Министерство науки и высшего образования Российской Федерации
Федеральное государственное автономное образовательное
учреждение высшего образования
«Московский физико-технический институт (национальный
исследовательский институт)»
Физтех школа прикладной математики и информатики
Кафедра «Интеллектуальные системы»

Выбор структуры моделей глубокого обучения

Реферат к вступительному испытанию
(Аспирантура)

Направление подготовки: 09.06.01 Информатика и вычислительная техника

Выполнил:
студент группы М05-904а
Грабовой Андрей Валериевич

Научный руководитель:
Доктор физико-математических наук
Стрижов Вадим Викторович

Москва 2021

Содержание

1 Введение **3**

2 Автоматическое определение релевантности параметров **6**
2.1 Описание задачи 6
2.2 Постановка задачи 6
2.3 Случайное удаление 8
2.4 Оптимальное прореживание 8
2.5 Удаление неинформативных параметров с помощью вариационного вывода 10
2.6 Прореживание сети на основе метода Белсли 10

3 Введение отношения порядка на множестве параметров аппроксимирующих моделей **13**
3.1 Описание задачи 13
3.2 Формальная постановка задачи 14
3.3 Задание отношение порядка на множестве параметров 15
3.4 Фиксация параметров 16

4 Заключение **17**

1 Введение

В силу высокой вычислительной сложности, время оптимизации нейронных сетей может занимать до нескольких дней [47]. Построение и выбор оптимальной структуры нейронной сети также является вычислительно сложной процедурой, которая значимо влияет на итоговое качество модели. При этом алгоритмы оптимизации сходится по большинству параметров сети уже после небольшого числа итераций [28]. Своевременное определение начала сходимости параметров позволит существенно снизить вычислительные затраты на обучение моделей с большим числом параметров. Примерами моделей, с большим число параметров, являются AlexNet [5], VGGNet [6], ResNet [12], BERT [8, 7], mT5 [10], GPT3[9] и другие.

Рост числа параметров моделей глубокого обучения влечет снижение интерпретируемости ответов этих моделей. Первые упоминания о данной проблемы рассмотрены А. Г. Ивахненко [18]. Проблема с неинтерпретируемыми моделями широко сейчас рассматривается в классе задач по adversarial attack [4].

Другой проблемой моделей с большим числом параметров является высокие требования к вычислителю в момент предсказания. Использование избыточно сложных моделей с избыточным числом неинформативных параметров является препятствием для использования глубоких сетей на мобильных устройствах в режиме реального времени. Для снижения числа параметров в литературе рассматривается метод дистилляции модели на основе предсказаний модели учителя [14, 25, 26]. Модель с большим числом параметров называется учитель. Модель учителя дистиллируется в модель с малым числом параметров, которая называется ученик. Основные идеи, которые описывают дистилляцию моделей глубокого обучения предложены в работах Дж. Е. Хинтона и В. Н. Вапником [14, 25, 26]. Работы предлагают использовать предсказания модели учителя для повышения качестве модели ученика. В работе [25] В. Н. Вапником вводится понятие привилегированной информации, которое позволяет использовать дополнительную информацию о данных в момент обучения модели. Работа [26] объединяет идеи дистилляции [14] с иде-

ями привилегированного обучения [25], предложив метод дистилляции модели учителя с большим числом признаков в модель ученика с меньшим числом признаков. В предложенном методе [26] решается двухэтапная задача. На первом этапе строится модель учителя с расширенным признаковым описанием. На втором этапе обучается модель ученика в исходном признаковом описании используя дистилляцию [14]. В работе Дж. Е. Хинтона [14] поставлено множество экспериментов по дистилляции моделей глубокого обучения для задачи классификации. Один из экспериментов проводился на выборке MNIST [39], который показал, что предложенный дистилляции позволяет построить нейросетевую модель меньшей сложности на основе модели большей сложности. Второй эксперимент показывал идею по дистилляции ансамбля моделей в одну нейросетевую модель для решения задачи распознания речи. В работе [14] проводится сравнение дистилляции с моделью смеси экспертов. Дальнейшие работы по дистилляции моделей глубокого обучения рассматривают возможность использования информации о значения параметров модели учителя для оптимизации параметров модели ученика. Работа [3] предлагает метод neuron selectivity transfer, который минимизирует специальную функцию потерь. Данная функция основается на maximum mean discrepancy между выходами слоев модели учителя и модели ученика. В рамках вычислительного эксперимента сравнивалось качество базовой дистилляции с предложенным методов на примере выборок CIFAR [1] и ImageNet [2].

Дистилляция моделей глубокого обучения работает в предположение, что архитектура модели ученика уже известная. Для выборка архитектуры модели ученика предлагается использовать методы прореживания нейросетевых моделей. Существует ряд подходов к построению оптимальной сети. В работах [40, 41] предлагается использовать модель градиентного спуска для оптимизации сети. В [45] используются байесовские методы [46] оптимизации параметров нейронных сетей. Другим методом поиска оптимальной структуры является прореживание избыточно сложной модели [44, 49, 48]. В работе [44] предлагается удалять наименее релевантные параметры на основе значений первой и второй производных функции ошибки. В [34]

предложен метод определения релевантности параметров аппроксимирующих моделей при помощи метода Белсли. Релевантность параметров в работе [34] определяется на основе ковариационной матрицы параметров модели. Другим примером задания порядка на множестве параметров служит l_1-регуляризация [29] и регуляризация ElasticNet [30] для линейных моделей. Порядок, заданный на множестве значений коэффициентов регуляризации, индуцирует порядок на множестве признаковых описаний и указывает на важность признаков. В случае нейросетей для регуляризации параметров используется метод исключения параметров [31, 45]. Данный метод также задает порядок на множестве параметров модели.

