

Обзор некоторых подходов к оптимизации нейронных сетей через эволюционные методы

Цветкова А.Ю., СПбГУ, Санкт-Петербург tsvetkova.anna.at6@gmail.com,
Михайлов Д.А., м.н.с. СПб ФИЦ РАН, Санкт-Петербург dam@dsdcs.pro

Аннотация

Оптимизация нейронных сетей остаётся одной из ключевых задач в машинном обучении. Помимо классических градиентных методов, активно развиваются альтернативные подходы, основанные на эволюционных принципах. Эволюционные алгоритмы позволяют эффективно подбирать веса нейросетей без использования производных, что делает их устойчивыми к шуму и локальным минимумам. В данной работе представлен обзор современных методов нейроэволюции, таких как NEAT, HyperNEAT, DeepNEAT, CoDeepNEAT и CoEGAN. Рассматриваются их принципы, архитектурные особенности и области применения.

Введение

Оптимизация нейронных сетей является активно развивающимся направлением в области машинного обучения: дополняются и совершенствуются существующие методы, разрабатываются новые. Наиболее широко используются классические градиентные методы, такие как стохастический градиентный спуск (SGD), метод импульсов (momentum), метод Нестерова, оптимизаторы Adam, AdaGrad, AdaDelta и другие [1]. Однако эти подходы обладают рядом ограничений: чувствительностью к шуму, склонностью к преждевременной сходимости, а также высокой зависимостью от настроек гиперпараметров. В качестве альтернативы были предложены эволюционные алгоритмы, которые осуществляют оптимизацию нейронных сетей без использования градиента. Такой подход менее подвержен локальным минимумам и обладает большей устойчивостью к нестабильным данным. Настоящая работа посвящена обзорному анализу методов, основанных на эволюционных принципах, и их применению в задачах построения и обучения нейросетей.

Нейроэволюция

Нейроэволюция [2] представляет собой метод обучения нейронных сетей, в основе которого лежат не классические методы оптимизации, такие как гра-

диентный спуск, а эволюционные алгоритмы. Нейроэволюционные алгоритмы можно условно разделить на две группы: алгоритмы, оптимизирующие и топологию сети и веса, и алгоритмы, которые работают только с оптимизацией весов. Например, более ранние вариации нейроэволюционных алгоритмов оптимизировали только веса, а топология сети оставалась фиксированной. Однако, структура сети сильно влияет на производительность, поэтому появилась потребность в методах, которые оптимизируют топологию. Структура нейронной сети определяет размер пространства поиска. Если топология сети слишком проста, есть риск не найти оптимальное решение. Методы, оптимизирующие и топологию и веса получили название топологические и весовые эволюционирующие искусственные нейронные сети (topology and weight evolving artificial neural network, TWEANN) [3].

NEAT

Нейроэволюционный алгоритм NEAT (NeuroEvolution of Augmenting Topologies) [2, 4, 5] был разработан в 2002 году Кеннетом Стэнли и Ристо Миккулайненом. NEAT относится к тем алгоритмам, которые способны оптимизировать и топологию и веса нейронных сетей. Метод позволяет решать проблемы, которые возникают из-за особенностей нейроэволюции: благодаря историческим меткам NEAT обеспечивает корректную работу кроссинговера нейронных сетей с разной топологией, а механизм видообразования препятствует преждевременному вымиранию хороших вариантов конфигураций сетей. Важно также отметить, что NEAT начинает работать с простыми сетями, постепенно добавляя новые узлы и связи. Более подробно стоит разобрать каждую из особенностей алгоритма.

Историческая маркировка используется для решения проблемы конкурирующих конвенций - одной из проблем нейроэволюции. Так как структуры развиваются независимо друг от друга в разных сетях, существует более одного способа представления нейронных сетей. Потомство после скрещивания таких сетей может потерять функциональность. Во избежание такой ситуации, система должна уметь определять идентичные структуры. NEAT делает это при помощи исторических меток: каждому новому гену, добавленному в геном, присваивается уникальное инновационное число, которое увеличивается с каждой новой мутацией. Маркировки облегчают кроссинговер, идентифицируя гомологичные участки между различными сетями. Инновационное число каждого гена наследуется потомством, что делает возможным отслеживание эволюции сети [5].

