

# Автоматическое построение нейросети оптимальной сложности

Губанов<sup>1</sup> С.Е.

sergey.gubanov@phystech.edu

<sup>1</sup>Московский физико-технический институт

Работа посвящена оптимизации структуры нейронной сети. Оптимизация нейронной сети предполагает заданную структуру и значения гиперпараметров. Подобная оптимизация приводит к чрезмерному количеству параметров и неоптимальности структуры, что приводит к невысокой скорости оптимизации и переобучению. В данной работе предлагается новый метод оптимизации, который позволяет учитывать особенности задачи, подстраивая структуру и гиперпараметры в процессе оптимизации. Результатом работы предложенного метода является устойчивая модель, дающая приемлемое качество результатов при меньшей вычислительной сложности.

**Ключевые слова:** *нейронные сети, оптимизация гиперпараметров, вычислительный граф, прореживание нейронной сети, устойчивость.*

## 1 Введение

Современные глубокие нейронные сети являются вычислительно емкими моделями и содержат сотни миллионов параметров [1]. Это обуславливает не только длительное время оптимизации, но и ресурсоемкость эксплуатации. Переусложненная модель требует много ресурсов и затрудняет использование в переносимых устройствах и микроконтроллерах. Также существует риск переобучения из-за чрезмерного числа параметров [2]. Целью данной работы является алгоритм построения нейросети, чтобы эти проблемы, а также проблема устойчивости модели, были учтены.

Идея автоматического поиска архитектуры нейросети (NAS) известна давно [3], а в современных работах такие алгоритмы показывают сравнимые со state-of-the-art архитектурами результаты [4]. Однако, используемая обычно методология оптимизации дискретной структуры нейросети [5], значительно ограничивает эффективность оптимизаций, не позволяя использовать методы градиентной оптимизации.

Альтернативный подход подразумевает переход от дискретной параметризации структуры нейросети к непрерывной. В работе [6], такой переход производится над функциями активации. Затем используется градиентная оптимизация [7], и выбирается функция с наибольшим весом в каждом отдельном случае.

В данной работе развивается идея релаксации. Оптимизируются не только функции активации, но и остальные структурные параметры нейросети. Предлагается ввести регуляризацию структуры, позволяющую калибровать дискретность параметризации структуры нейросети [8]. При снижении температуры распределение значений структурных параметров приближается к дискретному, что упрощает итоговый выбор структуры нейросети.

Для оценки полученной системы используются выборки MNIST [9], CIFAR-10. Предметом оценки является не только точность ответов на тестовой подвыборке, но и устойчивость результатов.

## 2 Постановка задачи

Пусть заданы обучающая и валидационная выборки:

$$\mathcal{D}^{\text{train}} = \{\mathbf{x}_i, y_i\}, \quad i = 1, \dots, m^{\text{train}},$$

$$\mathfrak{D}^{\text{valid}} = \{\mathbf{x}_i, y_i\}, \quad i = 1, \dots, m^{\text{valid}},$$

состоящие из множеств пар объект-метка,

$$\mathbf{x}_i \in \mathbf{X} \subset \mathbb{R}^n, \quad y_i \in \mathbf{Y} \subset \mathbb{R}.$$

Метка  $y$  объекта  $\mathbf{x}$  принадлежит множеству  $y \in \mathbf{Y} = \{1, \dots, Z\}$ , где  $Z$  - количество классов.

Модель задаётся ориентированным графом  $\mathbf{G} = (\mathbf{V}, \mathbf{E})$ , где для каждого ребра  $(i, j)$  заданы базовые функции  $\mathbf{g}^{i,j}$ ,  $|\mathbf{g}^{i,j}| = K^{i,j}$  и их веса  $\gamma^{i,j}$ . Требуется построить такую модель  $\mathbf{f}$  с параметрами  $\mathbf{W} \in \mathbb{R}^n$ :

$$\mathbf{f}(\mathbf{x}, \mathbf{W}) = \{\mathbf{f}_i(\mathbf{x}, \mathbf{w}_i)\}_{i=1}^{|\mathbf{V}|}$$

где  $\mathbf{f}_i(\mathbf{x}, \mathbf{w}_i)$  - подмодель с параметрами  $\mathbf{w}_i$  задаётся как:

$$\mathbf{f}_i(\mathbf{x}, \mathbf{w}_i) = \sum_{j \in \text{adj}(i)} \langle \gamma^{i,j}, \mathbf{g}^{i,j} \rangle \mathbf{f}_j(\mathbf{x}, \mathbf{w}_j)$$

Тогда параметры модели определяются как конкатенация всех параметров каждой подмодели:  $\mathbf{W} = [\mathbf{w}_1, \dots, \mathbf{w}_{|\mathbf{V}|}]$ , а структура модели  $\mathbf{\Gamma}$  задаётся вектором  $\{\gamma^{i,j}\}_{\mathbf{E}}$ .

