© 2022 г. М. ГОРПИНИЧ
(Московский физико-технический институт (государственный университет)),
О.Ю. БАХТЕЕВ, канд. физ.-мат. наук
(Вычислительный центр имени А.А. Дородницына Федерального исследовательского центра «Информатика и управление» Российской академии наук),
В.В. СТРИЖОВ, д-р физ.-мат. наук
(Вычислительный центр имени А.А. Дородницына Федерального исследовательского центра «Информатика и управление» Российской академии наук)

ГРАДИЕНТНЫЕ МЕТОДЫ ОПТИМИЗАЦИИ МЕТАПАРАМЕТРОВ В ЗАДАЧЕ ДИСТИЛЛЯЦИИ ЗНАНИЙ[1]

В работе исследуется задача дистилляции моделей глубокого обучения. Дистилляция знаний — это задача оптимизации метапараметров, в которой происходит перенос информации модели более сложной структуры, называемой моделью-учителем, в модель более простой структуры, называемой моделью-учеником. В работе предлагается обобщение задачи дистилляции на случай оптимизации метапараметров градиентными методами. Метапараметрами являются параметры оптимизационной задачи дистилляции. В качестве функции потерь для такой задачи выступает сумма слагаемого классификации и кросс-энтропии между ответами модели-ученика и модели-учителя. Назначение оптимальных метапараметров в функции потерь дистилляции является вычислительно сложной задачей. Исследуются свойства оптимизационной задачи с целью предсказания траектории обновления метапараметров. Проводится анализ траектории градиентной оптимизации метапараметров и предсказывается их значение с помощью линейных функций. Предложенный подход проиллюстрирован с помощью вычислительного эксперимента на выборках CIFAR-10 и Fashion-MNIST, а также на синтетических данных.

Ключевые слова: Машинное обучение, Дистилляция знаний, Оптимизация метапараметров, Градиентная оптимизация, Назначение метапараметров.

1. Введение

В работе рассматривается задача дистилляции моделей глубокого обучения. Оптимизация модели глубокого обучения является вычислительно сложной задачей [12]. В работе исследуется частный случай задачи оптимизации, называемый дистилляцией знаний. Он позволяет использовать одновременно обучающую выборку и информацию, содержащуюся в предобученных моделях. *Дистилляцией знаний* [5] назовем задачу оптимизации параметров модели, в которой учитывается не

[1]Работа выполнена при поддержке Научной академической стипендии имени К. В. Рудакова

Таблица 1: Сложность различных методов оптимизации метапараметров и гиперпараметров. Здесь $|\mathbf{w}|$ является числом параметров модели, $|\boldsymbol{\lambda}|$ — числом метапараметров, r — это количество запусков стохастических методов оптимизации, s — сложность порождения из вероятностных моделей.

Метод	**Тип метода оптимизации**	**Сложность**
Случайный поиск [2]	Стохастический	$O(r \cdot \|\mathbf{w}\|)$
Основанный на вероятностных моделях [3]	Стохастический	$O\left(r \cdot (\|\mathbf{w}\| + s)\right)$
Жадный градиентный [8]	Градиентный	$O(\|\mathbf{w}\| \cdot \|\boldsymbol{\lambda}\|)$
Жадный градиентный с разностной аппроксимацией [7]	Градиентный	$O(\|\mathbf{w}\| + \|\boldsymbol{\lambda}\|)$

только информация, содержащаяся в исходной выборке, но также и информация, содержащаяся в *модели-учителе*. Модель-учитель имеет высокую сложность. В ней содержится информация о выборке, а также о распределениях параметров модели, перенос которых будет осуществлен. Модель более простой структуры, называемая *моделью-учеником*, оптимизируется путем переноса знаний модели-учителя.

Исследуется процедура оптимизации метапараметров в задаче дистилляции знаний. *Метапараметрами* являются параметры оптимизационной задачи. Корректное назначение метапараметров может существенно повлиять на качество итоговой модели [11]. В отличие от [11, 9], в данной работе учитывается различие между *гиперпараметрами*, вероятностными параметрами априорного распределения [4] и метапараметрами. Несмотря на количество методов оптимизации метапараметров и гиперпараметров, использующихся в глубоком обучении, таких как случайный поиск [2] или модели, основанные на использовании вероятностных моделей [3], во многих подходах предлагается последовательно порождать случайное значение метапараметров и оценивать качество модели, обученной при данных значениях гиперпараметров. Данный подход может не подойти в случае обучения моделей, требующих значительных временных затрат для обучения. В Таблице 1 содержатся сложности различных подходов к оптимизации метапараметров. Видно, что в случае, если оптимизация параметров занимает значительное время, подходы, требующие несколько запусков оптимизации являются неэффективными.