Порядок на множестве параметров нейросети можно использовать не только для удаления неимение релевантных параметров, а и для фиксации параметров в процесе оптимизации параметров. Работе [35] посвящена оптимизации структуры нейронной сети, а также выбору параметров, которые можно зафиксировать после некоторой итерации градиентного метода.

2 Автоматическое определение релевантности параметров

2.1 Описание задачи

Данная работа посвящена прореживанию структуры сети. Предлагается удалять наименее релевантные параметры модели. Под релевантностью [44] подразумевается то, насколько параметр влияет на функцию ошибки. Малая релевантность указывает на то, что удаление этого параметра не влечет значимого изменения функции ошибки. Метод предлагает построение исходной избыточной сложности нейросети с большим количеством избыточных параметров. Для определения релевантности параметров предлагается оптимизацировать параметры и гиперпараметры в единой процедуре. Для удаления параметров предлагается использовать метод Белсли [43].

2.2 Постановка задачи

Задана выборка

$$\mathfrak{D} = \{\mathbf{x}_i, y_i\},\ i = 1, ..., N, \tag{2.1}$$

где $\mathbf{x}_i \in \mathbb{R}^m$, $y_i \in \{1, \ldots, Y\}$, Y — число классов. Рассмотрим модель $f(\mathbf{x}, \mathbf{w}) : \mathbb{R}^m \times \mathbb{R}^n \to \{1, \ldots, Y\}$, где $\mathbf{w} \in \mathbb{R}^n$ — пространство параметров модели,

$$f(\mathbf{x}, \mathbf{w}) = \mathrm{softmax}\big(f_1(f_2(...(f_l(\mathbf{x}, \mathbf{w})\big), \tag{2.2}$$

где $f_i(\mathbf{x}, \mathbf{w}) = \tanh(\mathbf{w}^\mathsf{T}\mathbf{x})$, l — число слоев нейронной сети, $i \in \{1 \ldots l\}$. Параметр w_j модели f называется активным, если $w_j \neq 0$. Множество индексов активных параметров обозначим $\mathcal{A} \subset \mathcal{J} = \{1, ..., n\}$. Задано пространство параметров модели:

$$\mathbb{W}_\mathcal{A} = \{\mathbf{w} \in \mathbb{R}^n \mid w_j \neq 0,\ j \in \mathcal{A}\}, \tag{2.3}$$

Для модели f с множеством индексов активных параметров $\mathcal{A}$ и соответствующего ей вектора параметров $\mathbf{w} \in \mathbb{W}_\mathcal{A}$ определим лога-

рифмическую функцию правдоподобия выборки:

$$\mathcal{L}_{\mathfrak{D}}(\mathfrak{D}, \mathcal{A}, \mathbf{w}) = \log p(\mathfrak{D}|\mathcal{A}, \mathbf{w}), \tag{2.4}$$

где $p(\mathfrak{D}|\mathcal{A}, \mathbf{w})$ — апостериорная вероятность выборки $\mathfrak{D}$ при заданных $\mathbf{w}, \mathcal{A}$. Оптимальные значения $\mathbf{w}, \mathcal{A}$ находятся из минимизации $-\mathcal{L}_{\mathcal{A}}(\mathfrak{D}, \mathcal{A})$ — логарифма правдоподобия модели:

$$\mathcal{L}_{\mathcal{A}}(\mathfrak{D}, \mathcal{A}) = \log p(\mathfrak{D}|\mathcal{A}) = \log \int_{\mathbf{w} \in \mathbb{W}_{\mathcal{J}}} p(\mathfrak{D}|\mathbf{w}) p(\mathbf{w}|\mathcal{A}) d\mathbf{w}, \tag{2.5}$$

где $p(\mathbf{w}|\mathcal{A})$ — априорная вероятность вектора параметров в пространстве $\mathbb{W}_{\mathcal{J}}$.

Так как вычисление интеграла (2.5) является вычислительно сложной задачей, рассмотрим вариационный подход [42] для решения этой задачи. Пусть задано распределение q:

$$q(\mathbf{w}) \sim \mathcal{N}(\mathbf{m}, \mathbf{A}_{\text{ps}}^{-1}), \tag{2.6}$$

где $\mathbf{m}, \mathbf{A}_{\text{ps}}^{-1}$ — вектор средних и матрица ковариации, аппроксимирующее неизвестное апостериорное распределение $p(\mathbf{w}|\mathfrak{D}, \mathcal{A})$:

$$p(\mathbf{w}|\mathcal{A}) \sim \mathcal{N}(\boldsymbol{\mu}, \mathbf{A}_{\text{pr}}^{-1}), \tag{2.7}$$

где $\boldsymbol{\mu}, \mathbf{A}_{\text{pr}}^{-1}$ — вектор средних и матрица ковариации.

