

## ПЕРЕВОД НЕЙРОННОЙ СЕТИ В ВЕКТОРНОЕ ПРЕДСТАВЛЕНИЕ

### TRANSLATION OF A NEURAL NETWORK INTO A VECTOR REPRESENTATION

V. Shalamov  
V. Efimova  
A. Filchenkov

*Summary.* Recently, deep learning models, in particular neural networks, have been widely used in the world. Many neural network architectures are created manually by humans, which does not allow achieving optimal results. There are neural network architecture search algorithms that allow you to automate the process of searching and creating neural network architecture. Recent studies have shown that if you use a vector representation of a neural network, you can reduce the search for neural network architecture. An algorithm is proposed that converts a neural network into a graph vector representation and vice versa, which minimizes information loss during conversion. Three compression models were considered: an auto-encoder, a variational auto-encoder, and a sequence-in-sequence. The variations of the proposed model were compared with each other and with well-known models based on a variational auto-encoder: D-VAE and DVAE-EMB by compression losses and by the Kullback-Leibler divergence. The results of the work of neural networks compressed and decompressed using the proposed method remain no worse than those of existing methods, while the proposed solution allows achieving a smaller dimension of the hidden space using the graph structure and encoding attributes separately from the vertices of the graph.

*Keywords:* neural network architecture search, vector representation, variational auto-encoder, graphs with attributes.

**Шаламов Вячеслав Владимирович**

Аспирант

Университет ИТМО

Россия, г. Санкт-Петербург

sslavian812@yandex.ru

**Ефимова Валерия Александровна**

Аспирант

Университет ИТМО

Россия, г. Санкт-Петербург

valeryefimova@gmail.com

**Фильченков Андрей Александрович**

К.ф.-м.н.

Университет ИТМО

Санкт-Петербург

afilchenkov@itmo.ru

*Аннотация.* В последнее время в мире широко применяются модели глубокого обучения, в частности нейронные сети. Многие архитектуры нейронных сетей создаются человеком вручную, что не позволяет достичь оптимальных результатов. Существуют алгоритмы поиска архитектуры нейронных сетей, которые позволяют автоматизировать процесс поиска и создания архитектуры нейронных сетей. В недавних исследованиях было показано, что если использовать векторное представление нейронной сети, то можно сократить поиск архитектуры нейронной сети. Предложен алгоритм, переводящий нейронную сеть в графовое векторное представление и обратно, который минимизирует потери информации при конвертации. Были рассмотрены три модели для сжатия: автокодировщик, вариационный автокодировщик и последовательность-в-последовательность. Было проведено сравнение вариаций предложенной модели между собой и с известными моделями на основе вариационного автокодировщика: D-VAE и DVAE-EMB по потерям при сжатии и по расхождению Кульбака-Лейблера. Результаты работы сжатых и разжатых с помощью предложенного метода нейронных сетей остаются не хуже, чем у существующих методов, при этом предложенное решение позволяет достичь меньшую размерность скрытого пространства, используя структуру графа и кодируя атрибуты отдельно от вершин графа.

*Ключевые слова:* поиск архитектуры нейронной сети, векторное представление, вариационный автокодировщик, графы с атрибутами.

### Введение

**В** последнее время во многих сферах информационной индустрии применяются глубокие нейронные сети: для перевода текста, распознавания объектов и текста на изображении, предсказания и анализа событий. До сих пор многие архитектуры нейронных сетей создаются человеком вручную, что не позволяет достичь оптимальных результатов. Поэтому появились алгоритмы и модели для поиска ар-

хитектуры нейронных сетей, которые позволяют автоматизировать процесс поиска и создания архитектуры нейронных сетей [1]. Важная часть таких алгоритмов — правильный выбор пространства для поиска архитектуры: оно выбирается под конкретную задачу и тем самым существенно ограничивает возможные варианты получаемых архитектур.

В недавних исследованиях было показано, что можно уменьшить пространство поиска архитектуры ней-

ронной сети, используя векторное представление нейронных сетей как графов [2, 3]. Однако, в слоях нейронной сети содержится дополнительная информация — их атрибуты, а чтобы представить вершину графа, данные методы используют лишь тип слоя, что ограничивает пространство поиска.

