретное множество возможных значений параметра $g_{ij}(t)$.

Принимая во внимание эти два условия, можно вычислить конечное множество $G = \{g_{ij}(t) | g_{ij}(t) = \pm nR(\Delta g^+, \Delta g^-), n = 1, 2, \dots; g_{ij}(t) \in [G^-, G^+]\}$ и соответствующее ему множество $K = \{k_{ij}(t) | k_{ij}(t) = \text{th}(\gamma g_{ij}(t)/2), g_{ij}(t) \in G\}$ значений возможных весовых коэффициентов синапсов. Здесь $R(\Delta g^+, \Delta g^-)$ есть наибольшее значение, на которое нацело делятся Δg^+ и Δg^- .

В случаях, когда множество G невелико, предлагается при инициализации сети рассчитывать значения множеств G и K и хранить их в отдельных массивах. При этом отсутствует необходимость в выделении памяти под полноценный массив элементов $k_{ij}(t)$ для каждого синапса, а весовые коэффициенты рассчитываются в процессе выполнения тактов РНС следующим образом. Для ij -го синапса, у которого необходимо вычислить весовой коэффициент $k_{ij}(t)$, оценивается величина $g_{ij}(t)$. Если $g_{ij}(t) > G^+$, то $k_{ij}(t) = 1$; когда $g_{ij}(t) < G^-$, $k_{ij}(t) = -1$. В остальных случаях определяется элемент множества G , равный $g_{ij}(t)$, после чего определяется соответствующий ему элемент множества K . Значение этого элемента и является весовым коэффициентом ij -го синапса.

Оценка эффективности программных реализаций

Для определения целесообразной архитектуры программной системы нейросетевого

прогнозирования оценивали четыре варианта их реализации.

Первый вариант программного обеспечения (ПО-1) предусматривал реализацию согласно прототипу, раскрытому в постановке задачи. В этом варианте элементы $k_{ij}(t)$ и $\beta(r_{ij}(t))$ хранятся в массивах, элементы $w_{ij}(t)$ рассчитываются "на ходу". Данный вариант приводится для сравнения.

Варианты ПО-2, ПО-3 и ПО-4 предусматривают реализацию с учетом оптимизационных решений, включая использование усовершенствованного метода прогнозирования. Их отличие друг от друга заключается в подходе к соотношению "память—вычисления".

Во втором варианте (ПО-2) значения $\beta(r_{ij}(t))$, $k_{ij}(t)$, $w_{ij}(t)$ хранятся в массивах. Этот вариант отражает стремление максимально сократить объем вычислений, необходимых для функционирования программной системы прогнозирования.

Третий вариант (ПО-3) аналогичен варианту ПО-2, за тем исключением, что для хранения значений $k_{ij}(t)$ и $w_{ij}(t)$ память не выделяется, и они рассчитываются по ходу выполнения тактов. Иными словами, ПО-3 отражает идею максимальной экономии памяти.

Четвертый вариант (ПО-4) использует ускоренный расчет весов синапсов. Весовые коэффициенты $k_{ij}(t)$ хранятся в массиве, но хранение организовано специальным образом, что позволяет без потери точности сократить размер массива. Данный вариант отражает комбинированный подход между оптимизацией памяти и вычислительных затрат.

Для проведения экспериментов использовали компьютер под управлением ОС Windows 7 x64, с процессором Intel Core i7-4790 CPU 3.6 GHz, 32 ГБ ОЗУ. Программное обеспечение разрабатывали на языке программирования C++, с применением фреймворка Qt 5.14.1. Сборку проводили компиляторами MinGW 7.3.0 64bit и MSVC 14.0 amd64.

Тесты проводили на примере прогнозирования наборов слов. Для этого наборы слов кодируются на матрицах размером 20×50 (размер словаря равен 1000 слов), в которых за каждым нейроном закрепляется определенное слово из словаря. Размер РНС для прогнозирования — 4×6 логических полей, число тактов рефракторности нейронов равно 7. Таким образом, число нейронов в слое равняется 24000. Сеть обучается на выборке из 50 наборов, прогнозируются следующие 10 наборов.

Результаты приведены в табл. 1 и 2.