Предлагается рассматривать задачу оптимизации метапараметров как двухуровневую задачу оптимизации. На первом уровне оптимизируются параметры модели, на втором — метапараметры [8, 1, 9]. Жадный градиентный метод для решения двухуровневой задачи описан в [8]. В [1] проанализированы различные градиентные методы и случайный поиск. В данной работе анализируется подход к оптимизации и предсказанию метапараметров, полученных после применения градиентных методов. Из Таблицы 1 можно увидеть, что для больших задач предпочтительны градиентные методы оптимизации метапараметров. Тем не менее, даже с применением жадного алгоритма оптимизации метапараметров с разностной аппрокисимацией, оптимизация метапараметров становится значительно требовательнее к вычислительным ресурсам, что было продемонстрировано в работе [7]. Для уменьшения затрат на оп-

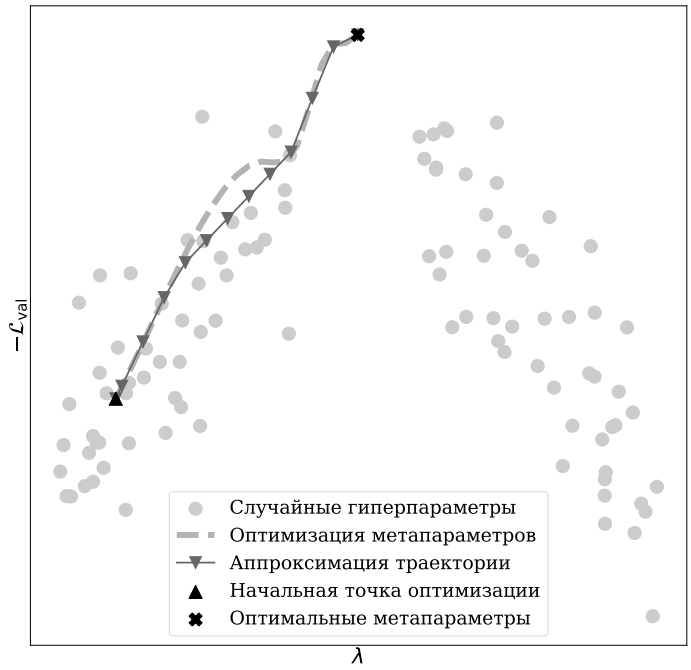


Рис. 1: Схема работы предложенного метода: вместо непосредственной оптимизации значений метапараметра λ предлагается аппроксимировать траекторию оптимизации с помощью линейных моделей для достижения минимума функции потерь на валидационной части выборки $\mathcal{L}_{\text{val}}$. Случайные метапараметры не являются точками минимума функции $\mathcal{L}_{\text{val}}$ и доставляют субоптимальное качество модели.

тимизацию в настоящей работе проводится анализ траектории оптимизации метапараметров и предсказывается ее значение с помощью линейных моделей. Этот метод проиллюстрирован на Рис. 1. Данный метод оценивается и сравнивается с другими методами оптимизации метапараметров на выборках изображений CIFAR-10 [6], Fashion-MNIST [14] и синтетической выборке.

2. Постановка задачи

Решается задача классификации вида:

$$\mathfrak{D} = \{(\mathbf{x}_i, \mathbf{y}_i)\}_{i=1}^{m},\ \mathbf{x}_i \in \mathbb{R}^n,\ \mathbf{y}_i \in \mathbb{Y} = \{\mathbf{e}_k | k = \overline{1, K}\},$$

где $\mathbf{e}_k$ — k-й столбец единичной матрицы, $\mathbf{y}_i$ — вектор с единицей на месте класса $\mathbf{x}_i$.

Разделим выборку на два подмножества $\mathfrak{D}$: $\mathfrak{D} = \mathfrak{D}_{\text{train}} \sqcup \mathfrak{D}_{\text{val}}$. Подмножество $\mathfrak{D}_{\text{train}}$ будем использовать для оптимизации параметров модели, а подмножество $\mathfrak{D}_{\text{val}}$ — для оптимизации метапараметров.