Приблизим интеграл (2.5) методом предложеном в [42]:

$$\mathcal{L}_{\mathcal{A}}(\mathfrak{D}, \mathcal{A}) = \log p(\mathfrak{D}|\mathcal{A}) =$$

$$= \int_{\mathbf{w} \in \mathbb{W}_{\mathcal{J}}} q(\mathbf{w}) \log \frac{p(\mathfrak{D}, \mathbf{w}|\mathcal{A})}{q(\mathbf{w})} d\mathbf{w} - \int_{\mathbf{w} \in \mathbb{W}_{\mathcal{J}}} q(\mathbf{w}) \log \frac{p(\mathbf{w}|\mathfrak{D}, \mathcal{A})}{q(\mathbf{w})} d\mathbf{w} \approx$$

$$\approx \int_{\mathbf{w} \in \mathbb{W}_{\mathcal{J}}} q(\mathbf{w}) \log \frac{p(\mathfrak{D}, \mathbf{w}|\mathcal{A})}{q(\mathbf{w})} d\mathbf{w} =$$

$$= \int_{\mathbf{w} \in \mathbb{W}_{\mathcal{J}}} q(\mathbf{w}) \log \frac{p(\mathbf{w}|\mathcal{A})}{q(\mathbf{w})} d\mathbf{w} + \int_{\mathbf{w} \in \mathbb{W}_{\mathcal{J}}} q(\mathbf{w}) \log p(\mathfrak{D}|\mathcal{A}, \mathbf{w}) d\mathbf{w} =$$

$$= \mathcal{L}_{\mathbf{w}}(\mathfrak{D}, \mathcal{A}, \mathbf{w}) + \mathcal{L}_E(\mathfrak{D}, \mathcal{A}). \tag{2.8}$$

Первое слагаемое формулы (2.8) — это сложность модели. Оно определяется расстоянием Кульбака-Лейблера:

$$\mathcal{L}_{\mathbf{w}}(\mathfrak{D}, \mathcal{A}, \mathbf{w}) = -D_{KL}\big(q(\mathbf{w})||p(\mathbf{w}|\mathcal{A})\big). \tag{2.9}$$

Второе слагаемое формулы (2.8) является матожиданием правдоподобия выборки $\mathcal{L}_{\mathfrak{D}}(\mathfrak{D}, \mathcal{A}, \mathbf{w})$. В данной работе оно является функцией ошибки:

$$\mathcal{L}_E(\mathfrak{D}, \mathcal{A}) = \mathsf{E}_{\mathbf{w}\sim q}\mathcal{L}_{\mathfrak{D}}(\mathbf{y}, \mathfrak{D}, \mathcal{A}, \mathbf{w}). \tag{2.10}$$

Требуется найти параметры, доставляющие минимум суммарному функционалу потерь $\mathcal{L}_{\mathcal{A}}(\mathfrak{D}, \mathcal{A}, \mathbf{w})$ из (2.8):

$$\hat{\mathbf{w}} = \underset{\mathcal{A}\subset\mathcal{J},\ \mathbf{w}\in\mathbb{W}_{\mathcal{A}}}{\arg\min} -\mathcal{L}_{\mathcal{A}}(\mathfrak{D}, \mathcal{A}, \mathbf{w}) =$$

$$= \underset{\mathcal{A}\subset\mathcal{J},\ \mathbf{w}\in\mathbb{W}_{\mathcal{A}}}{\arg\min} D_{KL}\big(q(\mathbf{w})||p(\mathbf{w}|\mathcal{A})\big) - \mathcal{L}_{\mathfrak{D}}(\mathfrak{D}, \mathcal{A}, \mathbf{w}). \tag{2.11}$$

2.3 Случайное удаление

Метод случайного удаления заключается в том, что случайным образом удаляется некоторый параметр w_ξ из множества активных параметров сети. Индекс параметра ξ из равномерного распределения случайная величина, предположительно доставляющая оптимум в (2.11).

$$\xi \sim \mathcal{U}(\mathcal{A}). \tag{3.1.1}$$

2.4 Оптимальное прореживание

Метод оптимального прореживания [44] использует вторую производную целевой функции (2.4) по параметрам для определения нерелевантных параметров. Рассмотрим функцию потерь $\mathcal{L}$ (2.4) разло-

женную в ряд Тейлора в некоторой окрестности вектора параметров $\mathbf{w}$:

$$\delta\mathcal{L} = \sum_{j\in\mathcal{A}} g_j \delta w_j + \frac{1}{2}\sum_{i,j\in\mathcal{A}} h_{ij}\delta w_i \delta w_j + O(||\delta\mathbf{w}||^3), \tag{3.2.1}$$

где δw_j — компоненты вектора $\delta\mathbf{w}$, g_j — компоненты вектора градиента $\nabla\mathcal{L}$, а h_{ij} — компоненты гессиана $\mathbf{H}$:

$$g_j = \frac{\partial\mathcal{L}}{\partial w_j}, \qquad h_{ij} = \frac{\partial^2\mathcal{L}}{\partial w_i \partial w_j}. \tag{3.2.2}$$

Задача является вычислительно сложной в силу размерности матрицы $\mathbf{H}$. Введем следующее предположение [44], о том что удаление нескольких параметров приводит к такому же изменению функции потерь $\mathcal{L}$, как и суммарное изменение при индивидуальном удалении:

$$\delta\mathcal{L} = \sum_{j\in\mathcal{A}} \delta\mathcal{L}_j, \tag{3.2.3}$$

где $\mathcal{A}$ — множество активных параметров, $\delta\mathcal{L}_j$ — изменение функции потерь, при удалении одного параметра $\mathbf{w}_j$.