Таблица 1

Показатели эффективности ПО (компилятор MinGW)

Показатели эффективности	ПО-1	ПО-2	ПО-3	ПО-4
Объем памяти, Мбайт	21975,0	2153,8	1078,1	1078,4
Время инициализации ядра, мс	88216	15585	521	541
Время копирования ядра, мс	6322	77	76	77
Время обучения, мс	774133	88960	119104	52133
Время прогнозирования, мс	11655	698	40260	7823

Таблица 2

Показатели эффективности ПО (компилятор MSVC)

Показатели эффективности	ПО-1	ПО-2	ПО-3	ПО-4
Объем памяти, Мбайт	21975,0	2153,8	1078,1	1078,4
Время инициализации ядра, мс	108738	19185	430	464
Время копирования ядра, мс	4730	71	65	72
Время обучения, мс	531297	45745	151093	148211
Время прогнозирования, мс	13244	1240	39604	7074

Как видно из табл. 1 и 2, предложенные архитектурные решения позволяют заметно снизить затраты памяти и вычислений в системах нейросетевого прогнозирования. Так, варианты ПО-2, ПО-3, ПО-4 продемонстрировали, что по сравнению с ПО-1 требуется в среднем в 15 раз меньше оперативной памяти, инициализация ядра происходит быстрее примерно в 16 раз, на копирование нейросетевых ядер затрачивается в среднем в 77 раз меньше времени. Процесс обучения ускоряется в среднем в 6 раз.

Ситуация с временем прогнозирования не так однозначна. Результаты показывают, что расчет параметров синапсов "на ходу" (вариант ПО-3) является слишком дорогостоящим с точки зрения быстродействия. Это свидетельствует о нецелесообразности применения стратегии максимальной экономии памяти. В то же время, комбинированный метод (ПО-4) демонстрирует практически столь же низкие затраты памяти, при этом на прогнозирование затрачивается времени меньше, чем в ПО-1. Однако в ситуациях, когда критерий времени является определяющим, предпочтительно использовать вариант ПО-2, несмотря на то, что

он требует в два раза больше памяти. Таким образом, выбор оптимальной программной структуры имеет смысл осуществлять между ПО-2 и ПО-4, и он сводится к удовлетворению конкретных требований и условий, существующих в среде внедрения программной системы. В системах с высокими требованиями ко времени предпочтительнее использование варианта ПО-2, в обратных ситуациях целесообразнее использовать ПО-4.

Следует отметить, что возможности по оптимизации программно-аппаратных реализаций нейросетевых систем прогнозирования еще не исчерпаны. Все рассмотренные варианты использовали стандартные средства Qt без привлечения дополнительных пакетов обработки данных. В перспективе предполагается рассмотреть дополнительные варианты, использующие как средства распараллеливания вычислений на центральном процессоре, так и графические процессоры.

Заключение

В настоящей работе рассмотрена задача поиска целесообразной программной реализации нейросетевой системы прогнозирования на основе РНС с управляемыми элементами. В основе таких систем лежит метод прогнозирования с непрерывным обучением, предполагающий использование двух нейросетевых ядер, одно из которых непрерывно функционирует в режиме обучения. В момент, когда необходимо выполнить прогноз, блок управления прогнозированием копирует состояние первого ядра во второе ядро, которое выполняет прогноз одновременно с продолжением обучения первого ядра.

В статье рассмотрена типовая программная реализация нейросетевых систем прогнозирования, а затем предложены и оценены архитектурные решения, направленные на ее оптимизацию. Предложен усовершенствованный метод нейросетевого прогнозирования с непрерывным обучением машин и улучшены правила расчета весов синапсов.

Для проведения тестов на языке C++ разработаны различные варианты программных систем прогнозирования. Первый вариант представляет собой "типовую" неоптимизированную реализацию (ПО-1). Варианты ПО-2, ПО-3, ПО-4 разрабатывались с учетом предложенных архитектурных решений. Показано, что они многократно превосходят ПО-1 по

показателям эффективности. Среди них наилучшим образом показали себя реализации ПО-2 и ПО-4. Ответ на вопрос о том, какая из них более эффективна, может быть определен только с учетом конкретных требований и условий, имеющих место в среде внедрения программной системы.