Рассмотрим модель-учителя $\mathbf{f}(\mathbf{x})$, которая была обучена на выборке $\mathfrak{D}_{\text{train}}$. Оптимизируем модель-ученика $\mathbf{g}(\mathbf{x}, \mathbf{w})$, $\mathbf{w} \in \mathbb{R}^s$ путем переноса знаний модели-учителя.

Определим данную задачу формально.

Определение 1. Пусть функция $D : \mathbb{R}^s \to \mathbb{R}_+$ задает расстояние между моделями $\mathbf{g}$ и $\mathbf{f}$. Назовем D-дистилляцией модели-ученика такую задачу оптимизации параметров модели-ученика, которая минимизирует функцию D.

Определим функцию потерь $\mathcal{L}_{\text{train}}$, которая учитывает перенос знаний от модели $\mathbf{f}$ к модели $\mathbf{g}$:

$$\mathcal{L}_{\text{train}}(\mathbf{w}, \boldsymbol{\lambda}) = -\lambda_1 \sum_{(\mathbf{x},\mathbf{y})\in\mathfrak{D}_{\text{train}}} \underbrace{\sum_{k=1}^{K} y_k \log \frac{e^{\mathbf{g}(\mathbf{x},\mathbf{w})_k}}{\sum\limits_{j=1}^{K} e^{\mathbf{g}(\mathbf{x},\mathbf{w})_j}}}_{\text{слагаемое классификации}}$$

$$- (1-\lambda_1) \sum_{(\mathbf{x},y)\in\mathfrak{D}_{\text{train}}} \underbrace{\sum_{k=1}^{K} \frac{e^{\mathbf{f}(\mathbf{x})_k/T}}{\sum\limits_{j=1}^{K} e^{\mathbf{f}(\mathbf{x})_j/T}} \log \frac{e^{\mathbf{g}(\mathbf{x},\mathbf{w})_k/T}}{\sum\limits_{j=1}^{K} e^{\mathbf{g}(\mathbf{x},\mathbf{w})_j/T}}}_{\text{слагаемое дистилляции}},$$

где y_k — это k-я компонента вектора ответов, T — параметр температуры в задаче дистилляции. Температура T имеет следующие свойства:

1) если $T \to 0$, то получаем единичный вектор $\left\{ \frac{e^{\mathbf{g}(\mathbf{x},\mathbf{w})_k/T}}{\sum\limits_{j=1}^{K} e^{\mathbf{g}(\mathbf{x},\mathbf{w})_j/T}} \right\}_{k=1}^{K}$;

2) если $T \to \infty$, то получаем вектор с равными вероятностями.

Покажем, что оптимизация $\mathcal{L}_{\text{train}}$ является D-дистилляцией при $\lambda_1 = 0$.

Предложение 1. Если $\lambda_1 = 0$, то оптимизация функции потерь (1)*, является D-дистилляцией с $D = D_{KL}\left(\sigma\left(\mathbf{f}(\mathbf{x})/T\right), \sigma\left(\mathbf{g}(\mathbf{x},\mathbf{w})/T\right)\right)$, где σ — это функция softmax $= \frac{e^{x_i}}{\sum\limits_{j=1}^{K} e^{x_j}}$, D_{KL} — дивергенция Кульбака-Лейблера.*

Доказательство. При $\lambda_1 = 0$ имеем:

$$(1)\quad \mathcal{L}_{\text{train}}(\mathbf{w}, \boldsymbol{\lambda}) = \sum_{(\mathbf{x},y)\in\mathfrak{D}_{\text{train}}} \sum_{k=1}^{K} \frac{e^{\mathbf{f}(\mathbf{x})_k/T}}{\sum\limits_{j=1}^{K} e^{\mathbf{f}(\mathbf{x})_j/T}} \log \frac{e^{\mathbf{g}(\mathbf{x},\mathbf{w})_k/T}}{\sum\limits_{j=1}^{K} e^{\mathbf{g}(\mathbf{x},\mathbf{w})_j/T}}$$

$$= D_{KL}\left(\sigma(\mathbf{f}(\mathbf{x})/T), \sigma(\mathbf{g}(\mathbf{x},\mathbf{w})/T)\right) - C.$$

