Предсказание графовой структуры нейросетевой модели

Калугин Д.И., Бахтеев О.Ю., Стрижов В.В.

26 апреля 2018 г.

Аннотация

Статья посвящена задаче поиска оптимальной структуры нейронной сети. Предлагается построить соответствие между подвыборками обучающей выборки и структурами, которые будут качественно решать соответствующую задачу, и на основе этих структур прогнозировать оптимальную архитектуру всей нейросети. Анализ качества предложенного метода проводится на основе вычислительных экспериментов на выборке MNIST, а также на синтетических данных.

1 Введение

Одна из основных проблем глубоких нейронных сетей — это большие вычислительные затраты, связанные с их обучением. Так, авторы [1] пишут, что для задачи классификации изображений использовали 500 GPU. В статье [7] авторы пишут, что в аналогичной задаче использовали нейросеть с 650,000 нейронов и 60 миллионами параметров. Из этих примеров видно, что оптимизационная задача, решаемая в модели нейронных сетей вычислительно сложная.

Основной вопрос, возникающий при построении модели нейросети — это выбор количества слоев в сети и их архитектура. Этот выбор значительно влияет на итоговое качество модели. Кроме того, нейронные сети являются сильно избыточными по количеству параметров. В статье [6] авторы показывают, что более 95% параметров нейросети можно предсказывать без потери качества.

Есть разные подходы к задаче поиска оптимальной архитектуры нейросети. Авторы [2] предлагают алгоритм бустинга AdaNet для поиск оптимальной структуры модели. В статье [3] для этого используются гауссовские процессы. Эти алгоритмы имеют кубическую сложность оптимизации по количеству запусков оптимизации. В [4,5] предлагается использовать для решения данной задачи методы обучения с подкреплением.

Предлагается альтернативный подход к этой задаче, заключающийся в следующем. Сперва для небольших подмножеств исходной выборки находим структуры сети с малым количеством неинформативнных параметров и показывающие хорошее качество прогнозирования. После этого, на полученных парах "подвыборка - оптимальная структура сети"обучаем метамодель, с помощью которой уже предсказываем структуру модели и ее параметры для всей выборки.

Для анализа качества предложенного алгоритма проводится ряд вычислительных экспериментов на данных выборки MNIST, а также на синтетических данных. Полученные результаты сравниваются с результатами алгоритма AdaNet[2] и с результатами алгоритмов, использующих обучение с подкреплением [4], [5].

2 Постановка задачи

Введем дополнительные обозначения, соответствующие архитектуре нейросетевой модели.

Определение 1 *Пусть дана многослойная нейросеть, у которой t слоев, в i-ом слое n_i нейронов, и j-ому нейрону соответствует функция активации α_{ij}. Обозначим такую модель как $\mathfrak{f} = \mathfrak{f}(t, n_1, \ldots, n_t) \in \mathfrak{F}$, где $\mathfrak{F}$ — множество всех нейросетевых моделей. Вектор из функций активации для i-го слоя обозначим за $\vec{\alpha}_i$. Матрицу весов между i-ым и $(i+1)$-ым слоем обозначим за $W_i \in \mathbb{R}^{n_i \times n_{i+1}}$. Тогда для того, чтобы задать нейросеть, нам необходимо знать матрицы весов между всеми слоями, а также все функции активации. Введем обозначение:*

$$\Gamma = \{W_1, \vec{\alpha}_1, \ldots, W_t, \vec{\alpha}_t\}.$$

Размерность Γ есть:

$$\dim \Gamma = \sum_{i=1}^{t} (n_{i-1} \cdot n_i + n_i).$$

Первое слагаемое соответствует матрице весов между слоями $(i-1)$ и i, а второе — вектору из функций активаций i-го слоя. При этом параметры, соответствующие матрице весов — вещественные, а функциям активации — классовые.