Список литературы

1. Kotseruba Y., Tsotsos J. 40 years of cognitive architectures: core cognitive abilities and practical applications // *Artificial Intelligence Review*. 2018. doi: 10.1007/s10462-018-9646-y.
2. Redmon J., Divvala S., Girshick R., Farhadi A. You Only Look Once: Unified, Real-Time Object Detection // *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2016. doi: 10.1109/CVPR.2016.91.
3. Singh P. Applications of Soft Computing in Time Series Forecasting. Cham, Springer International Publishing Switzerland, 2016. doi:10.1007/978-3-319-26293-2.
4. Timmermann A. Forecasting methods in finance (Review) // *Annual Review of Financial Economics*. 2018. Vol. 10. P. 449–479. doi: 10.1146/annurev-financial-110217-022713.
5. Brockwell P., Davis R. Introduction to Time Series and Forecasting. Springer, 2016. 425 p. doi:10.1007/978-3-319-29854-2.
6. Nagy A., Simon V. Survey on traffic prediction in smart cities. // *Pervasive and Mobile Computing*. 2018. Vol. 50. P. 148–163. doi:10.1016/j.pmcj.2018.07.004.
7. Bok B., Caratelli D., Giannone D., Sbordon A., Tambalotti A. Macroeconomic Nowcasting and Forecasting with Big Data. // *Annual Review of Economics*. 2018. Vol. 10. P. 615–643. doi:10.1146/annurev-economics-080217-053214.
8. Гудфеллоу Я., Бенджио И., Курвилль А. Глубокое обучение. М.: ДМК-Пресс, 2018. 652 с.
9. Скоробогатченко Д. А., Степанова П. Ю. Применение искусственных нейронных сетей при краткосрочном прогнозировании уровня загрузки городских автомобильных дорог // *Известия Волгоградского государственного технического университета*. 2015. № 14(209). С. 24–30.
10. Zhou T., Han G., Xu X., Lin Z., Han C., Huang Y., Qin J. δ -agree AdaBoost stacked autoencoder for short-term traffic flow forecasting // *Neurocomputing*. 2017. Vol. 247. P. 31–38. doi:10.1016/j.neucom.2017.03.049.
11. Egrioglu E., Yolcu U., Bas E., Dalar A. Median-Pi artificial neural network for forecasting // *Neural Computing & Applications*. 2019. Vol. 31. P. 307–316. doi: 10.1007/s00521-017-3002-z.
12. Herrera I., Chicaiza J., Herrera E., Lorente-Leyva L., Caraguay-Procel J., García-Santillán I., Peluffo D. Artificial Neural Networks for Bottled Water Demand Forecasting: A Small Business Case Study // *Advances in Computational Intelligence (IWANN 2019), Lecture Notes in Computer Science*. 2019. Vol. 11507. doi:10.1007/978-3-030-20518-8_31.
13. Chawla A., Singh A., Lamba A., Gangwani N., Soni U. Demand Forecasting Using Artificial Neural Networks—A Case Study of American Retail Corporation // *Applications of Artificial Intelligence Techniques in Engineering. Advances in Intelligent Systems and Computing*, 2019. Vol. 697. doi:10.1007/978-981-13-1822-1_8.
14. Wu W., Wang X. The Coal Demand Prediction Based on the Grey Neural Network Model // *LISS 2014*. Springer, Heidelberg, 2015. doi:10.1007/978-3-662-43871-8_194.
15. Christopher J., Mou J., Yin D. Convolutional Neural Network Deep-Learning Models for Prediction of Shared Bicycle Demand // *International Conference on Applications and Techniques in Cyber Security and Intelligence (ATCI 2018)*.

Advances in Intelligent Systems and Computing, 2019. Vol. 842. doi:10.1007/978-3-319-98776-7_1.

16. Bianchi F., Maiorino E., Kampffmeyer M., Rizzi A., Jensen R. Recurrent Neural Networks for Short-Term Load Forecasting. Springer, 2017. 72 p. doi:10.1007/978-3-319-70338-1.

17. Yang B., Sun S., Li J., Lin X., Tian Y. Traffic flow prediction using LSTM with feature enhancement // Neurocomputing. 2019. Vol. 332. P. 320–327. doi:10.1016/j.neucom.2018.12.016.