В силу данного предположения будем рассматривать только диагональные элементы матрицы $\mathbf{H}$. После введенного предположения, выражение (3.2.1) принимает вид

$$\delta\mathcal{L} = \frac{1}{2}\sum_{j\in\mathcal{A}} h_{jj}\delta w_j^2, \tag{3.2.4}$$

Получаем следующую задачу оптимизации:

$$\xi = \arg\min_{j\in\mathcal{A}} h_{jj}\frac{w_j^2}{2}, \tag{3.2.5}$$

где ξ — индекс наименее релевантного, удаляемого параметра, предположительно доставляющая оптимум в (2.11).

2.5 Удаление неинформативных параметров с помощью вариационного вывода

Для удаления параметров в работе [48] предлагается удалить параметры, которые имеют максимальное отношение плотности $p(\mathbf{w}|\mathcal{A})$ априорной вероятности в нуле к плотности вероятности априорной вероятности в математическом ожидании параметра.
Для гауссовского распределения с диагональной матрицей ковариации получаем:

$$p_j(\mathbf{w}|\mathcal{A})(x) = \frac{1}{\sqrt{2\sigma_j^2}} \exp(-\frac{(x-\mu_j)^2}{2\sigma_j^2}). \tag{3.3.1}$$

Разделив плотность вероятности в нуле к плотности в математическом ожидание

$$\frac{p_j(\mathbf{w}|\mathcal{A})(0)}{p_j(\mathbf{w}|\mathcal{A})(\mu_j)} = \exp(-\frac{\mu_j^2}{2\sigma_j^2}), \tag{3.3.2}$$

Получаем следующую задачу оптимизации:

$$\xi = \arg\min_{j \in \mathcal{A}} \left| \frac{\mu_j}{\sigma_j} \right|, \tag{3.3.3}$$

где ξ — индекс наименее релевантного, удаляемого параметра.

2.6 Прореживание сети на основе метода Белсли

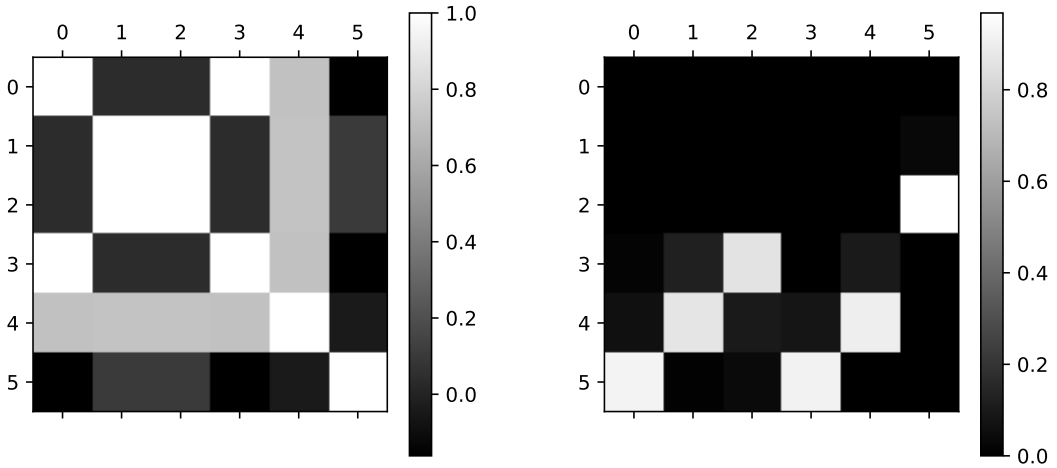
Предлагается метод основанный на модификации метода Белсли. Пусть $\mathbf{w}$ — вектор параметров доставляющий минимум функционалу потерь $\mathcal{L}$ на множестве $\mathbb{W}_{\mathcal{A}}$, а $\mathbf{A}_{\text{ps}}$ соответствующая ему ковариационная матрица.

Выполним сингулярное разложение матрицы

$$\mathbf{A}_{\text{ps}} = \mathbf{U}\mathbf{\Lambda}\mathbf{V}^{\mathsf{T}}. \tag{4.1}$$

Индекс обусловленности η_j определим как отношение максимального элемента к j-му элементу матрицы $\mathbf{\Lambda}$. Для нахождения мультиколлинеарных признаков требуется найти индекс ξ вида:

$$\xi = \arg\max_{j \in \mathcal{A}} \eta_j. \tag{4.2}$$



(a) Матрица ковариации

(b) Дисперсионные доли

Рис. 1: Илюстрация метода Белсли

Дисперсионный долевой коэффициент q_{ij} определим как вклад j-го признака в дисперсию i-го элемента вектора параметра $\mathbf{w}$:

$$q_{ij} = \frac{u_{ij}^2/\lambda_{jj}}{\sum_{j=1}^{n} u_{ij}^2/\lambda_{jj}}. \tag{4.3}$$

Большие значение дисперсионных долей указывают на наличие зависимости между параметрами. Находим долевые коэффициенты, которые вносят максимальный вклад в дисперсию параметра w_ξ:

$$\zeta = \arg\max_{j \in \mathcal{A}} q_{\xi j}. \tag{4.4}$$

Таблица 1: Илюстрация метода Белсли

η	q_1	q_2	q_3	q_4	q_5	q_6
1.0	$2 \cdot 10^{-17}$	$4 \cdot 10^{-17}$	$1 \cdot 10^{-16}$	$2 \cdot 10^{-17}$	$6 \cdot 10^{-17}$	$3 \cdot 10^{-4}$
1.5	$5 \cdot 10^{-17}$	$9 \cdot 10^{-17}$	$2 \cdot 10^{-16}$	$5 \cdot 10^{-17}$	$3 \cdot 10^{-20}$	$3 \cdot 10^{-2}$
3.3	$9 \cdot 10^{-18}$	$1 \cdot 10^{-17}$	$2 \cdot 10^{-17}$	$9 \cdot 10^{-18}$	$2 \cdot 10^{-19}$	$9 \cdot 10^{-1}$
$2 \cdot 10^{15}$	$1 \cdot 10^{-2}$	$1 \cdot 10^{-1}$	$8 \cdot 10^{-1}$	$2 \cdot 10^{-3}$	$9 \cdot 10^{-2}$	$1 \cdot 10^{17}$
$8 \cdot 10^{15}$	$6 \cdot 10^{-2}$	$8 \cdot 10^{-1}$	$9 \cdot 10^{-2}$	$8 \cdot 10^{-2}$	$9 \cdot 10^{-1}$	$2 \cdot 10^{17}$
$1 \cdot 10^{16}$	$\mathbf{9 \cdot 10^{-1}}$	$1 \cdot 10^{-2}$	$4 \cdot 10^{-2}$	$\mathbf{9 \cdot 10^{-1}}$	$1 \cdot 10^{-3}$	$5 \cdot 10^{-21}$

Параметр с индексом ζ определим как наименее релевантный параметр нейросети.

Проиллюстрируем принцип работы метода Белсли на примере. Рассмотрим данные порожденные следующим образом:

$$\mathbf{w} = \begin{bmatrix} \sin(x) \\ \cos(x) \\ 2{+}\cos(x) \\ 2{+}\sin(x) \\ \cos(x) + \sin(x) \\ x \end{bmatrix}$$

с матрицей ковариации на рис. 1.a, где $x \in [0.0, 0.02, ..., 20.0]$.

В табл. 1 приведены индексы обусловленности и соответствующие им дисперсионные доли, которые также изображены на рис. 1.b. Согласно этим данным, максимальный индекс обусловленности $\eta_6 = 1.2 \cdot 10^{16}$. Ему соответствуют максимальные дисперсионные доли признаков с индексами 1 и 4, которые, как видно из построения выборки, являются линейно зависимые.

3 Введение отношения порядка на множестве параметров аппроксимирующих моделей

3.1 Описание задачи

В данной работе предлагается метод введения отношения порядка на множестве параметров сложных параметрических моделей, таких как нейросеть. Рассматривается порядок, заданный при помощи ковариационной матрицы градиентов функции ошибки по параметрам модели [36]. В работе [28] предложен итерационный метод для поиска ковариационной матрицы градиентов. Данный итерационный метод интегрируется в градиентный метод оптимизации Adam [37].

Множество параметров упорядочивается по возрастанию дисперсии: от параметра с минимальной дисперсией до параметра с максимальной дисперсией градиента функции ошибки по соответствующему параметру модели. Предполагается, что малая дисперсия градиента указывает на то, что соответствующий параметр можно зафиксировать.

Для задания порядка на множестве параметров при помощи ковариационной матрицы вводится предположение о том, что фиксация параметров происходит в момент, когда все параметры модели находятся в некоторой окрестности локального минимума функции ошибки. Данное условие накладывается для корректного использования итерационного метода поиска ковариационной матрицы градиентов.

Заданный порядок на множестве параметров модели используется для фиксации тех параметров модели, которые оказываются предстоящими с точки зрения заданного порядка. Сначала фиксируются те параметры, которые имеют минимальную дисперсию градиента в окрестности локального минимума функции ошибки.

Для анализа свойств предложенного метода задания порядка на множестве параметров проводился вычислительный эксперимент. В качестве моделей рассматривались модели различной структурной

сложности: линейные модели, нейросетевые модели. Предложенный метод задания порядка сравнивается с методом, в котором порядок задан произвольным образом.

3.2 Формальная постановка задачи

Задана выборка:

$$\mathfrak{D} = \big\{ \big(\mathbf{x}_i, y_i\big) \big\}_{i=1}^{m}, \quad \mathbf{x}_i \in \mathbb{X} = \mathbb{R}^n, \quad y_i \in \mathbb{Y}, \tag{3.1}$$

где n — размерность признакового пространства, m — число объектов в выборке. Пространство ответов $\mathbb{Y} = \mathbb{R}$ в случае задачи регрессии и $\mathbb{Y} = \{1, \cdots, K\}$ в случае задачи классификации, где K — число классов.

Задано семейство моделей параметрических функций с наперед заданной структурой:

$$\begin{aligned} \mathfrak{F} &= \big\{ f\big(\mathbf{w}\big) : \mathbb{X} \to \mathbb{Y} \mid \mathbf{w} \in \mathbb{R}^p \big\}, \\ \mathbf{h}\big(\mathbf{w}, \mathbf{x}\big) &= \mathbf{W}_1 \boldsymbol{\sigma}\big(\mathbf{W}_2 \boldsymbol{\sigma}\big(\cdots \boldsymbol{\sigma}\big(\mathbf{W}_r \mathbf{x}\big) \cdots \big)\big), \\ f_{\text{cl}}\big(\mathbf{w}, \mathbf{x}\big) &= \arg \max_{j \in \big\{1, \cdots, K\big\}} \operatorname{softmax}\big(\mathbf{h}\big(\mathbf{w}, \mathbf{x}\big)\big)_j, \\ f_{\text{reg}}\big(\mathbf{w}, \mathbf{x}\big) &= \mathbf{h}\big(\mathbf{w}, \mathbf{x}\big), \end{aligned} \tag{3.2}$$