Множество всех возможных Γ *обозначим за* $\mathfrak{G}$*. Таким образом, можно построить однозначное соответствие между* $\mathfrak{F}$ *и множеством* $\mathfrak{G}$*, сопоставляя каждой нейросетевой модели ее представление как последовательности матриц и функций активации.*
Для каждой матрицы W_k *введем матрицу*

$$V_k = ([(W_k)_{ij} \neq 0])_{i,j},$$

которая показывает наличие связи между соответствующими нейронами. Тогда $V_i \in \{0,1\}^{n_i \times n_{i+1}}$*. Количество активных нейронов на* i*-ом слое будем обозначать* k_i*, оно соответствует количеству ненулевых элементов матрицы* W_i*, или, что то же самое, количеству единиц в матрице* V_i*. Общее количество активных нейронов в модели обозначим как* $K = \sum_{i=1}^{t} k_i$*. Введем обозначение:*

$$Z = \{V_1, \vec{\alpha}_1, \ldots, V_t, \vec{\alpha}_t\}.$$

Таким образом, объект Z *задает архитектуру нейросети. Множество всех возможных* Z *обозначим за* $\mathfrak{Z}$*. Если считать, что функцию активации выбирается из* a *вариантов, то для* Z *будет*

$$N = \prod_{i=1}^{t} 2^{n_{i-1} \cdot n_i} \cdot a^{n_i}.$$

Первый множитель соответствует количеству бинарных матриц между соседними уровнями, а второй — количеству возможных векторов из функций активаций.
Обученную нейросетевую модель, с архитектурой Z *будем обозначать* $\mathfrak{f}_Z$*.*

Определение 2 *Пусть задана архитектура нейросети* Z *и задача предсказания по данным* X *ответов* y*. Ответы, выдаваемые моделью* $\mathfrak{f}_Z$ *на данных* X *обозначим за* $\mathfrak{f}_Z(X)$*. Введем функционал качества модели:*

$$\mathfrak{L}(Z) = L(y, \mathfrak{f}_Z(X)) + g(K(Z)),$$

где $L(y,)$ *— функция ошибки на предсказаниях(accuracy, ROCAUC, etc — для классификации; MSE, MAE, etc — для регрессии), а* $g(K)$ *— монотонно-возрастающая функция, которая штрафует модель за количество активных нейронов, регуляризатор модели.*

Определение 3 *Пусть задана выборка данных* $D = (X, y)$*, где* X *— матрица объекты-признаки, а* y *— вектор ответов. Также дана архитектура* Z_{opt}*, оптимальная относительно функционала качества* $\mathfrak{L}$ *при некоторых фиксированных функции потерь* L *и функции-регуляризатора* g*. Пусть нам необходимо определить, насколько новая архитектура* Z *"близка"к* Z_{opt}*. Введем функцию близости для архитектур с одинаковым количеством слоев и одинаковом количестве нейронов в каждом из слоев, то есть* $(t', n'_1, \ldots, n'_t)_Z = (t, n_1, \ldots, n_t)_{Z_{opt}}$*:*

$$F(Z_{opt}, Z) = \mathfrak{L}(Z) - \mathfrak{L}(Z_{opt}) + \sum_{k=1}^{t} \Big(\sum_{i=1}^{n_t} \sum_{j=1}^{n_{t-1}} (\alpha \cdot [(V_k)_{ij} = 1, (V_{opt,k})_{ij} = 0] + \beta[(V_k)_{ij} = 0, (V_{opt,k})_{ij} = 1]) \Big).$$

Первые два слагаемых показывают разницу функционала качества моделей, а сумма, идущая после этого — штраф, накладываемый на саму архитектуру Z*, за ее отличие от* Z_{opt}*. При этом если у нас в* Z *неактивен нейрон, который активен в* Z_{opt}*, это не так критично, как в обратном случае. Это регулируется параметрами* α *и* β*. Так, если мы хотим минимизировать количество активных нейронов, то нужно брать* $\alpha > \beta$*.*

Теперь перейдем к описанию задачи. Разберем алгоритм на примере задачи бинарной классификации. Дана выборка данных:

$$\mathcal{D} = \{(\mathbf{x}_i, y_i)\}, i \in 1, \ldots, m$$

из m пар объект-класс, где $\mathbf{x}_i \in \mathbb{R}^n, y_i \in \{0, 1\}$, где n — количество признаков.