18. Tian Y., Zhang K., Li J., Lin X., Yang B. LSTM-based traffic flow prediction with missing data // Neurocomputing. 2018. Vol. 318. P. 297–305. doi:10.1016/j.neucom.2018.08.067.

19. Prokoptsev N. G., Alekseenko A. E., Kholodov Ya. A. Traffic flow speed prediction on transportation graph with convolutional neural networks // Computer research and modeling. 2018. Vol. 10, N. 3. P. 359–367. doi:10.20537/2076-7633-2018-10-3-359-367.

20. Goudarzi S., Kama M., Anisi M., Soleymani S., Doctor F. Self-organizing traffic flow prediction with an optimized deep belief network for Internet of vehicles // Sensors. 2018. Vol. 18. P. 3459. doi:10.3390/s18103459.

21. Krichene E., Masmoudi Y., Alimi A., Abraham A., Chabchoub H. Forecasting Using Elman Recurrent Neural Network. // Intelligent Systems Design and Applications (ISDA 2016). Advances in Intelligent Systems and Computing, Springer, 2017. Vol. 557. doi:10.1007/978-3-319-53480-0_48.

22. Schmidhuber J. Deep learning in neural networks: An overview // Neural Networks. 2015. Vol. 61, P. 85–117. doi:10.1016/j.neunet.2014.09.003.

23. Shen Z., Zhang Y., Lu J., Xu J., Xiao G. A novel time series forecasting model with deep learning // Neurocomputing. 2019. doi:10.1016/j.neucom.2018.12.084.

24. Fan C., Ding C., Zheng J., Xiao L., Ai Z. Empirical Mode Decomposition based Multi-objective Deep Belief Network for short-term power load forecasting // Neurocomputing. 2020. doi:10.1016/j.neucom.2020.01.031.

25. Junxiu L., Tiening S., Yuling L., Su Y., Yi C., Jia Z. An echo state network architecture based on quantum logic gate and its optimization // Neurocomputing. 2020. Vol. 371. P. 100–107. doi:10.1016/j.neucom.2019.09.002.

26. Haitao L. Research on prediction of traffic flow based on dynamic fuzzy neural networks // Neural Computing & Applications. 2016. Vol. 27. P. 1969–1980. doi:10.1007/s00521-015-1991-z.

27. Осипов В. Ю. Нейронная сеть с прошедшим, настоящим и будущим временем // Информационно-управляющие системы. 2011. № 4. С. 30–33.

28. Osipov V., Osipova M. Space–time signal binding in recurrent neural networks with controlled elements // Neurocomputing. 2018. Vol. 308. P. 194–204. doi:10.1016/j.neucom.2018.05.009.

29. Osipov V., Miloserdov D. Neural network forecasting of traffic congestion // Digital Transformation and Global Society (DTGS 2019). Communications in Computer and Information Science, 2020. Vol. 1038. doi:10.1007/978-3-030-37858-5_20.

30. Osipov V., Zhukova N., Miloserdov D. Neural Network Associative Forecasting of Demand for Goods // Experimental Economics and Machine Learning, 2019. Vol. 2479.

31. Osipov, V., Nikiforov, V., Zhukova, N., Miloserdov D. Urban traffic flows forecasting by recurrent neural networks with spiral structures of layers // Neural Computing and Applications. 2020. doi:10.1007/s00521-020-04843-5.

D. I. Miloserdov, PhD Student, e-mail: dmmil94@yandex.ru,

St. Petersburg Institute for Informatics and Automation of the Russian Academy of Sciences,
St. Petersburg, 199178, Russian Federation

Architectural Features of Neural Network Forecasting Software Systems with Continuous Training

In recent years, a method of neural network event forecasting has been developed, based on the use of a pair of recurrent neural networks with controlled elements. This method allows you to make predictions without interrupting training. However, for its full use, a reasonable software implementation is necessary. This study considers the problem of searching for a software architecture that implements the method of neural network forecasting with continuous learning. Offers an improved prediction method that significantly reduces the required amount of memory. A procedure for accelerated calculation of the weights of neural network synapses has been developed. To assess the effectiveness of the proposed architectural solutions, a comparative analysis of various variants of software implementations was conducted. In systems developed with the proposed innovations, the requirements for memory and computing resources are much lower than in software implementations of the prototype method. For example, the amount of memory required has decreased by an average of 15 times, and system initialization has taken 16 times less time. At the same time, the strategy of maximum memory saving in such systems proved to be unproductive compared to the combined approach. Based on the obtained comparison results, recommendations are given for the use and choice of architectures depending on the specific tasks facing the end user, and the hardware and software environment in which the forecasting systems are supposed to operate.