Получаем, что $\mathcal{L}_{\text{train}}(\mathbf{w}, \boldsymbol{\lambda})$ равняется $D_{KL}\left(\sigma(\mathbf{f}(\mathbf{x})/T), \sigma(\mathbf{g}(\mathbf{x},\mathbf{w})/T)\right)$ с точностью до константы C, не влияющей на оптимизацию. Константа является энтропией от $\sigma(\mathbf{f}(\mathbf{x})/T)$. Функция $D_{KL}\left(\sigma\left(\mathbf{f}/T\right), \sigma\left(\mathbf{g}/T\right)\right)$ определяет расстояние между логитами модели $\mathbf{f}$ и модели $\mathbf{g}$. Получаем, что определение D-дистилляции выполняется.

Определим множество метапараметров $\boldsymbol{\lambda}$ как вектор, компонентами которого являются коэффициент λ_1 перед слагаемыми в $\mathcal{L}_{\text{train}}$ и температура T:

$$\boldsymbol{\lambda} = [\lambda_1, T].$$

Определим двухуровневую задачу:

$$\hat{\boldsymbol{\lambda}} = \arg\min_{\boldsymbol{\lambda}\in\mathbb{R}^2} \mathcal{L}_{\text{val}}(\hat{\mathbf{w}}, \boldsymbol{\lambda}), \tag{2}$$

$$\hat{\mathbf{w}} = \arg\min_{\mathbf{w}\in\mathbb{R}^s} \mathcal{L}_{\text{train}}(\mathbf{w}, \boldsymbol{\lambda}), \tag{3}$$

где $\mathcal{L}_{\text{val}}$ — это функция потерь на валидации:

$$\mathcal{L}_{\text{val}}(\mathbf{w}, \boldsymbol{\lambda}) = -\sum_{(\mathbf{x},y)\in\mathfrak{D}_{\text{val}}} \sum_{k=1}^{K} y^k \log \frac{e^{\mathbf{g}(\mathbf{x},\mathbf{w})_k/T_{\text{val}}}}{\sum\limits_{j=1}^{K} e^{\mathbf{g}(\mathbf{x},\mathbf{w})_j/T_{\text{val}}}},$$

метапараметр T_{val} определяет температуру в валидационной функции потерь. Его значение выбрано вручную и не является предметом оптимизации.

3. Градиентная оптимизация метапараметров

Одним из методов оптимизации метапараметров является использование градиентных методов. Ниже приведена схема их применения и подход к оптимизации траектории метапараметров.

Определение 2. Определим оператор оптимизации как алгоритм U, который выбирает вектор параметров модели $\mathbf{w}'$ используя значения параметров на предыдущем шаге $\mathbf{w}$.

Оптимизируем параметры $\mathbf{w}$ используя η шагов оптимизации:

$$\hat{\mathbf{w}} = U \circ U \circ \cdots \circ U(\mathbf{w}_0, \boldsymbol{\lambda}) = U^{\eta}(\mathbf{w}_0, \boldsymbol{\lambda}),$$

где $\mathbf{w}_0$ — начальное значение вектора параметров $\mathbf{w}$, $\boldsymbol{\lambda}$ — множество метапараметров.

Переформулируем оптимизационную задачу, используя определение оператора U:

$$\hat{\boldsymbol{\lambda}} = \arg\min_{\boldsymbol{\lambda}\in\mathbb{R}^2} \mathcal{L}_{\text{val}}\big(U^{\eta}(\mathbf{w}_0, \boldsymbol{\lambda})\big).$$

Решим оптимизационную задачу (2) и (3) с помощью оператора градиентного спуска:

$$U(\mathbf{w}, \boldsymbol{\lambda}) = \mathbf{w} - \gamma\nabla\mathcal{L}_{\text{train}}(\mathbf{w}, \boldsymbol{\lambda}),$$

где γ — длина шага градиентного спуска. Для оптимизации метапараметров используется жадный градиентный метод, который зависит только от значения параметров $\mathbf{w}$ на предыдущем шаге. На каждой итерации получим следующее значение метапараметров:

$$\boldsymbol{\lambda}' = \boldsymbol{\lambda} - \gamma_{\boldsymbol{\lambda}} \nabla_{\boldsymbol{\lambda}} \mathcal{L}_{\text{val}}(U(\mathbf{w}, \boldsymbol{\lambda}), \boldsymbol{\lambda}) = \boldsymbol{\lambda} - \gamma_{\boldsymbol{\lambda}} \nabla_{\boldsymbol{\lambda}} \mathcal{L}_{\text{val}}(\mathbf{w} - \gamma \nabla \mathcal{L}_{\text{train}}(\mathbf{w}, \boldsymbol{\lambda}), \boldsymbol{\lambda}). \tag{4}$$

