УДК 62-50:004.032.26:621.313.333

Аверьянов Максим Вячеславович, аспирант, Национальный исследовательский Мордовский государственный университет, г. Саранск

Таланов Михаил Викторович, доцент кафедры электроники и электротехники Национальный исследовательский Мордовский государственный университет, г. Саранск

СРАВНИТЕЛЬНЫЙ АНАЛИЗ НЕЙРОСЕТЕВЫХ АРХИТЕКТУР ДЛЯ АДАПТИВНОГО БИХ-ФИЛЬТРА В СИСТЕМЕ УПРАВЛЕНИЯ ЭЛЕКТРОПРИВОДОМ

Аннотация. В статье проведен сравнительный анализ нейросетевых архитектур LSTM, GRU, TCN для расчета коэффициентов адаптивного БИХ-фильтра в системе управления асинхронным электроприводом. Цель — обеспечить точную настройку фильтра по проекциям линейных напряжений, токов в неподвижной системе координат и ошибкам предыдущих оценок в реальном времени на микроконтроллере. Критерии сравнения включают точность предсказания и вычислительную сложность. Необходимость проведения такого анализа обусловлено требованием БИХ-фильтра к нейросети в точности оценки весовых коэффициентов, быстроте вычислений, устойчивости в условиях изменяющихся режимов работы. Сравнительный анализ позволяет оценить, какая сеть лучше всего справляется с этими задачами.

Annotation. The paper presents a comparative analysis of neural network architectures LSTM, GRU, TCN for calculating the coefficients of the adaptive IIR filter in the control system of asynchronous electric drive. The objective is to provide accurate filter tuning from projections of linear voltages, currents in a fixed coordinate system and errors of previous estimates in real time on a microcontroller. Comparison criteria include prediction accuracy and computational complexity. The need for such an analysis is necessitated by the requirement of the IIR filter for the neural network in accuracy of weight coefficient estimation, speed of computation,

and robustness under changing operating conditions. Comparative analysis allows us to evaluate which network copes best with these tasks.

Ключевые слова: электропривод, векторное управление, наблюдатель состояний, нейросетевая модель, математическая модель.

Keywords: electric drive, vector control, state observer, neural network model, mathematical model.

Система управления получает на вход векторные проекции линейных напряжения и тока (координаты α, β в неподвижной системе) и текущую ошибку оценки e(k), а нейросеть должна в каждый момент времени выдавать коэффициенты адаптивного БИХ-фильтра. Задачей адаптивного фильтра является определение коэффициентов z-передаточной функции для дальнейшего определения параметров электродвигателя. Математическая модель такого фильтра основана на дифференциальных уравнениях асинхронного электродвигателя [1]. В данной статье будут проанализированы пять архитектур (LSTM, GRU, TCN и простую полносвязную сеть) по ключевым критериям: способность моделировать временные зависимости, эффективность вычислений, применимость в режиме реального времени, сложность обучения/объем данных и возможность встраивания (генерации кода Simulink/MATLAB).

LSTM (LONG SHORT-TERM MEMORY) имеет специальную ячейку памяти и фильтры (входной, забывания, выходной), позволяющие сохранять информацию о прошлых входах на длительных отрезках времени. Благодаря этому LSTM хорошо справляется с моделированием долгосрочных зависимостей во временных рядах. За счёт богатой структуры с несколькими фильтрами каждая LSTM-ячейка требует нескольких матричных умножений на шаг и большого числа параметров. Это приводит к существенному расходу памяти и вычислений [2]. Такие решения повышают вычислительные накладные расходы по сравнению с более простыми моделями [2]. LSTM обрабатывает последовательность пошагово, что даёт детерминированную задержку, но может быть медленным при большой длине последовательности.

Высокая вычислительная нагрузка и необходимость обновлять состояние в каждом такте увеличивают задержку отклика, что нежелательно в системах управления с жёсткими временными ограничениями. Специальная архитектура LSTM сглаживает проблему затухающих/взрывающихся градиентов при обучении на длинных сериях, однако сама по себе требует много данных и вычислений для надёжного обучения. Большое число параметров означает долгий процесс обучения и риск переобучения при малом объёме данных. LSTM широко поддерживается в инструментах MATLAB/Simulink (Deep Learning Toolbox) и может быть скомпилирована в C-код для микроконтрроллеров. Однако с учётом ограничения памяти такая модель должна быть сильно упрощена. LSTM надёжно моделирует сложную динамику системы благодаря памяти состояния. При этом данная архитектура очень ресурсоёмкая и сложная. Большой объём весов и вычислений делает LSTM неприемлемой для жёстко ресурсно-ограниченных систем.

