

Дифференцируемый алгоритм поиска архитектуры модели с контролем её сложности

К. Д. Яковлев, О. С. Гребенькова, О. Ю. Бакhteев

iakovlev.kd@phystech.edu; grebenkova.os@phystech.edu; bakhtelev@phystech.edu

В работе исследуется задача построения модели глубокого обучения. Предлагается метод поиска архитектуры модели, позволяющий контролировать её сложность с небольшими вычислительными затратами. Под сложностью модели понимается минимальная длина описания, минимальное количество информации, которое требуется для передачи информации о модели и выборке. В основе метода лежит дифференцируемый алгоритм поиска архитектуры модели (DARTS). Предлагается использовать гиперсеть в качестве функции релаксации. Под гиперсетью понимается модель, генерирующая параметры другой модели. Предложенный метод позволяет контролировать сложность модели в процессе поиска архитектуры. Для оценки качества предлагаемого алгоритма проводятся эксперименты на выборках MNIST и CIFAR-10.

Ключевые слова: дифференцируемый алгоритм поиска архитектуры; глубокое обучение; гиперсети; нейронные сети; контроль сложности модели

1 Введение

В последнее время растет интерес к разработке алгоритмических решений для автоматизации процесса проектирования архитектуры. Лучшие существующие алгоритмы поиска архитектуры требуют больших вычислительных затрат, несмотря на их высокое качество.

В данной работе рассматривается задача поиска архитектуры модели глубокого обучения с контролем её сложности. В качестве базового алгоритма используется дифференцируемый алгоритм поиска архитектуры (DARTS) [1]. Данный метод решает задачу поиска архитектуры модели путем перевода пространства поиска из дискретного в непрерывное представление. В связи с этим появляется возможность использовать градиентные методы оптимизации, позволяющие использовать меньше вычислительных ресурсов, чем методы, работающие на дискретном множестве. Данный алгоритм универсален для работы как со сверточными, так и с рекуррентными нейронными сетями.

В работе [2] стабильность алгоритма DARTS была поставлена под сомнение. Одним из источников нестабильности является этап получения фактической дискретной архитектуры из архитектуры непрерывной смеси. На этом этапе часто наблюдается снижение качества модели. Это связано с тем, что алгоритм сходится в узкий регион, поэтому небольшие возмущения архитектуры ведут к значительному понижению качества на валидационной выборке.

В работе [3] было замечено, что операция *softmax* обладает существенным недостатком. Оказывается, что пропуск соединений постепенно становится доминирующим в процессе оптимизации. Вес данного соединения увеличивается гораздо быстрее, чем у его конкурентов. В связи с этим предлагается использовать сигмоидную функцию потерь. Таким образом, каждая операция может давать существенный вклад независимо от других. В нашей работе предлагается в качестве функции релаксации использовать гиперсеть [6]. Подход заключается в использовании небольшой сети для генерации весов операций. Предлагаются альтернативные подходы к решению задачи поиска архитектуры модели. В работе [4] формулируется задача обучения распределению. Веса смешанной

операции подчинены распределению Дирихле, так как они определены на вероятностном симплексе. Таким образом, задача поиска архитектуры сводится к поиску параметров распределения Дирихле.

В работе [5] строится алгоритм поиска нейронной архитектуры с ограниченным ресурсом (RC-DARTS). К базовому алгоритму DARTS добавляются ресурсные ограничения, такие как размер модели и вычислительная сложность. Для решения задачи условной оптимизации вводится алгоритм итерационной проекции, поскольку функции ограничений невыпуклые. В нашей работе для контроля сложности используется гиперсеть.

Вычислительный эксперимент проводится на выборках MNIST [7] и CIFAR-10 [8].

2 Постановка задачи

2.1 Дифференцируемый алгоритм поиска архитектуры

Поставим задачу поиска архитектуры ячейки. Пусть внутри ячейки есть N занумерованных узлов, представленных в виде ориентированного ациклического графа. Каждому ребру (i, j) поставлена в соответствие операция $o^{(i,j)} \in \mathcal{O}$, где \mathcal{O} – множество операций. Значения в каждом из промежуточных узлов $x^{(j)}$ определяются через значения в узлах с меньшим номером:

$$x^{(j)} = \sum_{i < j} o^{(i,j)}(x^{(i)}) \quad (1)$$

Таким образом, задача поиска архитектуры сводится к задаче выбора операций между узлами ячейки. Для того, чтобы свести задачу дискретной оптимизации к задаче непрерывной оптимизации, введем смешанную операцию для каждого ребра (i, j) :

$$\hat{o}^{(i,j)}(x) = \sum_{o \in \mathcal{O}} \frac{\exp(\alpha_o^{(i,j)})}{\sum_{o' \in \mathcal{O}} \exp(\alpha_{o'}^{(i,j)})} o(x), \quad (2)$$

