Автоматическое построение нейросети оптимальной сложности

Горян[1] ***Н. А. Бахтеев***[1] ***О. Ю. Стрижов***[2] ***В. В.***
goryan.na@phystech.edu; bakhteev@phystech.edu; strijov@phystech.edu
[1]Московский физико-технический институт
[2]Вычислительный центр им. А. А. Дородницына ФИЦ ИУ РАН

Работа посвящена выбору оптимальной модели нейронной сети. Нейронная сеть рассматривается как вычислительный граф, рёбра которого — примитивные функции, а вершины — промежуточные представления выборки. Предполагается, что структуру нейронной сети можно упростить без значимой потери качества классификации. Структура нейросети опеределяется вершинами симплекса. Для определения нужной структуры нейронной сети предлагается проводить оптимизацию гиперпараметров и структурных параметров. Для решения задачи оптимизации предлагается проводить релаксацию структуры. Для анализа качества представленного алгоритма проводятся эксперименты на выборках Boston, MNIST и CIFAR-10.

Ключевые слова: *нейронные сети, оптимизация гиперпараметров, прореживание нейронной сети, оптимальная структура нейронной сети, вариационный вывод.*

Введение

В данной работе рассматривается алгоритм построения оптимальной нейронной сети. Одной из основных областей применения являются мобильные устройства, которые в силу своих ограниченных вычислительных ресурсов не могут справляться с избыточно сложными неросетями [1]. Существует ряд способов построения нейронных сетей. Для того, чтобы подобрать нужную сеть требуется определить оптимальные значения структурных параметров [2]: количество слоёв, нейронов в каждом слое и функции активации каждого нейрона. Выбор этих гиперпараметров является вычислительно сложной задачей [3].

В работах [4, 5] используется алгоритм прореживания нейросети. Он заключается в построении заведомо переусложнённой модели, которая в последствии упрощается. Ещё одиним способом, предложенным в работе [6], является метаобучение, которое получая на вход некоторую выборку возвращает оптимальные гиперпараметры. В данной работе исследуется алгоритм, который оптимизирует параметры, гиперпараметры, структурные параметры нейросети в единой процедуре. В основе лежит алгоритм DARTS, предложенным в работе [7], в основе которого лежит процедура релаксации: переход от дискретного множества структурных параметров к непрерывному, что позволяет использовать методы градиентной оптимизации для нахождения лучших гиперпараметров. Входными данными алгоритма являются некоторый набор данных и заранее определённый набор функций активации. Как результат мы получаем оптимальную нейросеть.

Проверка и анализ метода проводится на выборке Boston Housing [8], MNIST [9], CIFAR-10 [10] и синтетических данных. Полученная модель сравнивается с моделями, полученными при помощи базовых алгоритмов.

Постановка задачи

Пусть заданы обучающая и валидационная выборки

$$\mathfrak{D}^{train} = \{\mathbf{x}_i, y_i\},\ i = 1, ..., m^{train}, \qquad (?)$$

$$\mathfrak{D}^{valid} = \{\mathbf{x}_i, y_i\},\ i = 1, ..., m^{valid}, \tag{?}$$

где $\mathbf{x}_i \in \mathbb{R}^n$ — объекты, а $y_i \in \{1, ..., Z\}$ — метки объектов $\mathbf{x}_i$, где Z — количество классов. Модель описывается ориентированным графом (V, E), для каждого ребра которого $(j, k) \in E$ определён вектор базовых функций $\mathbf{g}_{j,k}$ мощностью $K_{j,k}$. Модель $\mathbf{f}(\mathbf{x}, \mathbf{W})$ задаётся параметрами подмоделей $\{f_v\}_{v=1}^{|V|}$ и структурными параметрами γ. Каждая подмодель f_v описывается через графовое представление модели:

$$f_v(\mathbf{x}) = \sum_{k \in Adj(v_i)} < \gamma_{j,k}, g_{j,k} > f_k(\mathbf{x}, \mathbf{w}), \quad f_0(\mathbf{x}) = \mathbf{x}.$$

Параметры модели $\mathbf{W}$ — конкатенация параметров всех подмоделей $\{f_v\}_{v=1}^{|V|}$.
Структура модели $\mathbf{\Gamma}$ — конкатенация структурных параметров γ.

Функции потерь на обучении L и валидации Q задаются:

$$L = \log p(\mathfrak{D}^{train}, \mathbf{W}, \mathbf{\Gamma}) + e^{\mathbf{A}}||\mathbf{W}||,$$

$$Q = \log p(\mathfrak{D}^{valid}, \mathbf{W}, \mathbf{\Gamma}).$$

Таким образом, получили задачу двухуровневой оптимизации:

$$\mathbf{W}^* = \arg\min_{\mathbf{W}} L(\mathfrak{D}^{train}, \mathbf{W}, \mathbf{\Gamma}, \mathbf{A})$$

$$\mathbf{\Gamma}^*, \mathbf{A}^* = \arg\min_{\mathbf{\Gamma}, \mathbf{A}} Q(\mathfrak{D}^{valid}, \mathbf{W}, \mathbf{\Gamma}, \mathbf{A}).$$

