

## ОТКРЫТАЯ БИБЛИОТЕКА ДЛЯ АВТОМАТИЗАЦИИ СОЗДАНИЯ АРХИТЕКТУР СВЕРТОЧНЫХ НЕЙРОННЫХ СЕТЕЙ

Кузнецов А.Д. (Университет ИТМО)

Борисова Ю.И. (Университет ИТМО)

Научный руководитель - Борисова Ю.И. (Университет ИТМО)

**Введение.** При конструировании архитектуры сверточных нейронных сетей (CNN), чтобы реализовать корректно работающую модель, следует не только уделять внимание размерностям входных данных, но также вручную отслеживать их изменения в процессе прохождения через сверточные слои. Дополнительные сложности возникают за счет глубины моделей, многомерной природы данных, внедрения дополнительных слоев нормализации [1, 2], pooling или sampling слоев и различной настройки сверточных или транспонированных сверточных последовательностей. Более того, гиперпараметры каждого слоя зависят от предыдущего, и, следовательно, размерность входных данных влияет не только на их подбор и конфигурацию, но и на количество этих слоев. На данный момент не существует готовых решений, обеспечивающих возможность реализации автоматической сборки сверточной последовательности по некоторым заданным пользователем параметрам.

**Основная часть.** Тестирование и апробация реализуемого фреймворка проводится на задаче прогнозирования  $n$ -мерных временных рядов (карты концентрации льда в северных морях) [3], для которых используется encoder-decoder CNN-архитектура. Для решения задачи был добавлен дополнительный инструментарий для агрегации многомерных временных данных. Предложенные алгоритмы позволяют разбивать обучающую выборку для последующего предсказания как для всей предыстории данных, так и для “скользящего” окна с использованием заданного пользователем лага. Более того, есть возможности первичной предобработки данных в виде их выборочной бинаризации или объединения нескольких временных рядов посредством конкатенации по одному из измерений или формирования параллельных тренировочных датасетов для ансамблевых архитектур [4].

Основной инструментарий программного обеспечения представляет собой классы и методы, обеспечивающие итеративное создание сверточных и транспонированных сверточных последовательностей с выбором гиперпараметров слоев, их количества, видом нормализации и функций активации между слоями. В случае необходимости более тонкой настройки слоев архитектуры представленные методы позволяют строить подобные последовательности из отдельных сверточных блоков наподобие U-Net [5] или других CNN-моделей. Такие структуры являются набором нескольких сверточных операций, не меняющих размерность тензора, но в совокупности с функциями активации они оказывают нелинейное преобразование данных. За счет реализации основных алгоритмов в рамках ООП парадигмы есть возможность отслеживания истории изменения размеров тензоров и истории количества каналов для каждой собранной сверточной последовательности.

Разработанная библиотека TorchCNNBuilder открыто публикуется [<https://github.com/ITMO-NSS-team/TorchCNNBuilder>] и является нативной надстройкой для фреймворка PyTorch [6], не имея при этом каких-либо других зависимостей.

Подобно PyTorch ПО поддерживает использование одномерных, двумерных и трехмерных сверточных операций вместе с соответствующими видами слоев нормализации и функциями расчета выходных размерностей тензора. Также существуют готовые шаблоны CNN-архитектур [3], которые отличаются от своих аналогов адаптивностью и возможностью настройки слоев при инициализации за счет использования автоматически собираемых сверточных последовательных.

**Выводы.** Открытый фреймворк обладает интуитивно понятным инструментарием, уже сейчас позволяющим создавать CNN-модели с помощью сверточных последовательностей и блоков. Имея при этом минимальное количество зависимостей, подобный продукт может использоваться и с другими открытыми ПО, а также участвовать в экосистеме популярных AutoML решений как FEDOT [7]. Полученное решение планируется поддерживать и улучшать посредством расширения спектра покрываемых задач, но уже сейчас оно может помочь исследователям в применимости CNN моделей, взяв на себя весомую часть написания кода архитектуры.

#### **Список использованных источников:**

1. Ioffe S., Szegedy C. Batch normalization: Accelerating deep network training by reducing internal covariate shift // International conference on machine learning. – PMLR, 2015. – Pp. 448-456.
2. Wan L. et al. Regularization of neural networks using dropconnect // International conference on machine learning. – PMLR, 2013. – Pp. 1058-1066.
3. Borisova J., Titov R., Shakhkhan K., Hvatov A. Forecasting of Sea Ice Concentration using CNN, PDE discovery and Bayesian Networks // Procedia Computer Science. – Volume 229. – 2023. – Pp. 177-187. – ISSN 1877-0509. – DOI: [10.1016/j.procs.2023.12.019](https://doi.org/10.1016/j.procs.2023.12.019)
4. Borisova J., Nikitin N. Surrogate Modelling for Sea Ice Concentration using Lightweight Neural Ensemble // arXiv preprint arXiv:2312.04330. – 2023. URL: [2312.04330.pdf \(arxiv.org\)](https://arxiv.org/pdf/2312.04330.pdf)
5. Ronneberger O., Fischer P., Brox T. U-Net: Convolutional Networks for Biomedical Image Segmentation // arXiv preprint arXiv:1505.04597. – 2023. URL: [arxiv.org/pdf/1505.04597.pdf \(arxiv.org\)](https://arxiv.org/pdf/1505.04597.pdf)
6. Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., ... Chintala, S. PyTorch: An Imperative Style, High-Performance Deep Learning Library // Advances in Neural Information Processing Systems 32. – Curran Associates, Inc., 2019. – Pp. 8024-8035. – URL: [papers.neurips.cc/paper/9015-pytorch-an-imperative-style-high-performance-deep-learning-library.pdf](https://papers.neurips.cc/paper/9015-pytorch-an-imperative-style-high-performance-deep-learning-library.pdf)
7. Nikitin N.O., Vychuzhanin P., Sarafanov M., Polonskaia I.S., Revin I., Barabanova I.V., Maximov G., Kalyuzhnaya A.V., Boukhanovsky A., Automated evolutionary approach for the design of composite machine learning pipelines, Future Generation Computer Systems, Volume 127, 2022, Pp. 109-125, ISSN 0167-739X, <https://doi.org/10.1016/j.future.2021.08.022>.