В данной работе используется численная разностная аппроксимация для данной процедуры оптимизации [7]:

$$\frac{d\mathcal{L}_{\text{val}}(\mathbf{w}', \boldsymbol{\lambda})}{d\boldsymbol{\lambda}} = \nabla_{\boldsymbol{\lambda}} \mathcal{L}_{\text{val}}(\mathbf{w}', \boldsymbol{\lambda}) - \gamma \nabla^2_{\boldsymbol{\lambda}, \mathbf{w}'} \mathcal{L}_{\text{val}}(\mathbf{w}', \boldsymbol{\lambda}) \nabla_{\mathbf{w}'} \mathcal{L}_{\text{val}}(\mathbf{w}', \boldsymbol{\lambda}),$$

$$\nabla^2_{\boldsymbol{\lambda}, \mathbf{w}'} \mathcal{L}_{\text{val}}(\mathbf{w}', \boldsymbol{\lambda}) \nabla_{\mathbf{w}'} \mathcal{L}_{\text{val}}(\mathbf{w}', \boldsymbol{\lambda}) \approx \frac{\nabla_{\boldsymbol{\lambda}} \mathcal{L}_{\text{val}}(\mathbf{w}^+, \boldsymbol{\lambda}) - \nabla_{\boldsymbol{\lambda}} \mathcal{L}_{\text{val}}(\mathbf{w}^-, \boldsymbol{\lambda})}{2\varepsilon},$$

$$\boldsymbol{\lambda}' \approx \boldsymbol{\lambda} - \gamma_{\boldsymbol{\lambda}} \nabla_{\boldsymbol{\lambda}} \mathcal{L}_{\text{val}}(\mathbf{w}', \boldsymbol{\lambda}) + \gamma \frac{\nabla_{\boldsymbol{\lambda}} \mathcal{L}_{\text{val}}(\mathbf{w}^+, \boldsymbol{\lambda}) - \nabla_{\boldsymbol{\lambda}} \mathcal{L}_{\text{val}}(\mathbf{w}^-, \boldsymbol{\lambda})}{2\varepsilon}$$

где $\mathbf{w}' = \mathbf{w} - \gamma \nabla \mathcal{L}_{\text{train}}(\mathbf{w}, \boldsymbol{\lambda})$, $\mathbf{w}^{\pm} = \mathbf{w}' \pm \varepsilon \nabla_{\mathbf{w}'} \mathcal{L}_{\text{val}}(\mathbf{w}', \boldsymbol{\lambda})$, ε — некоторая заданная константа.

Для дальнейшего уменьшения стоимости оптимизации предлагается аппроксимировать траекторию оптимизации метапараметров. Траектория предсказывается с помощью линейных моделей, которые используются периодически после заданного числа итераций e_1. После этого линейная модель используется для предсказания метапараметров на протяжении e_2 итераций:

$$\boldsymbol{\lambda}' = \boldsymbol{\lambda} + \mathbf{c}^\top \begin{pmatrix} z \\ 1 \end{pmatrix}, \tag{5}$$

где $\mathbf{c}$ — это вектор параметров линейной модели, оптимизированный с помощью метода наименьших квадратов, z — число итераций оптимизации. Алгоритм предложенного метода приведен на Рис. 2.

Диаграмма на Рис. 3 описывает полученный метод оптимизации. Параметры модели оптимизируются на первом уровне двухуровневой оптимизационной задачи с помощью подмножества $\mathfrak{D}_{\text{train}}$ и функции потерь $\mathcal{L}_{\text{train}}$. Метапараметры оптимизируются на втором уровне с помощью подмножества $\mathfrak{D}_{\text{val}}$ и функции потерь $\mathcal{L}_{\text{val}}$. На протяжении e_1 итераций метапараметры оптимизируются с помощью метода стохастического градиентного спуска. На протяжении e_2 итераций предсказываются с помощью линейных моделей.

