

ПОИСК АРХИТЕКТУРЫ НЕЙРОННОЙ СЕТИ

SEARCH FOR THE ARCHITECTURE OF A NEURAL NETWORK

V. Popukaylo
A. Shmelyova

Summary. The article describes the problem of introducing artificial neural networks with deep learning into enterprises. The method of automatic search of neural architecture is described, which allows optimizing and automating the process of developing deep learning models. The components of neural architecture search are considered, in particular: search space, search strategies and evaluation.

Keywords: neural networks, deep learning, architecture of neural systems, architecture search.

Глубокое обучение все чаще внедряется в большее количество отраслей. Однако разработка высокопроизводительных, готовых к эксплуатации глубоких нейронных сетей может быть сложной задачей, потому что требует передовых навыков и ресурсов.

Поиск нейронной архитектуры (Neural Architecture Search, NAS) позволяет оптимизировать и автоматизировать громоздкий процесс разработки модели глубокого обучения, а также быстро и эффективно создавать глубокие нейронные сети, адаптированные к конкретным производственным требованиям. В этой статье представлено полное введение в NAS, а также краткий обзор AutoML, частью которого является NAS [1].

Искусственные нейронные сети (ИНС) — представляют собой вычислительные системы, вдохновленные биологическими нейронными сетями, основана на наборе связанных единиц или узлов, называемых искусственными нейронами. Эти нейроны могут передавать и обрабатывать сигналы. ИНС построены в три слоя: входной, для получения данных, скрытый, для их обработки, и выходной, для получения результатов [2].

Глубокие нейронные сети (Deep Neural Networks, DNN) можно определить, как ИНС с дополнительной глубиной, включающие несколько скрытых слоев единиц, между входным и выходным слоями. DNN могут изучать более сложные шаблоны, извлекая признаки на разных уровнях.

Обучение DNN основано на алгоритме обратного распространения ошибки, который можно считать одной из самых фундаментальных основ нейронных сетей. DNN вычисляют параметры после каждого прохода. затем итеративно корректируют и уточняют их во время

Попукайло Владимир Сергеевич

Кандидат технических наук, доцент,

ПГУ им. Т.Г. Шевченко

vsp.science@gmail.com

Шмельёва Анастасия Владимировна

Аспирант, ПГУ им. Т.Г. Шевченко

avshmlva@gmail.com

Аннотация. В статье описана проблема внедрения в предприятия искусственных нейронных сетей с глубоким обучением. Описан метод автоматического поиска нейронной архитектуры, который позволяет оптимизировать и автоматизировать процесс разработки моделей глубокого обучения. Рассмотрены компоненты поиска нейронной архитектуры, в частности: пространство поиска, стратегии поиска и оценки.

Ключевые слова: Нейронные сети, глубокое обучение, архитектура нейронных систем, поиск архитектур.

обратного прохода, чтобы улучшить извлечение признаков входных данных.

DDN добились успеха в компьютерном зрении, обработке естественного языка, распознавании речи, беспилотных автомобилях и многом другом. Их высокая производительность, позволяет решать сложные задачи, что приводит к длительному времени вывода результата, огромным затратам на вычисления и большим требованиям к памяти, что является препятствием для успешной коммерциализации приложений, основанных на глубоком обучении.

Поиск нейронной архитектуры — это подраздел автоматизированного машинного обучения (AutoML), представляет собой всеобъемлющий термин, относящийся к процессу автоматизации различных задач, связанных с применением машинного обучения к реальным проблемам, начиная с набора необработанных данных и заканчивая развертыванием готовой к производству модели. На каждом этапе разработки традиционной модели машинного обучения есть компоненты, которые требуют много времени, ресурсов, сложны и итеративны.

Использование методов на основе искусственного интеллекта (ИИ) для автоматизации процесса разработки машинного обучения может значительно сократить время производства. Когда дело доходит до классических алгоритмов машинного обучения (например, случайных лесов и нейронных сетей), инструменты AutoML используются для упрощения выбора модели и выбора обучающих гиперпараметров.