Обращаясь к существующим решениям CAN [4] и SAT [5], можно заметить, что архитектура нейронной сети представляется в виде вектора не полностью: сохраняется только тип слоя, не позволяя получать дополнительные параметры в виде атрибутов вершины, либо невозможно представить в виде вектора ориентированные ациклические графы, которыми являются нейронные сети.

Целью данной работы является представление нейронной сети в виде вектора, построенного на основе графа вычислений с атрибутами слоев. В данной работе рассматриваются сверточные нейронные сети, однако этот метод, может быть обобщает на другие нейронные сети, если поддерживать весь необходимый набор операций и атрибутов.

## Метод

Сначала рассмотрим **прямое преобразование нейронной сети в векторное представление**. Сперва необходимо представить архитектуру нейронной сети в виде графа. Для этого каждый слой или операция нейронной сети представляется как вершина графа, а связь между слоями — как ребра графа. Список поддерживаемых слоев и операций: Conv, LeakyRelu, MaxPool, Flatten, Linear, Sigmoid, BatchNorm, Relu, Add, GlobalAveragePool, AveragePool, Concat, Pad, ReduceMean, Tanh, ConvTranspose, Slice, Elu, Reshape, Mul [6]. В вершине графа содержится дополнительная информация о слое — его атрибуты. Список поддерживаемых атрибутов: alpha, axes, axis, dilations, ends, epsilon, group, keepdims, kernel\_shape, mode, momentum, op, output\_shape, pads, starts, steps, strides, value, perm [6]. Операция нейронной сети представляет собой функцию над входными данными, например, сложение выходов нескольких слоев, срез или конкатенация. Такие операции позволяют создавать более сложные и разнообразные архитектуры, находить наиболее эффективные структуры, в отличие от решений, где используются только слои сети.

Получившийся граф будет связным, ориентированным и ациклическим, иметь один исток — входной слой, а также один сток — выходной слой. Такой граф может не представлять собой простую структуру из связанных последовательно вершин, в связи с существованием пропускных соединений [7] (англ. skip-connections),

которые обеспечивают разветвление последовательности слоев, а также решают проблему исчезающего градиента.

Граф представляет собой достаточно сложную структуру, однако его можно представить в виде матрицы смежности либо списка смежности, то есть в виде набора векторов, где каждой вершине соответствует свой вектор с набором смежных вершин. Набором векторов является и матрица атрибутов: каждой вершине соответствует свой вектор атрибутов. Выбранное представление графа — в виде набора векторов — подходит больше, потому что для единого вектора нет способов восстановить граф целиком, а только предсказать, какие существующие подграфы наиболее вероятно принадлежат данному графу.

Такое векторное представление графа является достаточно большим, поэтому необходимо его сжать с помощью нейросетевых методов сжатия: автокодировщика, последовательность-в-последовательность, вариационного автокодировщика (VAE), вариационного автокодировщика для ориентированных ациклических графов (D-VAE) [2] и его модифицированной версии DVAE-EMB [3]. При обучении данные модели создают скрытое пространство, которое является эмбедингом нейронной сети, и в дальнейшем используется при восстановлении векторного представления графа с помощью декодировщика.

Рассмотрим **обратное преобразование векторного представления в нейронную сеть**. Имея модель, позволяющую сжимать векторное представление графа в скрытое пространство меньшей размерности, можно производить обратное преобразование из сжатого векторного представления в исходное. Обратное преобразование происходит аналогичным образом, что и прямое: для сжатых структуры графа и атрибутов (из скрытого пространства) получаем полную версию (из исходного пространства) с помощью декодирования в блоке декодировщика; объединяем данные для модели, таким же способом, которым данные были разделены; восстанавливаем матрицу смежности и атрибуты каждой.

Данные восстанавливаются из скрытого пространства меньшей размерности, а значит при восстановлении возникают потери информации, которые могут быть существенными, поэтому граф, полученный на выходе, может не соответствовать исходному графу. Поэтому необходимо дополнительно обработать полученный результат, чтобы структура графа была корректной.