Зафиксируем некоторую функцию качества предсказаний L и возрастающую функцию g, которые вместе задают функционал качества модели $\mathfrak{L}$.

Также дана выборка пар "подвыборка-архитектура":

$$\mathcal{P} = \{(D_j, Z_j)\}, j \in 1, \ldots, k,$$

где $D_j \in \mathcal{D}, Z_j \in \mathfrak{Z}$, причем известно, что архитектура Z_j оптимальна для выборки D_j в терминах функционала $\mathfrak{L}$.

Необходимо предсказать оптимальную архитектуру для всей выборки данных, то есть $Z_{opt}(\mathcal{D})$.

Также, как и в обычных алгоритмах машинного обучения, здесь есть компромисс между предсказательной силой модели и ее ограниченностью по параметрам. В нашем случае это регулируется функцией $g(K)$ — если она будет расти очень быстро, модели будет выгодно занулить большинство весов при нейронах, и нейросеть может иметь слабую предсказательную способность. Если же функция $g(K)$ будет расти медленно, модели будет выгодно не занулять веса при нейронах, в пользу слабого прироста предсказательной способности, и архитектура окажется избыточной.

2.1 Базовый алгоритм

В рамках базового эксперимента решается задача построения выборки, которая при постановке задачи считалась заданной — выборка пар "подвыборка-архитектура". Для поиска оптимальной архитектуры используется случайный поиск. В качестве функции ошибки на предсказаниях используется метрика accuracy. Функция-регуляризатор выбрана как $g(K) = C \cdot K$, где константа C настраивается отдельно.

Архитектуры перебираются среди двухуровневых нейронных сетей. Результат действия такой модели на исходный объект можно записать следующим образом:

$$f(\mathbf{x}) = \sigma_2 \left(\underset{n \times n}{W_2} \sigma_1 \left(\underset{n \times n}{W_1} \cdot \mathbf{x} \right) \right).$$

Здесь σ_1, σ_2 — функции активации первого и второго уровней соответственно, W_1, W_2 — матрицы весов соответствующих уровней (считаем, что в скрытом уровне также n нейронов). В случае, когда σ_i — функция, одинаковая для всех нейронов одного уровня, будет существовать много изоморфных друг другу архитектур. Так как мы хотим разнообразить класс архитектур, которые мы можем получать, для каждого из нейронов мы будем выбирать функцию активации отдельно из некоторого заранее определенного класса.

Итого мы имеем $(2n^2 + 2k^n)$ параметров, которые необходимо оптимизировать, где k — количество различных функций активаций. Первое слагаемое соответствует элементам матрицы W_i, второе — различным комбинациям функций активации для нейронов i-го слоя.

Полный перебор всех возможных архитектур вычислительно сложный, поэтому проводится перебор по случайному подмножеству с фиксированной долей активных нейронов.

3 Эксперимент

Список литературы

[1] Barret Zoph, Vijay Vasudevan, Jonathon Shlens, Quoc V. Le *Learning Transferable Architectures for Scalable Image Recognition*

[2] Corinna Cortes, Xavi Gonzalvo, Vitaly Kuznetsov, Mehryar Mohri, Scott Yang *AdaNet: Adaptive Structural Learning of Artificial Neural Networks*

[3] Kevin Swersky, David Duvenaud, Jasper Snoek, Frank Hutter, Michael A. Osborne *Raiders of the Lost Architecture: Kernels for Bayesian Optimization in Conditional Parameter Spaces*

[4] Barret Zoph, Quoc V. Le *Neural Architecture Search with Reinforcement Learning*

[5] Bowen Baker, Otkrist Gupta, Nikhil Naik, Ramesh Raskar *Designing Neural Network Architectures using Reinforcement Learning*

[6] Misha Denil, Babak Shakibi, Laurent Dinh, Marc'Aurelio Ranzato, Nando de Freitas *Predicting Parameters in Deep Learning*

[7] A. Krizhevsky, I. Sutskever, and G. Hinton *Imagenet classification with deep convolutional neural networks*

[8] Dougal Maclaurin, David Duvenaud, Ryan P. Adams *Gradient-based Hyperparameter Optimization through Reversible Learning*