где p — размерность пространства параметров, r — число слоев нейросети, $\mathbf{w} = \operatorname{vec}[\mathbf{W}_1, \mathbf{W}_2, \cdots, \mathbf{W}_r]$, а $\boldsymbol{\sigma}$ — функция активации. В случае задачи регрессии структура модели имеет вид f_{reg}, а в случае классификации имеет вид f_{cl}. Задана функция потерь:

$$\begin{aligned} \mathcal{L}\big(\mathbf{w}, \mathfrak{D}\big) &= \frac{1}{m} \sum_{i=1}^{m} l\big(\mathbf{x}_i, y_i, \mathbf{w}\big), \\ l_{\text{reg}}\big(\mathbf{x}, y, \mathbf{w}\big) &= \big(y - f\big(\mathbf{w}, \mathbf{x}\big)\big)^2, \\ l_{\text{cl}}\big(\mathbf{x}, y, \mathbf{w}\big) &= -\sum_{j=1}^{K} \big([y = j] \ln \operatorname{softmax}_j\big(\mathbf{h}\big(\mathbf{w}, \mathbf{x}\big)\big)\big), \end{aligned} \tag{3.3}$$

где l_{reg} — это функция ошибки на одном элементе для задачи регрессии, l_{cl} — для задачи классификации. Оптимальный вектор параметров $\hat{\mathbf{w}}$ получим минимизацией функции потерь:

$$\hat{\mathbf{w}} = \arg\min_{\mathbf{w}\in\mathbb{R}^p} \mathcal{L}\big(\mathbf{w}, \mathfrak{D}\big). \tag{3.4}$$

3.3 Задание отношение порядка на множестве параметров

Для поиска оптимальных параметров модели используется градиентный метод оптимизации:

$$\mathbf{w}_t = \mathbf{w}_{t-1} + \Delta\mathbf{w}\big(\mathbf{g}_{S,t}, \mathbf{w}_{t-1}, \mathbf{w}_{t-2}, \cdots\big), \quad \mathbf{g}_{S,t} = \frac{\partial\mathcal{L}\big(\mathbf{w}_t, \mathbf{X}_S, \mathbf{Y}_S\big)}{\partial\mathbf{w}}, \tag{3.5}$$

где t — номер итерации, $\mathbf{g}_{S,t}$ — значение градиента на подвыборке размера S, $\Delta\mathbf{w}$ — приращение вектора параметров.

Порядок на множестве параметров модели задается при помощи ковариационной матрицы $\mathbf{C}$ градиентов функции ошибки $\mathcal{L}$ по параметрам модели $\mathbf{w}$. Для вычисления ковариационной матрицы $\mathbf{C}$ используется итерационная формула [28], которая вычисляется на каждой итерации (3.5) градиентного метода оптимизации параметров:

$$\mathbf{C}_t = \big(1 - \kappa_t\big)\mathbf{C}_{t-1} + \kappa_t\big(\mathbf{g}_{1,t} - \mathbf{g}_{S,t}\big)\big(\mathbf{g}_{1,t} - \mathbf{g}_{S,t}\big)^{\mathsf{T}}, \tag{3.6}$$

где t — номер итерации, $\mathbf{g}_{S,t}$ — значение градиента на подвыборке размера S, $\mathbf{g}_{1,t}$ — значение градиента на первом элементе подвыборки, $\kappa_t = \frac{1}{t}$ — параметр сглаживания, $\mathbf{C}_0$ инициализируются из равномерного распределения.

Пусть известно t_0 — число итераций, после которого все параметры находятся в некоторой локальной окрестности минимума, тогда, как показано в работе [28], матрица $\mathbf{C}_{t_0}$ аппроксимирует истинную ковариационную матрицу $\mathbf{C}$. Ковариационная матрица $\mathbf{C}_{t_0}$ используется для упорядочения параметров модели $\mathbf{w}_{t_0}$.

Пусть $\mathcal{I}$ — упорядоченный вектор индексов $[1, 2, \cdots, p]$. Обозначим $\mathcal{I}_{\mathbf{w}_{t_0}}$ вектор индексов, порядок которого задан при помощи ковариационной матрицы $\mathbf{C}_{t_0}$.

Например, если ковариационная матрица $\mathbf{C}_{t_0}$ имеет вид

$$\begin{bmatrix} 0{,}3 & 0 & 0 \\ 0 & 0{,}2 & 0 \\ 0 & 0 & 0{,}25 \end{bmatrix},$$

то вектор индексов $\mathcal{I}_{\mathbf{w}_{t_0}} = [3, 1, 2]$.

3.4 Фиксация параметров

Для фиксации параметров $\mathbf{w}_{t_0}$ при помощи вектора индексов $\mathcal{I}_{\mathbf{w}_{t_0}}$ используется бинарный вектор $\boldsymbol{\alpha}\big(k\big)$:

$$\alpha_i\big(k\big) = \begin{cases} 1, & \text{если } \mathcal{I}_{\mathbf{w}_{t_0}}[j] \leq k; \\ 0 & \text{иначе,} \end{cases} \tag{3.7}$$

где k — число фиксирующих параметров.

Учитывая (3.7), уравнение (3.5) приводится к виду

$$\mathbf{w}_t = \mathbf{w}_{t-1} + \boldsymbol{\alpha}\big(k\big) \cdot \Delta\mathbf{w}\big(\mathbf{g}_{S,t}, \mathbf{w}_{t-1}, \mathbf{w}_{t-2}, \cdots\big), \tag{3.8}$$

где t — номер итерации, $\mathbf{g}_{S,t}$ — значение градиента на подвыборке размера S, $\Delta\mathbf{w}$ — приращение вектора параметров. После умножения на бинарный вектор $\boldsymbol{\alpha}$ часть параметров не оптимизируется, что приводит к фиксации параметров.