На шаге **преобразования графа в нейросеть**, на вход поступает граф в виде матрицы смежности

Таблица 1. Точность моделей ResNet18 и AlexNet

Модель	Точность (%)	Корректность (%)
D-VAE	99,96	100
DVAE-EMB	99,99	100
Предложенная модель	99,99	100

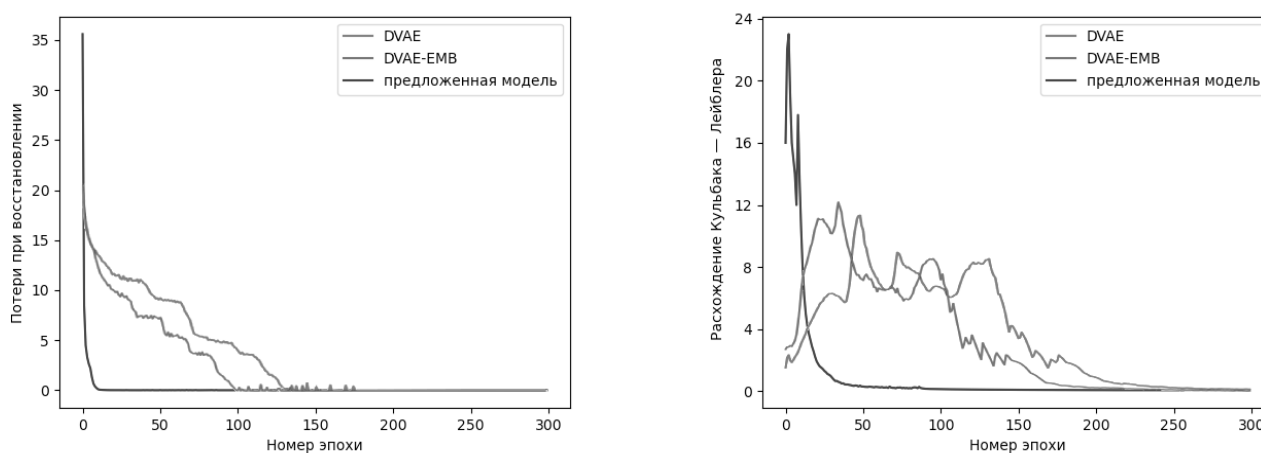


Рис. 1. Графики потерь при восстановлении (слева) и расхождения Кульбака — Лейблера (справа) во время обучения моделей DVAE, DVAE-EMB и предложенной модели

и атрибутов, затем для каждой вершины создается соответствующий ей слой или операция с атрибутами, образуя набор векторов, каждый вектор которого состоит из последовательно идущих друг за другом слоев. Одной из основных проблем восстановления является то, что при преобразовании данных в скрытое пространство происходит потеря части информации. Из-за этого велик риск того, что нейронная сеть, полученная итоговым способом, окажется неработоспособной. Поэтому требуется дополнительно обрабатывать полученную нейронную сеть с помощью правил.

### Результаты

Тестирование проводилось на наборе данных из 1500 архитектур сверточных нейронных сетей, которые классифицируют изображения, включающих в себя перечисленные выше операции и атрибуты. Для данного набора данных максимальное число вершин графа было 3000, а максимальное число ребер из одной вершины достигало 62.

Сравнение вариационного автокодировщика для ориентированных ациклических графов, его модификации и предложенной модели представлены на рисунке 1. Сравнение производится на наборе данных, состоящих из архитектур нейронных сетей, в каждой

по 8 вершин, а также каждая вершина содержит один из восьми типов слоев. Красным цветом обозначена модель D-VAE, DVAE-EMB обозначена зеленым, а синим обозначена предложенная модель. Для обучения в функции потерь применяется сжатие через сумму, а не через среднее значение. На рисунке 1 слева приведено сравнение потерь при восстановлении графа, справа — расхождение Кульбака-Лейблера [8].