Существует множество способов структурировать и модифицировать нейронную сеть. Для достижения

максимальной производительности необходимо учитывать различные элементы, включая типы слоев, операции и функции активации, а также обучающие данные и соображения по развертыванию (время выполнения, память, оборудование для логического вывода и его вычислительные ограничения).

Поиск наиболее подходящей архитектуры глубокого обучения — это процесс, который включает множество итераций проб и ошибок. Поиск нейронной архитектуры представляет собой альтернативу ручному проектированию DNN. Алгоритм выбора опирается на стратегию поиска, которая, в свою очередь, зависит от схемы объективной оценки. Популярная сверточная нейронная сеть EfficientNet является примером архитектуры, созданной NAS.

Компоненты поиска нейронной архитектуры

Поиск нейронной архитектуры состоит из трех основных строительных блоков, которые можно разделить на категории с точки зрения пространства поиска, стратегии/алгоритма поиска и стратегии оценки. Каждый из этих компонентов может использовать различные методы.

- Пространство поиска. Определение операций, используемых для проектирования DNN.
- Стратегия поиска. Оптимизация метрик в соответствии с подходом, используемым для исследования пространства поиска, необходима для стратегии поиска и оценки производительности.
- Стратегия оценки. Оценка производительности DNN до построения и обучения.

Определение области поиска

Пространство поиска NAS определяет, какой тип архитектуры может быть обнаружен алгоритмом NAS. Поиск определяется набором операций, определяющих общую структуру сети, тип блоков или блоков, определяющих уровни, а также допустимую связность между уровнями для создания архитектур.

Чем больше элементов имеет пространство поиска, тем более сложным и универсальным алгоритм становится. Но, естественно, по мере расширения пространства поиска растут и затраты на поиск лучшей архитектуры. Типы операций, используемых при определении пространства поиска, включают последовательные послойные операции, представление на основе ячеек, иерархическую структуру и многое другое.

Разработка стратегии поиска

Стратегия поиска определяет, как алгоритм NAS экспериментирует с различными нейронными сетями.

Из выборки кандидатов алгоритм оптимизирует показатели производительности дочерней модели. Существуют различные методы, которые оптимизируют стратегии поиска, чтобы процесс быстрее и стабильнее давал лучшие результаты. Типы поисковых алгоритмов включают случайный поиск, нейроэволюционные методы, байесовские подходы и обучение с подкреплением.

Нейроэволюционные методы работают так же хорошо, как и обучение с подкреплением, кроме того, эволюционные методы, как правило, имеют лучшую производительность и основываются на меньших моделях. В то время как более ранние методы NAS были основаны на дискретных пространствах поиска, непрерывная формулировка пространства поиска архитектуры представила дифференцируемые методы поиска, которые открыли путь для оптимизации на основе градиента [3].

Опишем некоторые методы поиска:

- Случайный поиск состоит из алгоритма NAS, который случайным образом выбирает архитектуру нейронной сети из пространства поиска. Это дорогостоящий процесс. Потому что вместо того, чтобы использовать более эффективный подход, он пробивается через пространство поиска. В зависимости от сложности пространства поиска на один поиск может потребоваться большое количество дней работы графического процессора.
- Обучение с подкреплением. Тип метода поиска, который обучает модели машинного обучения принимать последовательность решений. Модель ищет решение проблемы, на основе проб и ошибок и получает награды и штрафы. Это позволяет принимать решения в сложной среде и учиться выбирать конфигурации, которые создают лучшие нейронные сети для NAS.

Байесовская оптимизация

Байесовская оптимизация — это набор методов, которые часто используются для ускорения процесса поиска. Это популярный подход для оптимизации гиперпараметров. На высоком уровне он начинается со случайного выбора и оценки архитектур, а затем постепенно настраивает направление поиска на основе результатов оценки и собранной информации о производительности различных архитектур.

