

# Автоматическое построение нейросети оптимальной сложности

Илья Гридасов

Московский физико-технический институт

*Курс:* Численные методы обучения по прецедентам (практика, В. В. Стрижов), группа 694, весна 2019  
*консультанты:* О. Ю. Бахтеев, В. В. Стрижов

# Построение нейросети оптимальной сложности

## Цель исследования

Предложить алгоритм поиска оптимальной архитектуры нейронной сети, найдя баланс между её сложностью и качеством.

## Проблема

Многие известные state-of-the-art нейросети требуют избыточно большого количества параметров, вследствие требуют большой выборки, поэтому часто не применимы на практике.

## Метод решения

Ошибка декомпозируется на две независимые части - непосредственная ошибка алгоритма и стоимость описания его модели. Второй параметр оптимизируется методом вариационного вывода.

## Эволюционные методы

- Esteban Real, Alok Aggarwal, Yanping Huang, and Quoc V Le. Regularized evolution for image classifier architecture search, 2018.
- Hanxiao Liu, Karen Simonyan, Oriol Vinyals, Chrisantha Fernando, and Koray Kavukcuoglu. Hierarchical representations for efficient architecture search. ICLR, 2018b.

## RL методы

- Zhao Zhong, Junjie Yan, Wei Wu, Jing Shao, and Cheng-Lin Liu. Practical block-wise neural network architecture generation, 2018.
- Hieu Pham, Melody Y Guan, Barret Zoph, Quoc V Le, and Jeff Dean. Efficient neural architecture search via parameter sharing. ICML, 2018b.

# Постановка задачи поиска архитектуры

## Дано

Выборка размера  $m$ :  $D_m = \{\mathbf{x}_i, y_i\}_{i=1}^m$ ,

где  $\mathbf{x}_i \in \mathbb{R}^n$  - вектор признаков,  $y_i \in \mathbb{Y}$ .

$\alpha$  - параметры архитектуры,  $w$  - параметры модели.

## Функция ошибки

Определим функцию ошибки на обучающей выборке  $\mathcal{L}_{train}$  и на валидации  $\mathcal{L}_{val}$ , получаем задачу двух-уровневой оптимизации:

$$\min_{\alpha} \mathcal{L}_{val}(w^*(\alpha), \alpha) \quad (1)$$

$$\text{s.t. } w^*(\alpha) = \operatorname{argmin}_w \mathcal{L}_{train}(w, \alpha) \quad (2)$$

# Постановка задачи поиска архитектуры

## Декомпозиция ошибки

Ошибку нейронной сети определим как минус логарифм правдоподобия

$$\mathcal{L}^N(w, \alpha, D) = -\ln \Pr(D|w, \alpha) = - \sum_{(x,y) \in D} \ln \Pr(y|x, w, \alpha)$$

## Вариационный вывод

Зададим распределение на параметры  $Q(\beta)$ , будем минимизировать вариационную свободную энергию:

$$\mathcal{F} = -\left\langle \ln \frac{\Pr(D|w)P(w|\alpha)}{Q(w|\beta)} \right\rangle_{w \sim Q(\beta)}$$

где  $\langle \xi \rangle_{w \sim P}$  - математическое ожидание величины  $\xi$  по  $w \sim P$ .  $\mathcal{F}$  раскладывается на две части:

$$\mathcal{F} = \langle \mathcal{L}^N(w, D) \rangle_{w \sim Q(\beta)} + D_{KL}(Q(\beta) || P(\alpha))$$

## Пространство поиска

Представим архитектуру нейронной сети в виде графа вычислений, в котором каждая вершина зависит от предыдущих при помощи некоторой операций из фиксированного множества  $\mathcal{O}$ :

$$x^{(j)} = \sum_{i < j} o^{(i,j)}(x^{(i)}) \quad (3)$$

## Непрерывная релаксация

При помощи softmax можно сделать пространство поиска непрерывным.

$$\bar{o}^{(i,j)}(x) = \sum_{o \in \mathcal{O}} \frac{\exp(\alpha_o^{(i,j)})}{\sum_{o' \in \mathcal{O}} \exp(\alpha_{o'}^{(i,j)})} o(x) \quad (4)$$

## Аппроксимация градиента по $\alpha$

Для эффективного поиска градиента по архитектуре воспользуемся следующей аппроксимацией:

$$\nabla_{\alpha} \mathcal{L}_{val}(w^*(\alpha), \alpha) \quad (5)$$

$$\approx \nabla_{\alpha} \mathcal{L}_{val}(w - \xi \nabla_w \mathcal{L}_{train}(w, \alpha), \alpha) \quad (6)$$

## Алгоритм

Создать граф  $\bar{o}^{(i,j)}$ , параметризованный  $\alpha^{(i,j)}$  для каждого ребра  $(i, j)$  вычислительного графа

**while** веса меняются **do**

- 1. Обновить  $\alpha$  шагом спуска  $\nabla_{\alpha} \mathcal{L}_{val}(w - \xi \nabla_w \mathcal{L}_{train}(w, \alpha), \alpha)$
- 2. Обновить  $w$  по градиенту  $\nabla_w \mathcal{L}_{train}(w, \alpha)$

Вывести итоговую архитектуру по выученным весам  $\alpha$ .

## Цель эксперимента

Проверить работоспособность метода.

## Синтетическая выборка

Выборка генерируется из распределения:

$$y_i \sim \alpha_0 \sin(w_0 x_i) + \alpha_1 \cos(w_1 x_i) + \mathcal{N}(0, 0.1)$$

$$x_i \sim U[0, 1]$$

$w_0, w_1$  - выступают в качестве параметров модели,  $\alpha_0, \alpha_1$  - параметры архитектуры.



Траектория спуска при разных значения параметра  $\xi$  оптимизации.

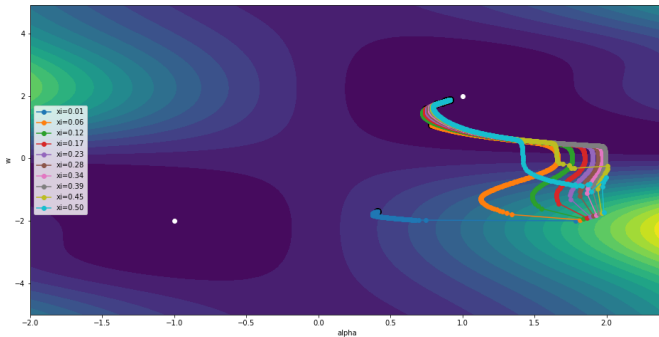


Рис.: метод DARTS, синтетическая выборка

- Задача поиска оптимальной архитектуры сведена к задаче двух-уровневой оптимизации, для которой существуют эффективные приближенные методы.
- Показана работоспособность предложенного метода на синтетических данных.
- Далее предлагается применить данный метод на реальных данных, для оптимизации архитектуры глубоких нейронных сетей.