4 Заключение

В рамках данного реферата приведен обзор существующих подходов для снижения сложности моделей глубокого обучения. Снижения сложности моделей глубокого обучения производится с целью улучшения интерпретируемости моделей глубокого обучения.

Рассмотрена проблема задания порядка на множестве параметров сложных аппроксимирующих моделей. Исследован метод задания порядка на основе анализа стохастических свойств градиента функции ошибки $\mathcal{L}$ по параметрам модели. Для задания порядка использовалась ковариационная матрица градиентов параметров $\mathbf{C}_{\eta_0}$, которая рассчитывается итеративно, в течение t_0 итераций градиентного метода параллельно оптимизации. Число итераций t_0 выбиралось заранее экспериментально. Отдельно стоит заметить, что данный метод позволяет упорядочивать параметры в процессе оптимизации параметров модели. Также рассмотрены методы оптимального прореживания, метод основаный на вариационном подходе, а также метод основанный на методе Белсли для удаления зависимых параметров модели. Все данные методы позволяет задать полный порядок на множестве параметров моделей глубокого обучения.

Полный порядок на множестве параметров позволяет выбирать архитектуры нейросетевых моделей ученика. Выбранные архитектуры рассматриваются в качестве модели ученика в методах дистилляции.

Список литературы

[1] *Alex Krizhevsky and Vinod Nair and Geoffrey Hinton* CIFAR-10 (Canadian Institute for Advanced Research) // `http://www.cs.toronto.edu/~kriz/cifar.html`

[2] *Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., Fei-Fei, L.* Imagenet: A large-scale hierarchical image database // IEEE conference on computer vision and pattern recognition, 2009. P. 248–255.

[3] *Huang, Zehao and Wang, Naiyan* Like What You Like: Knowledge Distill via Neuron Selectivity Transfer // arXiv e-prints, 2017.

[4] *Kui Ren and Tianhang Zheng and Zhan Qin and Xue Liu* Adversarial Attacks and Defenses in Deep Learning // Engineering, 2020. P. 346–360.

[5] *Alex Krizhevsky, Ilya Sutskever, Geoffrey Hinton* ImageNet Classification with Depp Convolutional Neural Networks // NIPS, 2012.

[6] *Karen Simonyan and Andrew Zisserman* Very Deep Convolutional Networks for Large-Scale Image Recognition // NIPS, 2014.

[7] *Vaswani A., Shazeer N., Parmar N., Uszkoreit J., Jones L., Gomez A., Kaiser L., Polosukhin I.* Attention Is All You Need // In Advances in Neural Information Processing Systems. 2017. V. 5. P. 6000–6010.

[8] *Devlin J., Chang M., Lee K., Toutanova K.* BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding // arXiv preprinted, 2018.

[9] *Tom B. Brown et al* GPT3: Language Models are Few-Shot Learners // arXiv preprinted, 2020.

[10] *Linting Xue and Noah Constant and Adam Roberts and Mihir Kale and Rami Al-Rfou and Aditya Siddhant and Aditya Barua and Colin Raffel.* mT5: A massively multilingual pre-trained text-to-text transformer // arXiv preprinted, 2021.

[11] *Yang, Ziqing and Cui, Yiming and Chen, Zhipeng and Che, Wanxiang and Liu, Ting and Wang, Shijin and Hu, Guoping* TextBrewer: An Open-Source Knowledge Distillation Toolkit for Natural Language Processing // Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: System Demonstrations. 2020. P. 9–16.

[12] *He K., Zhang X., Ren S., Sun J.* Deep Residual Learning for Image Recognition // Proc. of the IEEE Conference on Computer Vision and Pattern Recognition. Las Vegas, 2016. P. 770–778.

[13] *Бахтеев О. Ю., Стрижов В. В.* Выбор моделей глубокого обучения субоптимальной сложности // АиТ. 2018. № 8. С. 129–147.

[14] *Hinton G., Vinyals O., Dean J.* Distilling the Knowledge in a Neural Network // NIPS Deep Learning and Representation Learning Workshop. 2015.

[15] *LeCun Y., Cortes C., Burges C.* The MNIST dataset of handwritten digits, 1998. http://yann.lecun.com/exdb/mnist/index.html.

[16] *Vapnik V., Izmailov R.* Learning Using Privileged Information: Similarity Control and Knowledge Transfer // Journal of Machine Learning Research. 2015. No 16. P. 2023–2049.

[17] *Lopez-Paz D., Bottou L., Scholkopf B., Vapnik V.* Unifying Distillation and Privileged Information // In International Conference on Learning Representations. Puerto Rico, 2016.

[18] *Madala H., Ivakhnenko A.* Inductive Learning Algorithms for Complex Systems Modeling. Boca Raton: CRC Press Inc., 1994.

[19] *Xiao H., Rasul K.,Vollgraf R.* Fashion-MNIST: a Novel Image Dataset for Benchmarking Machine Learning Algorithms // arXiv preprint arXiv:1708.07747. 2017.

[20] *Wilson T., Kozareva Z., Nakov P., Rosenthal S., Stoyanov V., Ritter A.* SemEval-2013 Task 2: Sentiment Analysis in Twitter // Proceedings of the Seventh International Workshop on Semantic Evaluation (SemEval 2013). Atlanta, 2013. P. 312–320.