Установка стратегии оценки

Во время поиска в NAS алгоритм обучает, оценивает, проверяет и сравнивает производительность, прежде чем выбрать оптимальную нейронную сеть. Полное обучение каждой нейронной сети обычно требует много времени и высоких вычислительных ресурсов — тысячи дней использования графического процессора.

Чтобы снизить затраты на оценку моделей глубокого обучения, можно использовать несколько стратегий, в том числе:

- Производительность прокси-задачи.
- Оценка производительности с низкой точностью — ранний выход через несколько эпох, обучение на подмножестве данных, модели или данные с уменьшенным масштабом.
- Наследование и распределения веса.
- Экстраполяция кривой обучения.

Объединение поиска и оценки: одноразовый подход

Независимый поиск и оценка большого количества дочерних моделей обходится дорого. Чтобы решить эту проблему, группа методов NAS использует пространство поиска подархитектур, принадлежащих одной суперархитектуре, с обученными весами, которые являются общими для всех подмоделей. Однократные методы различаются в зависимости от обучения [1, 9].

Эффективный поиск нейронной архитектуры является примером однократного алгоритма, который обеспечивает ускорение поиска в 1000 раз по сравнению с предыдущими методами.

Было произведено сравнение методов NAS с небольшими наборами эталонных данных (например, Cifar-10, MNIST, OUI-Adience-Age или сокращенными версиями ImageNet). Время этапа поиска для всех методов NAS составляет два дня, как и время переобучения. Средние результаты сообщаются на основе трехкратного повторения, результаты эксперимента представлены на рисунке 1.

Повышение эффективности NAS, коммерчески успешно для пользователей, у которых нет больших вычислительных ресурсов.

Трансферное обучение

Трансферное обучение — это еще один подход AutoML, который повторно использует предварительно обученную модель, разработанную для одной задачи, в качестве отправной точки для работы над новой проблемой. Трансферное обучение популярно в глубоком обучении, поскольку изученные карты признаков можно использовать для обучения DNN с небольшим объемом данных [6].

Основная идея NAS заключается в том, что каждый набор данных и сопутствующее ему оборудование и производственная среда имеют особую и уникальную архитектуру, которая лучше всего работает с ним. В отличие от трансферного обучения, NAS обладает гибкостью и настройкой, что требует от ученых и разработчиков изучения и подготовки весов для новой архитектуры [7].

Преимущество NAS

Необходимость иметь возможность эффективно внедрять DNN растет, поскольку они все востребованы во многих отраслях. Использование DNN требует множества ресурсов и специалистов. NAS позволяет:

- Автоматизировать процесс выбора архитектуры и тестирования.
- Повысить точность и скорость логического вывода.
- Сократить итеративный и громоздкий процесс запуска моделей глубокого обучения в производство.

NAS	AutoKeras (%)	ENAS (macro) (%)	ENAS (micro) (%)	DARTS (%)	NAO-WS (%)
Fashion-MNIST	91.84	95.44	95.53	95.74	95.20
CIFAR-10	75.78	95.68	96.16	94.23	95.64
CIFAR-100	43.61	78.13	78.84	79.74	75.75
OUI-Adience-Age	63.20	80.34	78.55	76.83	72.96
ImageNet-10-1	61.80	77.07	79.80	80.48	77.20
ImageNet-10-2	37.20	58.13	56.47	60.53	61.20

Источник: Составлено автором на основании [9]

Рис. 1. Сравнение поиска нейронной архитектуры

Растущий интерес к глубокому обучению и AutoML ускорил разработку множества методов и алгоритмов для поиска в нейронной архитектуре. Использование NAS включает в себя разработку и тестирование.