Функция потерь на обучении  $L$  и функция потерь на валидации  $Q$  задаются как:

$$L(\mathbf{W}, \mathbf{\Gamma}) = \log p(\mathbf{Y}^{\text{train}} | \mathbf{X}^{\text{train}}, \mathbf{W}, \mathbf{\Gamma}) + \mathbf{e}^{\mathbf{A}} \|\mathbf{W}\|^2,$$

$$Q(\mathbf{W}, \mathbf{\Gamma}) = \log p(\mathbf{Y}^{\text{valid}} | \mathbf{X}^{\text{valid}}, \mathbf{W}, \mathbf{\Gamma}) + \lambda p(\mathbf{\Gamma}),$$

где  $\mathbf{A}$  и  $\lambda$  - регуляризационные слагаемые,  $p(\mathbf{\Gamma})$  - произведение всех произведение вероятностей всех  $\gamma^{i,j} \in \mathbf{\Gamma}$ . Перед подсчётом значения функции потерь на валидации делается априорное предположение о распределении вектора  $\mathbf{\Gamma} = \{\gamma^{i,j}\}$ : вектор структуры модели имеет распределение либо Дирихле [?] либо Gumbel-Softmax [?].

Вектор  $\{\gamma^{i,j}\}$  имеет распределение Дирихле с параметром  $\alpha$ , если:

$$f(\gamma) = f(\gamma_1, \dots, \gamma_K) = \begin{cases} \frac{\mathbf{F}(K \times \alpha)}{\mathbf{F}(\alpha)^K} \prod_{i=1}^K \gamma_i, & \gamma \in \mathbf{S} \\ 0, & \gamma \notin \mathbf{S} \end{cases}$$

, где  $\mathbf{F}$  - гамма-функция,  $\mathbf{S}$  - симплекс:  $\{\gamma \in \mathbb{R}^K : \sum_{i=1}^K \gamma_i = 1, \gamma_i \geq 0\}$ .

Вектор  $\{\gamma^{i,j}\}$  имеет распределение Gumbal-Softmax с параметром  $\alpha$  и параметром  $\tau$ , если:

$$f(\gamma_1, \dots, \gamma_K) = (K-1)! \tau^{K-1} \alpha^K \prod_{i=1}^K \frac{\gamma_i^{-\tau-1}}{\alpha \sum_{j=1}^K \gamma_j^{-\tau}}$$

При  $\tau \rightarrow \inf$  распределение Gumbal-Softmax эквивалентно многомерному равномерному распределению.

Требуется решить задачу двухуровневой оптимизации, оптимизируя параметры модели по обучающей выборке, а структуру модели по валидационной:

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

$$\Gamma, \mathbf{A} = \min_{\Gamma} Q(\mathbf{W}^*(\Gamma), \Gamma)$$

### 3 Релаксация

Известно множество всех возможных операций  $\mathbf{g}^{i,j} \in \mathbf{G}$ . Для перехода к непрерывному пространству таких функций проводится релаксация каждой операции:

$$\overline{\mathbf{g}(\mathbf{x})} = \sum_{\gamma \in \Gamma} \frac{e^{\gamma}}{\sum_{\gamma' \in \Gamma} e^{\gamma'}} \mathbf{g}(\mathbf{x})$$

После релаксации необходимо совместное исследование  $\Gamma$  и весов  $\mathbf{w}$  всех смешанных операциях  $\overline{\mathbf{g}^{i,j}}$ .

### Литература

- [1] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.
- [2] Gaurang Panchal, Amit Ganatra, Parth Shah, and Devyani Panchal. Determination of over-learning and over-fitting problem in back propagation neural network. *International Journal on Soft Computing*, 2(2):40–51, 2011.
- [3] Geoffrey F Miller, Peter M Todd, and Shailesh U Hegde. Designing neural networks using genetic algorithms. In *ICGA*, volume 89, pages 379–384, 1989.
- [4] Barret Zoph and Quoc V Le. Neural architecture search with reinforcement learning. *arXiv preprint arXiv:1611.01578*, 2016.
- [5] Renato Negrinho and Geoff Gordon. Deeparchitect: Automatically designing and training deep architectures. *arXiv preprint arXiv:1704.08792*, 2017.
- [6] Hanxiao Liu, Karen Simonyan, and Yiming Yang. Darts: Differentiable architecture search. *arXiv preprint arXiv:1806.09055*, 2018.
- [7] Joel Grus. *Data science from scratch: first principles with python*. "O'Reilly Media, Inc. 2015.
- [8] Eric Jang, Shixiang Gu, and Ben Poole. Categorical reparameterization with gumbel-softmax. *arXiv preprint arXiv:1611.01144*, 2016.
- [9] Yann LeCun and Corinna Cortes. MNIST handwritten digit database. 2010.

Поступила в редакцию