Предложенная модель восстанавливает отдельно матрицу смежности и тип операции. Предложенное решение показывает результаты не хуже, а размер скрытого пространства меньше: 56 для существующих решений и 4 для списка смежности предложенной модели, а также 2 для типа операции. Однако предложенная модель не объединяет пространства атрибутов и структуры графа.

В таблице 1 приведены результаты сравнения предложенной модели и существующих решений. Точность обозначает процент правильно восстановленных архитектур, а корректность — процент работающих архитектур, полученных из скрытого пространства.

В таблице 2 приведено сравнение точности моделей ResNet18 и AlexNet после обучения в течение 50 эпох. Для ResNet18 видно, что модель со сжатием обучает-

Таблица 2. Точность моделей ResNet18 и AlexNet

ResNet18		AlexNet	
Модель	Точность (%)	Модель	Точность (%)
Без сжатия	78,99 ± 0,39	Без сжатия	70,46 ± 0,50
Со сжатием	79,17 ± 0,23	Со сжатием	60,44 ± 0,39
Оригинальная	78,99 ± 0,39	Оригинальная	71,54 ± 0,42

ся быстрее и имеет более высокую точность, однако это происходит из-за того, что оригинальная модель не обучилась до конца, это видно на рисунке 5 слева. Для AlexNet видно, что оригинальная модель имеет высокую точность, а остальные модели меньшую, так как при преобразовании нейронная сеть теряет часть информации, а при сжатии и восстановлении потеря информации становится больше.

## Выводы

В данной работе был предложен метод, позволяющий трансформировать нейронную сеть в векторное представление ее графа и обратно. Было предложено и исследовано несколько вариантов сжатия векторного представления графа в скрытое пространство. Модель для сжатия скрытого пространства использует не только тип операции, как в существующих решениях, но также и дополнительные параметры слоев и опе-

раций нейронной сети, позволяющие расширить возможности представления скрытого пространства.

Сравнение предложенного способа представления нейронной сети как графа показало, что при прямом и обратном преобразовании предложенным способом результаты работы нейронных сетей не хуже, чем у существующих методов. При этом предложенное решение позволяет достичь меньшую размерность скрытого пространства, используя структуру графа и кодируя атрибуты отдельно от вершин графа. Следует, однако, подчеркнуть, что из-за сжатия пространства и, как следствие, потери части информации, точность оригинальной модели остается несколько выше, чем в преобразованных.

В дальнейшем предложенный метод можно развивать за счет использования новых моделей векторизации графа, что может улучшить результаты по сравнению с полученными.

## ЛИТЕРАТУРА

1. Elsken T., Metzen J.H., Hutter F. Neural architecture search: A survey //The Journal of Machine Learning Research. — 2019. — Т. 20. — № 1. — С. 1997–2017.
2. Zhang M. et al. D-vae: A variational autoencoder for directed acyclic graphs //Advances in Neural Information Processing Systems. — 2019. — Т. 32.
3. Chatzianastasis M. et al. Graph-based Neural Architecture Search with Operation Embeddings //Proceedings of the IEEE/CVF International Conference on Computer Vision. — 2021. — С. 393–402.
4. Meng Z. et al. Co-embedding attributed networks //Proceedings of the twelfth ACM international conference on web search and data mining. — 2019. — С. 393–401.
5. Chen X. et al. Learning on attribute-missing graphs //IEEE transactions on pattern analysis and machine intelligence. — 2020.
6. Jin T. et al. Compiling ONNX neural network models using MLIR //arXiv preprint arXiv:2008.08272. — 2020.
7. Adaloglou N. Intuitive explanation of skip connections in deep learning //AI Summer. — 2020.
8. Kullback-Leibler Divergence Explained [Электронный ресурс]. — 2017. — URL: <https://www.countbayesie.com/blog/2017/5/9/kullback-leibler-divergence-explained>.

© Шаламов Вячеслав Владимирович ( sslavian812@yandex.ru ),

Ефимова Валерия Александровна ( valeryefimova@gmail.com ), Фильченков Андрей Александрович ( afillchenkov@itmo.ru ).

Журнал «Современная наука: актуальные проблемы теории и практики»