Keywords: recurrent neural network, forecasting, software implementation, costs optimization, hardware resources

DOI: 10.17587/it.26.641-647

References

1. Kotseruba Y., Tsotsos J. 40 years of cognitive architectures: core cognitive abilities and practical applications, *Artificial Intelligence Review*, 2018, doi: 10.1007/s10462-018-9646-y.

2. Redmon J., Divvala S., Girshick R., Farhadi A. You Only Look Once: Unified, Real-Time Object Detection, *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, doi: 10.1109/CVPR.2016.91.

3. Singh P. Applications of Soft Computing in Time Series Forecasting, Cham, Springer International Publishing Switzerland, 2016, doi:10.1007/978-3-319-26293-2.

4. Timmermann A. Forecasting methods in finance (Review), *Annual Review of Financial Economics*, 2018, vol. 10, pp. 449–479, doi: 10.1146/annurev-financial-110217-022713.

5. Brockwell P., Davis R. Introduction to Time Series and Forecasting, Springer, 2016, 425 p. doi:10.1007/978-3-319-29854-2.
6. Nagy A., Simon V. Survey on traffic prediction in smart cities, *Pervasive and Mobile Computing*, 2018, vol. 50, pp. 148–163, doi:10.1016/j.pmcj.2018.07.004.
7. Bok B., Caratelli D., Giannone D., Sbordone A., Tam-balotti A. Macroeconomic Nowcasting and Forecasting with Big Data, *Annual Review of Economics*, 2018, vol. 10, pp. 615–643, doi:10.1146/annurev-economics-080217-053214.
8. Goodfellow Ya., Bengio I., Kourville A. Deep Learning, Moscow, DMK-Press, 2018, 652 p. (in Russian).
9. Skorobogatchenko D. A., Stepanova P. Yu. Application of artificial neural networks with short-term prediction of the level of loading of urban automobile roads, *Izvestiya VolGTU*, no. 14 (209), pp. 24–30 (in Russian).
10. Zhou T., Han G., Xu X., Lin Z., Han C., Huang Y., Qin J. δ -agree AdaBoost stacked autoencoder for short-term traffic flow forecasting, *Neurocomputing*, 2017, vol. 247, pp. 31–38, doi:10.1016/j.neucom.2017.03.049.
11. Egrioglu E., Yolcu U., Bas E., Dalar A. Median-Pi artificial neural network for forecasting, *Neural Computing & Applications*, 2019, vol. 31, pp. 307–316, doi: 10.1007/s00521-017-3002-z.
12. Herrera I., Chicaiza J., Herrera E., Lorente-Leyva L., Caraguay-Procel J., García-Santillán I., Peluffo D. Artificial Neural Networks for Bottled Water Demand Forecasting: A Small Business Case Study, *Advances in Computational Intelligence (IWANN 2019)*, Lecture Notes in Computer Science, 2019, vol. 11507, doi:10.1007/978-3-030-20518-8_31.
13. Chawla A., Singh A., Lamba A., Gangwani N., Soni U. Demand Forecasting Using Artificial Neural Networks—A Case Study of American Retail Corporation, *Applications of Artificial Intelligence Techniques in Engineering*, Advances in Intelligent Systems and Computing, 2019, vol. 697, doi:10.1007/978-981-13-1822-1_8.
14. Wu W., Wang X. The Coal Demand Prediction Based on the Grey Neural Network Model, *LISS 2014*, Springer, Heidelberg, 2015, doi:10.1007/978-3-662-43871-8_194.
15. Christopher J., Mou J., Yin D. Convolutional Neural Network Deep-Learning Models for Prediction of Shared Bicycle Demand, *International Conference on Applications and Techniques in Cyber Security and Intelligence (ATCI 2018)*, Advances in Intelligent Systems and Computing, 2019, vol. 842, doi:10.1007/978-3-319-98776-7_1.