где $\alpha_o^{(i,j)}$ обозначает соответствующий вес операции o на ребре (i, j) . Таким образом, каждому ребру (i, j) ставится в соответствие вектор $\alpha^{(i,j)}$ размерности $|\mathcal{O}|$. Пусть $\alpha = [\alpha^{(i,j)}]$. Сформулируем двухуровневую задачу оптимизации:

$$\begin{aligned} \min_{\alpha} \mathcal{L}_{val}(\mathbf{w}^*(\alpha), \alpha), \\ \text{s.t. } \mathbf{w}^* = \arg \min_{\mathbf{w}} \mathcal{L}_{train}(\mathbf{w}, \alpha) \end{aligned} \quad (3)$$

Здесь \mathcal{L}_{val} и \mathcal{L}_{train} функции потерь модели на валидации и на обучении соответственно.

Алгоритм 1 DARTS – Differentiable Architecture Search

- 1: Для каждого узла создадим смешанную операцию $\hat{o}^{(i,j)}$, параметризованную $\alpha^{(i,j)}$
 - 2: **пока** алгоритм не сошелся
 - 3: обновим α , сделав градиентный шаг вдоль $\nabla_{\alpha} \mathcal{L}_{val}(\mathbf{w} - \xi \nabla_{\mathbf{w}} \mathcal{L}_{train}(\mathbf{w}, \alpha), \alpha)$
 - 4: обновим веса \mathbf{w} , делая градиентный шаг вдоль $\nabla_{\mathbf{w}} \mathcal{L}_{train}(\mathbf{w}, \alpha)$
 - 5: получить окончательную архитектуру из полученного в результате алгоритма α
-

Дискретная архитектура получается следующим образом:

$$o^{(i,j)} = \arg \max_{o \in \mathcal{O}} \alpha_o^{(i,j)}$$

2.2 Линейная гиперсеть

Пусть Λ – множество параметров, контролирующие сложность модели. Под гиперсетью мы будем понимать следующее отображение:

$$\mathbf{G} : \Lambda \times \mathbb{U} \rightarrow \mathbb{A}, \quad (4)$$

где \mathbb{A} – пространство весов операций $\alpha_o^{(i,j)}$, а \mathbb{U} – множество параметров гиперсети. В данной работе для получения весов операций используется линейная гиперсеть:

$$\mathbf{G}_{linear}(\lambda) = \lambda \mathbf{b}_1 + \mathbf{b}_2, \quad (5)$$

где $\mathbf{b}_1, \mathbf{b}_2$ – обучаемые параметры гиперсети.

2.3 Алгоритм DARTS с контролем сложности

В качестве базового алгоритма используется алгоритм 1. Одним из основных отличий является то, что смешанная операция записывается следующим образом:

$$\hat{o}^{(i,j)} = \sum_{o \in \mathcal{O}} \alpha_o^{(i,j)} o(x)$$

Кроме того, сложность модели определяется параметром λ гиперсети. Остальные шаги алгоритма остаются без изменений.

Литература

- [1] Liu Hanxiao, Simonyan Karen, Yang Yiming. Darts: Differentiable architecture search // CoRR, 2018. Vol. abs/1806.09055. URL: <http://arxiv.org/abs/1806.09055>.
- [2] Chen Xiangning, Hsieh Cho-Jui. Stabilizing differentiable architecture search via perturbation-based regularization // CoRR, 2020. Vol. abs/2002.05283.
- [3] Chu Xiangxiang, Zhou Tianbao, 0046 Bo Zhang, Li Jixiang. Fair darts: Eliminating unfair advantages in differentiable architecture search // CoRR, 2019. Vol. abs/1911.12126. URL: <http://arxiv.org/abs/1911.12126>.
- [4] Chen Xiangning, Wang Ruochen, Cheng Minhao, Tang Xiaocheng, Hsieh Cho-Jui. Drnas: Dirichlet neural architecture search // CoRR, 2020. Vol. abs/2006.10355.
- [5] Jin Xiaojie, Wang Jiang, Slocum Joshua, 0001 Ming-Hsuan Yang, Dai Shengyang et al. Rc-darts: Resource constrained differentiable architecture search // CoRR, 2019. Vol. abs/1912.12814. URL: <http://arxiv.org/abs/1912.12814>.
- [6] Ha David, Dai Andrew M., Le Quoc V. Hypernetworks // CoRR, 2016. Vol. abs/1609.09106. URL: <http://arxiv.org/abs/1609.09106>.
- [7] LeCun Yann, Cortes Corinna. MNIST handwritten digit database, 2010. URL: <http://yann.lecun.com/exdb/mnist/>.
- [8] Krizhevsky Alex, Nair Vinod, Hinton Geoffrey. Cifar-10 (canadian institute for advanced research). URL: <http://www.cs.toronto.edu/~kriz/cifar.html>.

Received February 25, 2021