Релаксация модели

Пусть у нас есть множество кандидатов на роль базовой функции мощности $\mathbb{K}$. Каждая базовая функция применима к нейрону x^i сети. Для того чтобы перейти от дискретной задачи поиска оптимальной базовой функции к непрерывной, производим релаксацию структуры модели используя softmax:

$$\overline{g}^{(i,j)}(x) = \sum_{g \in \mathbb{K}} \frac{exp(\gamma_g^{(i,j)})}{\sum\limits_{\overline{g} \in \mathbb{K}} exp(\gamma_{\overline{g}}^{(i,j)})} g(x),$$

где $\gamma^{(i,j)}$ — вектор размерности $|\mathbb{K}|$, парамметризующий комбинацию базовых функций. Таким образом мы перешли к задаче поиска базовой функции, подбирая непрерывные параметры γ. В конце поиска, каждая комбинация базовых функций $\overline{g}^{(i,j)}(x)$ меняется на $g^{(i,j)} = \arg\max\limits_{g \in \mathbb{K}} \gamma_g^{(i,j)}$.

Регуляризация структуры модели

Регуляризация структуры проводится добавлением к фунции потерь Q слагаемого $\lambda P(\Gamma)$, где $P(\Gamma)$ есть произведение всех вероятностей возникновения веса $\gamma^{(i,j)}$.
В качестве вероятности для структуры можно использовать Gumble-Softmax или распределение Дирихле.

Оптимизация гиперпараметров и структурных параметров модели

Потери на валидационной и обучающей выборке обусловлены как структурой модели Γ

так и параметрами модели W. Цель поиска архитектуры найти модель Γ^*, которая минимизирует ошибку на валидационной выборке $Q(w^*, \Gamma^*)$. При этом w^* находится из условия минимизации функции потерь $L(w, \Gamma^*)$.
Таким образом получается задача двухуровневой оптимизации:

$$\min_{\Gamma} \quad L(w^*(\Gamma), \Gamma)$$

$$s.t. \quad w^*(\Gamma) = \arg\min_{w} Q(w, \Gamma)$$

.
Чтобы решить эту задачу мы используем итеративную оптимизационную процедуру, в которой Γ и w оптимизируются по очереди с помощью градиентного спуска. На k-м шаге, имея структуру модели Γ_{k-1}, полуачем w_k изменяя w_{k-1} в сторону минимизации $L(w_{k-1}, \Gamma_{k-1})$. Далее, фиксируя w_k, находим Γ_k, минимизируя $Q(w_{k-1} - \xi \nabla_w L(w_{k-1}, \Gamma_{k-1}), \Gamma_{k-1})$, где ξ шаг градиентного спуска.

Литература

[1] S Rallapalli, H Qiu, AJ Bency, S Karthikeyan, R Govindan, BS Manjunath, and R Urgaonkar. Are very deep neural networks feasible on mobile devices? 2016.

[2] In Jae Myung and Mark A. Pitt. Applying occam's razor in modeling cognition: A bayesian approach. *Psychonomic Bulletin & Review*, 4(1):79–95, Mar 1997.

[3] I. Sutskever, O. Vinyals, and Q. Le. Sequence to sequence learning with neural networks. In *Advances in Neural Information Processing Systems 27*, volume 2 of *28th Annual Conference on Neural Information Processing Systems*, pages 3104–3112, Montreal, Quebec, Canada, 08–13 December 2014. Curran Associates, Inc.

[4] Y. LeCun, J. Denker, and S. Solla. Optimal brain damage. In *Advances in Neural Information Processing Systems 2*, pages 598–605, Denver, Colorado, USA, 27–30 November 1989. Morgan-Kaufmann.

[5] A. Graves. Practical variational inference for neural networks. In *Advances in Neural Information Processing Systems 24*, 25th Annual Conference on Neural Information Processing Systems, pages 2348–2356, Granada, Spain, 12–14 December 2011. Curran Associates, Inc.

[6] Dougal Maclaurin, David Duvenaud, and Ryan P. Adams. Gradient-based hyperparameter optimization through reversible learning. In *Proceedings of the 32Nd International Conference on International Conference on Machine Learning - Volume 37*, ICML'15, pages 2113–2122. JMLR.org, 2015.

[7] Hanxiao Liu, Karen Simonyan, and Yiming Yang. Darts: Differentiable architecture search, 2018. cite arxiv:1806.09055.

[8] D. Harrison and D. Rubinfeld. Hedonic prices and the demand for clean air, 1978. `https://www.cs.toronto.edu/~delve/data/boston/bostonDetail.html`.

[9] Y. LeCun, C. Cortes, and C. J.C. Burges. The mnist datadase of handwritten digits, 1998. `http://yann.lecun.com/exdb/mnist/`.

[10] A. Krizhevsky, V. Nair, and G. Hinton. The cifar-10 dataset, 2009. `http://www.cs.toronto.edu/~kriz/cifar.html`.