16. Bianchi F., Maiorino E., Kampffmeyer M., Rizzi A., Jensen R. Recurrent Neural Networks for Short-Term Load Forecasting, Springer, 2017, 72 p., doi:10.1007/978-3-319-70338-1.
17. Yang B., Sun S., Li J., Lin X., Tian Y. Traffic flow prediction using LSTM with feature enhancement, *Neurocomputing*, 2019, vol. 332, pp. 320–327. doi:10.1016/j.neucom.2018.12.016.
18. Tian Y., Zhang K., Li J., Lin X., Yang B. LSTM-based traffic flow prediction with missing data // *Neurocomputing*, 2018, vol. 318, pp. 297–305. doi:10.1016/j.neucom.2018.08.067.
19. Prokoptsev N. G., Alekseenko A. E., Kholodov Ya. A. Traffic flow speed prediction on transportation graph with convolutional neural networks, *Computer research and modeling*, 2018, vol. 10, no. 3, pp. 359–367, doi:10.20537/2076-7633-2018-10-3-359-367.
20. Goudarzi S., Kama M., Anisi M., Soleymani S., Doctor F. Self-organizing traffic flow prediction with an optimized deep belief network for Internet of vehicles, *Sensors*, 2018, vol. 18, pp. 3459, doi:10.3390/s18103459.
21. Krichene E., Masmoudi Y., Alimi A., Abraham A., Chab-choub H. Forecasting Using Elman Recurrent Neural Network, *Intelligent Systems Design and Applications (ISDA 2016)*, Advances in Intelligent Systems and Computing, Springer, 2017, vol. 557, doi:10.1007/978-3-319-53480-0_48.
22. Schmidhuber J. Deep learning in neural networks: An overview, *Neural Networks*, 2015, vol. 61, pp. 85–117, doi: 10.1016/j.neunet.2014.09.003.
23. Shen Z., Zhang Y., Lu J., Xu J., Xiao G. A novel time series forecasting model with deep learning, *Neurocomputing*, 2019, doi: 10.1016/j.neucom.2018.12.084.
24. Fan C., Ding C., Zheng J., Xiao L., Ai Z. Empirical Mode Decomposition based Multi-objective Deep Belief Network for short-term power load forecasting, *Neurocomputing*, 2020, doi:10.1016/j.neucom.2020.01.031.
25. Junxiu L., Tiening S., Yuling L., Su Y., Yi C., Jia Z. An echo state network architecture based on quantum logic gate and its optimization, *Neurocomputing*, 2020, vol. 371, pp. 100–107, doi:10.1016/j.neucom.2019.09.002.
26. Haitao L. Research on prediction of traffic flow based on dynamic fuzzy neural networks, *Neural Computing & Applications*, 2016, vol. 27, pp. 1969–1980, doi:10.1007/s00521-015-1991-z.
27. Osipov V. Yu. Neural network with past, present and future time, *Information and Control Systems*, 2011, no. 4, pp. 30–33 (in Russian).
28. Osipov V., Osipova M. Space–time signal binding in recurrent neural networks with controlled elements, *Neurocomputing*, 2018, vol. 308, pp. 194–204, doi:10.1016/j.neucom.2018.05.009 (in Russian).
29. Osipov V., Miloserdov D. Neural network forecasting of traffic congestion, *Digital Transformation and Global Society (DTGS 2019)*, Communications in Computer and Information Science, 2020, vol. 1038, doi:10.1007/978-3-030-37858-5_20.
30. Osipov V., Zhukova N., Miloserdov D. Neural Network Associative Forecasting of Demand for Goods, *Experimental Economics and Machine Learning*, 2019, vol. 2479.
31. Osipov V., Nikiforov V., Zhukova N., Miloserdov D. Urban traffic flows forecasting by recurrent neural networks with spiral structures of layers, *Neural Computing and Applications*, 2020, doi:10.1007/s00521-020-04843-5.