Видообразованием называется процесс, при котором похожие сети груп-

пируются вместе и конкурируют только между собой, а не со всей популяцией. Это дает время новым сетям оптимизировать веса и улучшить производительность, прежде чем они будут сравнены с другими сетями. В NEAT сети группируются по видам на основе структуры их соединений. Топологическое сходство определяется с помощью выравнивания геномов на основе исторических меток и определения совпадающих, непересекающихся и избыточных генов среди индивидуумов [5].

- Совпадающие гены - это гены, которые одинаковые у двух сетей.
- Непересекающиеся гены - это гены, которые различаются между двумя сетями и расположены в середине генома.
- Избыточные гены - это гены, которые различаются между двумя сетями и расположены в конце генома.

Топологическое сходство вычисляется с помощью расстояния совместимости [5, 6]:

$$\delta = \frac{c_1 E}{N} + \frac{c_2 D}{N} + c_3 W$$

где E — число избыточных генов, D — число непересекающихся генов, W — средняя разница в весе совпадающих генов, N — число генов в большем геноме, а c_1 , c_2 , c_3 — коэффициенты, определяющие важность трех факторов. На практике N принимается равным 1.

Более подробно рассмотрим процесс эволюции нейронной сети. NEAT начинает постепенно наращивать сложность сети, начиная с полностью связанных сетей без скрытых узлов. Однако нейронная сеть модифицируется только в том случае, если добавление новых узлов и связей увеличит производительность сети. NEAT использует прямое кодирование: между фенотипом и генотипом существует взаимно-однозначное соответствие. Генотип состоит из двух типов генов: одни используются для кодирования узлов, другие - для кодирования связей. Гены узлов имеют поля, указывающие на тип узла: входной, выходной, скрытый. Гены связи содержат информацию об идентификаторах узлов, между которыми расположена связь, о весе соединения, а также о номере инновации, который позволяет найти соответствующие гены во время скрещивания.

HyperNEAT

HyperNEAT (Hypercube-based NeuroEvolution of Augmenting Topologies) [3] - одна из модификаций классического алгоритма NEAT, представленная

Кеннетом Стэнли в 2007 году. Особенностью метода HyperNEAT является проекция нейронной сети в гиперкубическое пространство, в котором каждому нейрону присваиваются координаты. Эти координаты подаются на вход в функцию CPPN (Compositional Pattern Producing Networks). CPPN реализует косвенное кодирование, которое не требует типичной стадии пошагового построения сети, в отличие от метода NEAT, в котором нейронная сеть постепенно эволюционирует.

Гиперкубическое пространство - это многомерное пространство, где каждая точка соответствует конкретной конфигурации нейронной сети. HyperNEAT хранит шаблон связей, где каждая пара кодирует соединение между двумя узлами и вычисляет четырехмерную функцию $S = F(x_1, y_1, x_2, y_2)$, где (x_1, y_1) - координаты исходного узла, а (x_2, y_2) - координаты целевого узла [6].

Расстояние между узлами сети определяется с помощью линейной комбинации количества избыточных генов (I), непересекающихся генов (E), разницы в весе совпадающих генов W [5, 6]:

$$D = k_1 \cdot \frac{I}{Q} + k_2 \cdot \frac{E}{Q} + k_3 \cdot W$$

Благодаря HyperNEAT появилась возможность компактно кодировать большие нейронные сети маленькими геномами и изменять размер эволюционировавших нейронных сетей после завершения обучения.

