Автоматическое построение нейросети оптимальной сложности

Криницкий К. Д. , Бахтеев О. Ю. , Стрижов В. В.
krinitskiy.kd@phystech.edu; bakhteev@phystech.edu; strijov@phystech.edu

Аннотация: В этой статье рассматривается задача поиска оптимальной структуры нейронной сети. Нейросеть рассматривается как вычислительный граф, реализуемый с помощью библиотеки Pytorch. Предусматривается, что число структурных параметров можно уменьшить без существенной потери качества классификации или регрессии. Исследуются изменения характеристик нейронной сети при колебании структурных параметров. Предлагается новый метод, учитывающий особенности задачи, который совершенствует структуру нейронной сети в процессе оптимизации. Итоговым результатом является модель , дающая приемлемое качество классификации либо регрессии, и не являющаяся избыточной по параметрам. Для анализа качества представленного алгоритма проводятся эксперименты на выборках Boston, MNIST и CIFAR-10.

Ключевые слова: *нейронные сети, графовые вычисления, оптимизация гиперпараметров, вариационный вывод*

Введение

В данной работе решается задача построения нейронной сети оптимальной сложности. Под оптимальной моделью имеется ввиду та модель, которая является не избыточной по своим параметрам, но при этом дающая приемлемый результат классификации либо регрессии. В данной статье рассматривается оптимизация структурных параметров, таких как: размерность слоев и их количество, функция активации.

Существует ряд способов выбора модели оптимальной сложности. В работе [1] рассматривается модель гауссовского процесса, поясняется как нужно оптимизировать структуру, в случае недостатка информации о входных данных. В [2] применяется баейсовская модель, а такаже говорится о принципе "Бритва Оккама", который гласит, что из моделей одинаковой точности выбирается наиболее простая. В работах [3, 4, 5, 6] рассматривается градиентный метод, также являющийся одним из способов оптимизации.

Построение оптимальной нейронной сети - задача ресурсоемкая и вычислительно трудная. Из-за большого количества структурных параметров время обучение сети сильно возрастает. В данной работе используется эффективный алгоритм, основанный на методе DARTS [7]. Выбор оптимальных значений структурных параметров происходит благодаря процедуре релаксации: переход от дискретного набора параметров к непрерывному.

Проверка полученного алгоритма произведена на данных MNIST [8], CIFAR-10 [9], Boston Housing [10] также на синтетических данных. Модели, полученные представленным алгоритмом сравниваются с моделями, построенными с использованием базовых алгоритмов.

Постановка задачи

Пусть заданы обучающая и валидационная выборки

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

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

$\mathbf{x_i}$ - вектор признаков i-го объекта, а $y_i \in \mathbf{Y} \subset \mathbb{R}$, $\mathbf{Y} = \{1, \ldots, Z\}$, Z - количество классов. Модель описывается ориентированным графом (V, E). Для каждого ребра $(j, k) \in E$ определён вектор базовых функций $\mathbf{g}_{j,k}$ мощностью $K_{j,k}$. Модель $\mathbf{f}(\mathbf{x}, \mathbf{W})$ задаётся параметрами подмоделей $\{\mathbf{f}_v\}_{v=1}^{|V|}$ и структурными параметрами $\boldsymbol{\gamma}^{j,k}$. Каждая подмодель f_v представляется следующим образом:

$$f_v(\mathbf{x}, \mathbf{w}_v) = \sum_{k \in adj(v_i)} \langle \gamma_{j,k}, g_{j,k} \rangle f_k(\mathbf{x}, \mathbf{w}_k), \quad f_0(\mathbf{x}) = \mathbf{x}.$$

Параметры модели $\mathbf{W} = [\mathbf{w}_1, \ldots, \mathbf{w}_{|\mathbf{V}|}]$ - конкатенация параметров всех подмоделей $\{f_v\}_{v=1}^{x}|V|$, а структура модели $\boldsymbol{\Gamma}$ - конкатенация структурных параметров $\boldsymbol{\gamma}^{j,k}$.
Пусть $L(\mathfrak{D}^{train}, \mathbf{W}, \boldsymbol{\Gamma}, \mathbf{A})$ - функция потерь на обучении, а $Q(\mathfrak{D}^{valid}, \mathbf{W}, \boldsymbol{\Gamma})$ - фунция потерь на валидации. Тогда:

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

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

$\mathbf{A}$ - гиперпараметр, отвечающий за регуляризацию.
Гиперпараметры находятся решением двухуровневой задачи оптимизации:

$$\boldsymbol{\Gamma}^*, \mathbf{A}^* = argmin_{\boldsymbol{\Gamma}, \mathbf{A}} Q(\mathfrak{D}^{valid}, \mathbf{W}^*(\boldsymbol{\Gamma}, \mathbf{A}), \boldsymbol{\Gamma}, \mathbf{A}),$$

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

Литература

[1] *Carl E.* Gaussian Processes in Machine Learning. 2005.

[2] *David J.C. MacKay* Information Theory, Inference, and Learning Algorithms. 2005.

[3] *J. Luketina, M. Berglund, T. Raiko, and K. Gref* Scalable gradient-based tuning of continuous regularization hyperparameters. 2016.

[4] *D. Maclaurin, D. Duvenaud, R. P. Adams* Gradient-based Hyperparameter Optimization through Reversible Learning. 2015.

[5] *L. Franceschi, M. Donini, P. Frasconi, M. Ponti* Forward and Reverse Gradient-Based Hyperparameter Optimization. 2017.

[6] *Anonymous authors* Online hyper-parameter optimization. 2018.

[7] *Hanxiao L., Simonyan K., Yang .Y* DARTS: Differentiable Architecture Search. 2018. URL: `https://arxiv.org/abs/1806.09055`.

[8] *Yann LeCun, Corinna Cortes, Christopher J.C. Burges,* The MNIST Database of Handwritten Digits 1998. URL:http://yann.lecun.com/exdb/mnist/

[9] *A. Krizhevsky, V. Nair, G. Hilton.* The CIFAR-10 dataset 2009. URL: `http://www.cs.toronto.edu/~kriz/cifar.html`

[10] *Harrison Jr. , Rubinfeld D., Daniel L.* Hedonic housing prices and the demand for clean air. 1978. URL:https://archive.ics.uci.edu/ml/machine-learning-databases/housing/.