GRU (GATED RECURRENT UNIT) – это упрощённая версия LSTM с двумя фильтрами (обновления и сброса) и без отдельного выходного фильтра. Она также хранит состояние и умеет учитывать предыдущие входы, формируя память. Несмотря на более простой вид, GRU моделирует временные зависимости почти так же эффективно, как LSTM [4]. GRU также является рекуррентной сетью, но из-за упрощённой архитектуры её расчёты на шаг выполняются быстрее, чем у LSTM. Тем не менее, последовательная зависимость (нужна обработка предыдущего состояния) сохраняется. В целом GRU предлагает более низкую задержку и меньшую нагрузку на MCU по сравнению с LSTM, что важно для систем реального времени. Наличие менее сложной структуры упрощает задачу обучения по сравнению с LSTM. GRU требует меньше данных и итераций для достижения сопоставимой точности. Она хорошо подходит для задач адаптации с ограниченным объёмом данных и средней длительностью зависимостей. GRU поддерживается в современных фреймворках и, доступен для кодогенерации(gruLayer). Небольшая архитектура облегчает её встраивание в Simulink/Embedded Coder. GRU-

модель обычно существенно компактнее LSTM при близких характеристиках результата [4]. Но GRU всё ещё требует значительных вычислительных ресурсов.

TCN (Temporal Convolutional Network) – это свёрточная сеть со сквозными и расширенными свёртками. Благодаря расширенной памяти она может охватывать очень длинный контекст при умеренном числе слоёв. TCN обрабатывает только текущий и прошлые входы, что сохраняется временная последовательность данных. Эта архитектура эффективно моделирует долгосрочные зависимости без рекурсии [2]. В отличие от рекуррентных сетей, все вычисления в TCN независимы и параллелизуемы, что упрощает обучение и вывод результатов работы обученной модели. Отмечено, что TCN может быть в ~100 раз компактнее по параметрам, чем гибрид CNN-LSTM на том же наборе данных [2]. TCN обычно обучаются быстрее и стабильнее, чем LSTM. В исследовании [5] обнаружено, что для одинаковой задачи TCN достигал лучшей точности с меньшим числом эпох по сравнению с LSTM. Более того, параллельность обучения свёрток ускоряет оптимизацию по сравнению с пошаговым RNN. Поскольку TCN – это, фактически, каскад свёрточных слоёв с остаточными блокми, её можно скомпилировать средствами Deep Learning Toolbox (обычно через GPU Coder или MATLAB Coder) аналогично обычной CNN [2]. Данная архитектура сочетает сильную способность моделировать длительный контекст с относительной компактностью [2]. Данная нейросеть быстро конвергирует весовые коэффициенты сети [5] и не страдает проблемой взрыва градиента. При этом у TCN более сложная конфигурация, также требуется буферизация и вычисление свёрток, что может быть сложно оптимизировать при малых вычислительных ресурсах.

FNN (Feedforward Neural Network) – нейронная сеть прямого распространения. Это самый базовый тип нейросети. В чистом виде FNN не хранит историю: каждая оценка зависит только от текущего входа. Чтобы учесть динамику, в FNN нужно вручную передавать задержки или ошибки

предыдущих шагов как дополнительные входы. Без такой явной «памяти» сеть не может самостоятельно моделировать длительные временные зависимости. FNN обладает наименьшей сложностью из всех перечисленных. Сеть может быть очень компактной, если число слоёв и нейронов небольшое. Обучение простое (обычный градиентный спуск). Сеть может требовать большого объёма данных, если важны данные за предыдущие итерации, но для задач с коротким «памятным» горизонтом FNN достаточно эффективна. Поддерживается повсеместно: в Simulink FNN-сети легко вставляются и компилируются в C/C++. Кодогенерация полносвязных сетей стандартна и при малых размерах сетей не вызывает проблем с ресурсами. Данная архитектура максимально простая. Отличается наименьшей вычислительной нагрузкой среди всех вариантов. Однако данная архитектура неспособна самостоятельно учитывать историю без явного расширения входа. Для динамических задач требуется вручную включать задержанные признаки, что может сильно усложнить модель. Полносвязная сеть может не успевать адекватно реагировать на динамику системы управления без дополнительной информации.