DeepNEAT и CoDeepNEAT

Модель NEAT также была расширена для работы с большими пространствами поиска, такими как глубокие нейронные сети, такой расширенный метод - DeepNEAT и его улучшенная версия - CoDeepNEAT [7, 8].

DeepNEAT - это расширение NEAT, которое рассматривает целые слои как гены, а не отдельные нейроны. Основное внимание уделяется определению составов слоев, а не выбору нейронов. Таким образом, становится удобнее работать с более крупными и глубокими сетями.

DeepNEAT работает по стандартному генетическому алгоритму: создает начальную популяцию особей, каждая из которых представлена графом, и развивает их на протяжении поколений. В течение этих поколений индивидуумы мутируют путем добавления или удаления узлов и ребер из своих графов, изменения фиксируются при помощи исторической маркировки (аналогично NEAT). Хромосомы сравниваются в каждом поколении метрикой сходства, затем классифицируются по субпопуляциям - видам. Каждый

вид оценивается по общей приспособленности его особей, которая рассчитывается при помощи функции распределения приспособленности. Выжившие виды развиваются отдельно друг от друга посредством кроссинговера между особями одного вида, таким образом формируется следующее поколение.

В силу того, что узлы теперь представлены слоями, а не нейронами, необходимо хранить дополнительную информацию о слоях: тип (сверточный, плотный, рекуррентный), свойство (количество нейронов, размер ядра, размер шага, функция активации), тип соединения узлов. Для этого используют таблицу возможных гиперпараметров (хромосомную карту) для каждого узла и еще одну таблицу - таблицу глобальных параметров, применимых ко всей сети (скорость обучения, алгоритм обучения и предварительная обработка данных) [7].

CoDeepNEAT расширяет метод DeepNEAT. Он разделяет построение топологии на два разных уровня: *module chromosomes* и *blueprint chromosomes*.

- *Module Chromosomes* или модульные хромосомы - это графы, которые описывают небольшую структуру связанных слоев. Такие графы называют модулями
- *Blueprint Chromosomes* или хромосомы-чертежи - это графы, которые описывают, как модули соединяются между собой. На основе таких графов строится нейросеть, где в качестве узлов используются эти модули.

Таким образом, CoDeepNEAT работает с каждым видом хромосом по отдельности, а потом компонует их в сети.

В CoDeepNEAT эволюция осуществляется независимо для каждой популяции хромосом и требует индивидуальной оценки качества особей. Отличие генетического алгоритма в CoDeepNEAT от генетического алгоритма DeepNEAT состоит в том, что оценка присваивается и *blueprint*-хромосомам и модулям, которые в них используются. Затем эта оценка распределяется между соответствующими видами [7].

COEGAN

COEGAN является адаптацией алгоритма DeepNEAT для генеративно-сопоставительной сети (Generative adversarial network, GAN) [9]. В этой модели сочетаются нейроэволюционные и коэволюционные подходы для обучения. Модель создана для обеспечения более стабильного метода обучения и автоматического проектирования архитектур нейронных сетей.

GAN активно применяется в компьютерном зрении. В системе GAN одна из сетей генерирует образцы (генератор), а другая старается отличить правильные образцы от неправильных (дискриминатор). Дискриминатор и генератор являются глубокими нейронными сетями с заранее определенной архитектурой: структура сетей и гиперпараметры выбираются экспериментально, поэтому временные затраты увеличиваются. Генеративная сеть пытается создать новый образец, смешав несколько исходных образцов. Дискриминативная сеть обучается различать подлинные и поддельные образцы. Таким образом целью сети-генератора является повышение процента ошибок сети-дискриминатора, а целью сети-дискриминатора является улучшение точности распознавания. Существуют подходы, которые могут автоматизировать проектирование архитектур таких нейронных сетей.