[21] *LeCun Y., Boser B., Denker J., Henderson D., Howard R., Hubbard W., Jackel L.* Backpropagation Applied to Handwritten Zip Code Recognition // Neural Computation. 1989. V. 1. No 4. P. 541–551.

[22] *Hochreiter S., Schmidhuber J.* Long short-term memory // Neural Computation. 1997. V. 9. No 8. P. 1735–1780.

[23] *Kingma D, Ba J.* Adam: A Method for Stochastic Optimization // arXiv preprint arXiv:1412.6980. 2014.

[24] *Graves A.* Practical Variational Inference for Neural Networks // Advances in Neural Information Processing Systems, 2011. Vol. 24. P. 2348–2356.

[25] *Vapnik V., Izmailov R.* Learning Using Privileged Information: Similarity Control and Knowledge Transfer // Journal of Machine Learning Research. 2015. No 16. P. 2023–2049.

[26] *Lopez-Paz D., Bottou L., Scholkopf B., Vapnik V.* Unifying Distillation and Privileged Information // In International Conference on Learning Representations. Puerto Rico, 2016.

[27] *Sutskever I., Vinyals O., Le Q.* Sequence to Sequence Learning with Neural Networks // Advances in Neural Information Processing Systems, 2014. Vol. 2. P. 3104–3112.

[28] *Li C., Chen C., Carlson D., Carin L.* Preconditioned Stochastic Gradient Langevin Dynamics for Deep Neural Networks //

Thirtieth AAAI Conference on Artificial Intelligence. — Phoenix, USA, 2016. P. 1788–1794.

[29] *Tibshirani R.* Regression shrinkage and selection via the Lasso // Journal of the Royal Statistical Society, 1996. Vol. 58. P. 267–288.

[30] *Zou H., Hastie T.* Regularization and variable selection via the Elastic Net // Journal of the Royal Statistical Society, 2005. Vol. 67. P. 301–320.

[31] *Srivastava N., Hinton G., Krizhevsky A., Sutskever I., Salakhutdinov R.* Dropout: A Simple Way to Prevent Neural Networks from Overfitting // Journal of Machine Learning Research, 2014. Vol. 15. P. 1929–1958.

[32] *Molchanov D., Ashukha A., Vetrov D.* Variational Dropout Sparsifies Deep Neural Networks // 34th International Conference on Machine Learning. — Sydney, Australia, 2017. Vol. 70. P. 2498–2507.

[33] *LeCun Y., Denker J., Solla S.* Optimal Brain Damage // Advances in Neural Information Processing Systems, 1989. Vol. 2. P. 598–605.

[34] *Грабовой А. В., Бахтеев О. Ю., Стрижов В. В.* Определение релевантности параметров нейросети // Информатика и ее применения, 2019. Т. 13. Вып. 2. С. 62–70.

[35] *Грабовой А. В., Бахтеев О. Ю., Стрижов В. В.* Введение отношения порядка на множестве параметров аппроксимирующих моделей // Информатика и ее применения, 2019. Т. 14. Вып. 2. С. 58–65.

[36] *Mandt S., Hoffman M., Blei D.* Stochastic Gradient Descent as Approximate Bayesian Inference // Journal Of Machine Learning Research, 2017. Vol. 18. P. 1–35.

[37] *Kingma D., Ba L.* Adam: A Method for Stochastic Optimization // 3rd International Conference on Learning Representations. — San Diego, USA, 2015.

[38] *Harrison D., Rubinfeld D.* Hedonic prices and the demand for clean air // Journal of Environmental Economics and Management, 1991. Vol. 5. P. 81–102.

[39] *LeCun Y., Cortes C., Burges C.* The MNIST dataset of handwritten digits, 1998. `http://yann.lecun.com/exdb/mnist/index.html`

[40] *Maclaurin D., Duvenaud D., Adams R.* Gradient-based Hyperparameter Optimization Through Reversible Learning // Proceedings of the 32th International Conference on Machine Learning, 2015. Vol. 37. P. 2113–2122.

[41] *Luketina J., Berglund M., Raiko T., Greff K.* Scalable Gradient-based Tuning of Continuous Regularization Hyperparameters // Proceedings of the 33th International Conference on Machine Learning, 2016. Vol. 48. P. 2952–2960.

[42] *Bishop C.* Pattern Recognition and Machine Learning, 2006. Pp. 396.

[43] *Neychev R., Katrutsa A., Strijov V.* Robust selection of multicollinear features in forecasting // Factory Laboratory, 2016. Vol. 82. P. 68–74.

[44] *LeCun Y., Denker J., Solla S.* Optimal Brain Damage // Advances in Neural Information Processing Systems, 1989. P. 598–605.

[45] *Molchanov D., Ashukha A., Vetrov D.* Variational Dropout Sparsifies Deep Neural Networks // Proceedings of the 34th International Conference on Machine Learning, 2017. Vol. 70. P. 2498–2507.

[46] *Neal A., Radford M.* Bayesian Learning for Neural Networks, 1995.

[47] *Sutskever I., Vinyals O., Le Q.* Sequence to Sequence Learning with Neural Networks, 2014. Vol. 2. P. 3104–3112.

[48] *Graves A.* Practical Variational Inference for Neural Networks, 2011. P. 2348–2356.

[49] *Louizos C., Ullrich K., Welling M.* Bayesian Compression for Deep Learning, 2017. P. 3288–3298.