

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. Будут изучены изменения характеристик нейронной сети при колебании структурных параметров. Итоговым результатом будет являться модель, дающая приемлемое качество классификации либо регрессии, и не являющаяся избыточной по параметрам.

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 [?] рассматривается модель гауссовского процесса, поясняется как нужно оптимизировать структуру, в случае недостатка информации о входных данных. В [?] применяется байесовская модель, а также говорится о принципе "Бритва Оккама", который гласит, что из моделей одинаковой точности выбирается наиболее простая. В работах [?, ?, ?, ?] 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 [?]. Выбор оптимальных значений структурных параметров происходит благодаря процедуре релаксации: переход от дискретного набора параметров к непрерывному.

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

2 Постановка задачи

Рассмотрим для нашей задачи обучающую и валидационную выборки

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

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

\mathbf{x}_i - вектор признаков i -го объекта, а $y_i \in \mathbf{Y} \subset \mathbb{R}$, $\mathbf{Y} = \{1, \dots, Z\}$, Z - количество классов. Модель описывается ориентированным графом (V, E) . Для каждого ребра $(j, k) \in E$ определен вектор базовых функций $g_{j,k}$ мощностью $K_{j,k}$. Модель $\mathbf{f}(\mathbf{x}, \mathbf{W})$ задаётся параметрами

подмоделей $\{f_v\}_{v=1}^{|V|}$ и структурными параметрами $\gamma^{j,k}$. Каждая подмодель f_v представляется следующим образом:

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

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

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

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

\mathbf{A} - гиперпараметр, отвечающий за регуляризацию

Гиперпараметры находятся решением двухуровневой задачи оптимизации:

$$\Gamma, \mathbf{A} = \underset{\Gamma, \mathbf{A}}{\operatorname{argmin}} Q(\mathfrak{D}^{\text{valid}}, \mathbf{W}^*(\Gamma, \mathbf{A}), \Gamma, \mathbf{A}),$$

$$\mathbf{W}^* = \underset{\mathbf{W}}{\operatorname{argmin}} L(\mathfrak{D}^{\text{train}}, \mathbf{W}, \Gamma, \mathbf{A})$$