Геном модели COEGAN может быть представлен в виде массива генов, где ген - описание слоя нейросети. Рассматриваются три типа слоев: линейные, сверточные и транспонированные сверточные. Такая последовательность генов преобразуется в архитектуру глубокой нейронной сети. Каждому гену сопоставляется случайным образом выбранная функция активации: ReLU, LeakyReLU, ELU, Sigmoid и Tanh [9]. Случайным параметром линейного слоя является количество входных параметров, а сверточного и транспонированного сверточного слоя - количество выходных каналов. Таким образом, гены имеют только функцию активации, выходные признаки и выходные каналы, подверженные мутации.

В COEGAN популяция состоит из двух субпопуляций генераторов G_i и дискриминаторов D_i . Внутри каждой субпопуляции применяется механизм видообразования, используемый в NEAT, чтобы способствовать инновациям в каждой субпопуляции. В COEGAN оптимизируется только топология нейронной сети, а веса оптимизируются при помощи градиентного спуска. Это обусловлено тем, что количество внутренних параметров нейронной сети (в частности, к ним относятся и веса) достаточно велико и эволюция такой сети приведет к сильному увеличению вычислительной сложности [9].

Заключение

В данной работе были исследованы несколько методов оптимизации нейронных сетей, основанных на эволюционных принципах. Среди них методы NEAT, HyperNEAT, DeepNEAT, CoDeepNEAT и метод COEGAN. Одним из первых эволюционных методов оптимизации был NEAT, оптимизирующий топологию и веса нейронных сетей, на основе которого впоследствии были разработаны некоторые другие методы. К таким расширениям NEAT отно-

сится, например, HyperNEAT, способный работать с более сложными сетями и использующий вместо прямой эволюции метод косвенного кодирования. Другими вариациями NEAT стали алгоритмы DeepNEAT, CoDeepNEAT, способные работать с глубокими нейронными сетями. Отдельно был рассмотрен метод COEGAN, оптимизирующий генеративно-состязательные сети, используя сочетание коэволюционных и нейроэволюционных подходов.

Список литературы

- [1] Kingma D. P., Ba J. Adam: A method for stochastic optimization // arXiv preprint arXiv:1412.6980. – 2014.
- [2] Stanley K. O., Miikkulainen R. Evolving neural networks through augmenting topologies // *Evolutionary Computation*. – 2002. – Т. 10. – № 2. – С. 99–127.
- [3] D'Ambrosio D. B., Gauci J., Stanley K. O. HyperNEAT: The first five years // *Growing Adaptive Machines: Combining Development and Learning in Artificial Neural Networks*. – 2014. – С. 159–185.
- [4] Ibrahim M. Y., Babu G. S., Manogaran G., Chilamkurti N. Advances in neuroevolution through augmenting topologies – a case study // *2019 11th International Conference on Advanced Computing (ICoAC)*. – IEEE, 2019. – С. 111–116.
- [5] Stanley K. O., Miikkulainen R. Efficient evolution of neural networks through complexification. – Texas: The University of Texas at Austin, 2004.
- [6] Zhukabayeva T., Omarov B., Yessengaliyev A., Seidakhmetov B., Rakhmetov A., Fattakhov R. Network attack detection using neuroevolution of augmenting topologies (NEAT) algorithm // *JOIV: International Journal on Informatics Visualization*. – 2024. – Т. 8. – № 1. – С. 387–394.
- [7] Bohrer J. S., Grisci B. I., Dorn M. Neuroevolution of neural network architectures using CoDeepNEAT and Keras // arXiv preprint arXiv:2002.04634. – 2020.
- [8] Miikkulainen R., Liang J., Meyerson E., Rawal A., Fink D., Francon O., Raju B., Shahrzad H., Navruzyan A., Duffy N., Hodjat B. Evolving deep neural networks // *Artificial Intelligence in the Age of Neural Networks and Brain Computing*. – Academic Press, 2024. – С. 269–287.

- [9] Costa V., Oliveira L. S., Bittencourt G. COEGAN: evaluating the coevolution effect in generative adversarial networks // Proceedings of the Genetic and Evolutionary Computation Conference. – 2019. – С. 374–382.