Библиотеки с открытым исходным кодом для разработки:

- NASLib имеет общую модульную кодовую базу, которая облегчает исследование NAS и сравнение различных методов NAS. Библиотека также предоставляет интерфейсы для некоторых современных поисковых пространств и оптимизаторов NAS [8].
- Auto-PyTorch сочетает в себе идеи эффективной многоцелевой оптимизации, метаобучения и ансамбля для оптимизации нейронных архитектур и гиперпараметров, разработан для поддержки табличных данных [10].

Тесты NAS для повышения эффективности:

- NAS-Bench101 — это первый общедоступный набор данных по архитектуре для исследований в области NAS. Запрашивая предварительно вычисленный набор данных, исследователи могут оценить качество моделей за миллисекунды [5].

- Surrogate NAS для произвольных областей поиска направлены на преодоление ограничений табличных тестов NAS; поскольку он фокусируется на небольших архитектурных пространствах поиска, что приводит к результатам, которые не переносятся на большие пространства [4].

- NAS-Bench-Suite предлагает исчерпывающий набор эталонных тестов NAS для облегчения воспроизводимых, обобщающих и быстрых исследований.

Вывод

NAS предоставляет множество преимуществ для специалистов по данным, стремящихся оптимизировать DNN и сделать их более эффективными, но у них все еще есть ограничения. Методы NAS являются дорогостоящими в вычислительном отношении, поскольку для поиска лучшей модели может потребоваться много дней и огромная вычислительная мощность. Модели NAS обучаются на автономных данных, поэтому может быть трудно предсказать, как они будут работать на архитектурах с реальными данными.

ЛИТЕРАТУРА

1. Баррет З., Виджай В., Джонатон Ш. Изучение переносимых архитектур для масштабируемого распознавания изображений. 2017. [Электронный источник] / URL: <https://arxiv.org/abs/1707.07012>
2. Галушкин А.И. Нейронная сеть // Большая российская энциклопедия / под редакцией: Осипов. Ю.С. Москва.: Изд-во Большая российская энциклопедия, 2021.
3. Давидюк Ю.И. Методы нейроэволюции сетей прямого распространения // Вестник Брестского государственного технического университета. 2021. С. 49–53. [Электронный источник] / URL: <https://rep.bstu.by/bitstream/handle/data/21563/49-53.pdf?sequence=1>
4. Зела А, Симс Ж., Циммер Л., Лукасик Й., Койпер М., Хаттер Ф. Surrogate NAS выход за пределы ограниченного пространства поиска табличных тестов NAS. 2022. [Электронный источник] / URL: <https://arxiv.org/abs/2008.09777>
5. Ин К., Клейн А., Реал Э., Кристиансен Э., Мерфи К., Хаттер Ф. NAS-Bench-101: Поиск воспроизводимой нейронной архитектуры. 2019. [Электронный источник] / URL: <https://arxiv.org/abs/1902.09635>
6. Марголина А. Что такое transfer learning в обучении нейросетей. 2023. [Электронный источник] / URL: <https://sysblok.ru/glossary/chto-takoe-transfer-learning-v-obuchenii-nejrosetej/>
7. Синь Х., Кайён Ч. AutoML: обзор современного состояния. 2021 [Электронный источник] / URL: <https://arxiv.org/pdf/1908.00709.pdf>
8. Уайт К., Зела А., Рухте М., Кришнакумар А., Мехта Я. Библиотека NASLib. 2023. [Электронный источник] / URL: <https://arxiv.org/abs/2006.13799>
9. Хайфэн Д. Эффективный поиск нейронной архитектуры с помощью сетевого морфизма. 2018. [Электронный источник] / URL: https://www.researchgate.net/publication/326029461_Efficient_Neural_Architecture_Search_with_Network_Morphism
10. Циммер Л., Линдауэр М., Хаттер Ф. Auto-PyTorch Tabular: для эффективного и надежного AutoDL. 2020. [Электронный источник] / URL: <https://github.com/automl/auto-pytorch/>