Автоматическое построение нейросети оптимальной сложности

Улитин А.Ю. , Бахтеев О.Ю , Стрижов В.В.
ulitin.ayu@phystech.edu
Московский физико-технический институт

Работа посвящена поиску оптимальной модели нейросети. Нейросеть представляется как граф, где ребрам соответствуют нелинейные операции, а вершины - промежуточные представления. Параметры сети разделим на три типа: Параметры, отвечающие за итоговое качество классификации,гиперпараметры, отвечающие за процесс переобучения и предотвращение переобучения, а также структурные параметры, которые отвечают за структуру модели. Структура нейросети опередеяется вершинами симплекса. Будем проводить релаксацию структуры для решения задачи оптимизации.

Ключевые слова: *нейросети, оптимизация гиперпараметров,робастность модели.*

Введение

В данной работе рассматривается метод построения оптимальной нейронной сети. Под оптимальной сетью понимается модель, дающая приемлемое качество при небольшом количестве параметров. Под структурой понимается набор структурных параметров: количество слоев, нейронов в каждом слое, а также функции активации в каждом нейроне. В данной работе в качестве критерия выбора модели предлагается сложность модели, то есть величина, учитывающая сложность описания совокупности выборки и модели.

Существует несколько способов построения оптимальной нейронной сети. Один из основных - оптимальное прореживание [1]. Этот способ заключается в том, что из максимально сложной модели удаляются связи, и получается упрощенная сеть. В работе [2] предложен байссовский метод оптимизации сети, а в работе [3] рассмотрен метод градиентного спуска. Кроме того в [4] используется метообучение, которое по некоторой входной выборке возвращает оптимальные гиперпараметры.

В виду того, что у моделей значительное количество параметров и гиперпараметров, процесс оптимизации может быть затратным. В данной работе используется эффективный по ресурсам метод, в основе которого лежит алгоритм DARTS [5], где на вход мы получаем некоторый набор входных данных, а также функции активации. Оптимизируя параметры и гиперпараметры параллельно, мы на выходе получим оптимальную нейронную сеть.

Проверка и анализ метода проводится на выборках [6, 7, 8] и синтетических данных. В эксперименте проводится сравнение полученного результата с моделями, полученными другими базовыми алгоритмами.

Постановка задачи

Пусть заданы обучающая и вылидационная выборки:

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

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

состоящие из множеств пар объект-метка,

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

$\mathbf{Y} = \{1, \ldots, Z\}$, где Z - количество классов.

Модель задаётся ориентированным графом $\mathbf{G} = (\mathbf{V}, \mathbf{E})$, где для каждого ребра (i, j) задан вектор базовых функций $\mathbf{g}^{i,j}$, с мощностью $|\mathbf{g}^{i,j}| = K^{i,j}$ и весами $\boldsymbol{\gamma}^{i,j}$. Требуется построить такую модель $\mathbf{f}$ с параметрами $\mathbf{W} \in \mathbb{R}^{\mathrm{n}}$:

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

где $\mathbf{f_i}(\mathbf{x}, \mathbf{w_i})$ - подмодель с параметрами $\mathbf{w}_i$ задаётся через графовое представление как:

$$\mathbf{f}_i(\mathbf{x}, \mathbf{w}_i) \;=\; \sum_{k \in adj(i)} \left\langle \boldsymbol{\gamma}^{i,k}, \mathbf{g}^{i,k} \right\rangle \mathbf{f}_k(\mathbf{x}, \mathbf{w}_k)$$

.

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

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

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

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

В итоге получаем задачу двухуровневой оптимизации, оптимизируя параметры модели по обучающей выборке, а структуру модели по валидационной:

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

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

Релаксация модели

Для более эффективного решения задачи поиска оптимальной структуры нейросети, переходим от дискретной задачи поиска оптимальной базовой функции к непрерывной, производя 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 $|\mathbb{K}|$, где $\mathbb{K}$ — мощность множества кандидатов на роль базовой функции. Этот вектор параметризует 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 $g^{(i,j)} = \arg\max_{g \in \mathbb{K}} \gamma_g^{(i,j)}$.

Регуляризация структуры модели

Регуляризация структуры проводится добавлением к фунции потерь Q слагаемого $\lambda P(\Gamma)$,

где $P(\Gamma)$ есть произведение всех вероятностей возникновения веса $\gamma^{(i,j)}$.
Таким образом функция потерь принимает вид:

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

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 структурой модели Γ 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] *Yann Le Cun, John S. Denker and Sara A. Solla.* Optimal Brain Damage. 1989.

[2] *A. Neal and M. Radfor* Bayesian Learning for Neural Networks.. 1995.

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

[4] *D. Maclaurin and D. Duvenaud and R. Adams.* Gradient-based Hyperparameter Optimization Through Reversible Learning 2015.

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

[6] *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-datab...

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

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