Вывод: С учётом всех факторов наиболее универсальным компромиссом выглядит GRU. Эта архитектура сочетает рекуррентный характер для учёта прошлых ошибок с относительно низкой сложностью и скоростью вычислений [4]. GRU обычно компактнее и быстрее LSTM при сопоставимом качестве. Если же задача позволяет немного большую сложность и требуется моделировать длинные последовательности, альтернативой может быть TCN, так как она имеет значительно меньший размер модели при такой же точности [2], а также быстрее обучается [5], но сложнее в реализации. В крайних случаях, когда ресурсы ограничены и динамика системы проста, можно рассмотреть и полносвязную сеть, хотя она может требовать ручного задания окна.

Литература

1. Таланов М.В., Таланов В.М Алгоритм идентификации электрических параметров асинхронного электродвигателя с использованием адаптивного БИХ-фильтра. // V Международная научная конференция по проблемам управления в технических системах (ПУТС-2023). СПб: СПбГЭТУ «ЛЭТИ», 2023. С. 135-137, doi: 10.1109/CTS59431.2023.10288770.
2. Saha S. S., Sandha S. S., Srivastava M. Machine Learning for Microcontroller-Class Hardware: A Review // IEEE Access. — 2020. — Vol. 8. — P. 108198–108221, doi: 10.1109/ACCESS.2020.3000843.
3. MathWorks. Generate Code for LSTM Network and Deploy on Cortex-M Target // MATLAB Help Center. — [Электронный ресурс]. — Режим доступа: https://www.mathworks.com/help/coder/ug/generate-code-for-lstm-network-and-deploy-on-cortex-m.html
4. Khodke, Y. and Deshpande, S. Stock Price Prediction Using LSTM and GRU // International Journal for Research in Applied Science & Engineering Technology (IJRASET). — 2023. — Vol. 11, Issue VI (June). — P. 1163–1167. — ISSN 2321-9653. — DOI: 10.22214/ijraset.2023.53826.
5. Gopali, S., Abri, F., Siami-Namini, S., Siami Namin, A. A Comparative Study of Detecting Anomalies in Time Series Data Using LSTM and TCN Models [Electronic resource] // arXiv preprint, 2021. — arXiv:2112.09293. — Available at: https://doi.org/10.48550/arXiv.2112.09293.
6. Козлова, Л. Е. Разработка нейросетевого наблюдателя угловой скорости ротора в электроприводе по схеме ТРН-АД: специальность 05.09.03 "Электротехнические комплексы и системы": диссертация на соискание ученой степени кандидата технических наук / Козлова Людмила Евгеньевна, 2016. – 144 с.

Literature

1. Talanov M.V., Talanov V.M Algoritm identifikatsii elektricheskikh parametrov asinkhronnogo elektrodvigatelya s ispol'zovaniem adaptivnogo BIKh-fil'tra. // V Mezhdunarodnaya nauchnaya konferentsiya po problemam

upravleniya v tekhnicheskikh sistemakh (PUTS-2023). SPb: SPbGETU «LETI», 2023. pp. 135-137. 10.1109/CTS59431.2023.10288770.

2. Saha S. S., Sandha S. S., Srivastava M.Machine Learning for Microcontroller-Class Hardware: A Review // IEEE Access. — 2020. — Vol. 8. — P. 108198–108221. – : 10.1109/ACCESS.2020.3000843.
3. MathWorks. Generate Code for LSTM Network and Deploy on Cortex-M Target [Electronic resource]. – Available at: https://www.mathworks.com/help/coder/ug/generate-code-for-lstm-network-and-deploy-on-cortex-m.html.
4. Khodke, Y. and Deshpande, S. Stock Price Prediction Using LSTM and GRU // International Journal for Research in Applied Science & Engineering Technology (IJRASET). — 2023. — Vol. 11, Issue VI (June). — P. 1163–1167. — ISSN 2321-9653. — DOI: 10.22214/ijraset.2023.53826.
5. Gopali, S., Abri, F., Siami-Namini, S., Siami Namin, A. A Comparative Study of Detecting Anomalies in Time Series Data Using LSTM and TCN Models [Electronic resource] // arXiv preprint, 2021. — arXiv:2112.09293. — Available at: https://doi.org/10.48550/arXiv.2112.09293
6. Kozlova, L. E.Development of a Neural Network Observer of Rotor Angular Velocity in an Electric Drive Based on the TRN-IM Scheme: PhD thesis in Technical Sciences, specialty 05.09.03 – Electrical Complexes and Systems / Lyudmila Evgenevna Kozlova. — 2016. — 144 p.