Следующая теорема доказывает корректность предложенной аппроксимации для простого случая: когда параметры $\mathbf{w}$ модели $\mathbf{g}$ достигли оптимума задачи (3), гессиан $\mathbf{H} = \nabla^2_{\mathbf{w}} \mathcal{L}_{\text{train}}$ является единичной матрицей, и оптимизация метапараметров ведется в области, в которой градиент метапараметров можно аппроксимировать константой. Отметим, что в общем случае, данные условия при оптимизации моделей глубокого обучения не выполняются. В работах [8, 13] было показано, что использование методов нормализации промежуточных представлений выборки под действием нелинейных функций, входящих в модель глубокого обучения, приближает гессиан функции потерь к единичному. Анализ качества градиентной оптимизации метапараметров для случая, когда параметры модели не достигли оптимума, приведен в [11].

Algorithm 1 Оптимизация метапараметров

Require: число e_1 итераций с использованием градиентной оптимизации
Require: число e_2 итераций с предсказанием $\boldsymbol{\lambda}$ линейными моделями

1: **while** нет сходимости **do**
2: Оптимизация $\boldsymbol{\lambda}$ и $\mathbf{w}$ на протяжении e_1 итераций, решая двухуровневую задачу
3: **traj** =траектория ($\nabla\boldsymbol{\lambda}$) изменяется во время оптимизации;
4: Положим $\mathbf{z} = [1, \ldots, e_1]^\mathsf{T}$
5: Оптимизация $\mathbf{c}$ с помощью МНК:

$$\hat{\mathbf{c}} = \arg\min_{\mathbf{c}\in\mathbb{R}^2} ||\mathbf{traj} - \mathbf{z} \cdot c_1 + c_2||_2^2$$

6: Оптимизация $\mathbf{w}$ и предсказание $\boldsymbol{\lambda}$ на протяжении e_2 итераций с помощью линейной модели с параметрами $\mathbf{c}$.
7: **end while**

Рис. 2: Алгоритм для предложенного метода.

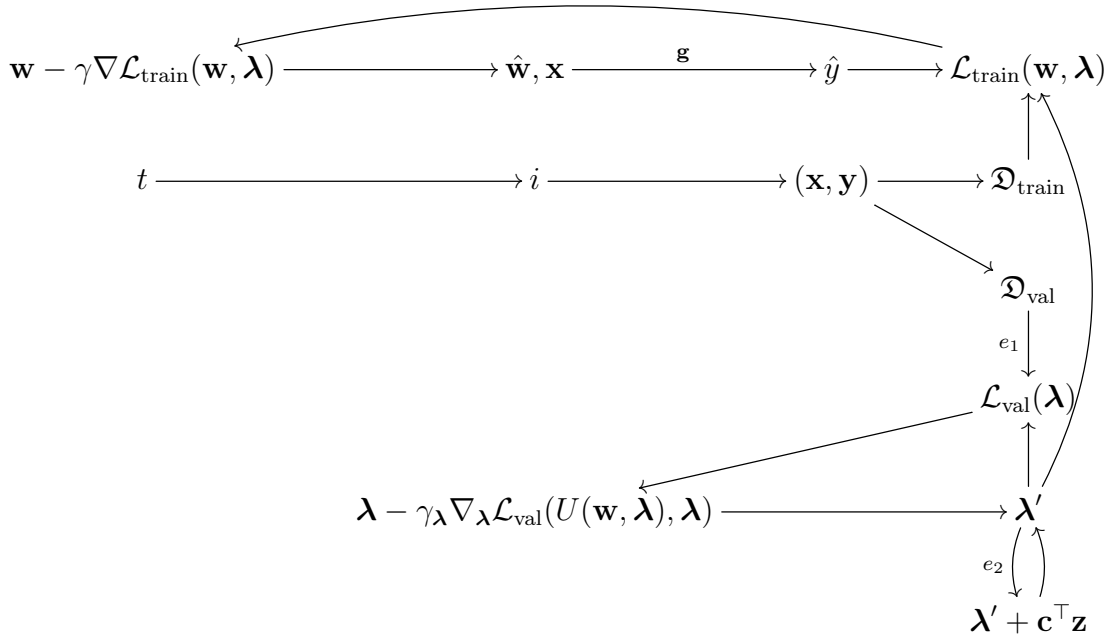


Рис. 3: Схема оптимизации метапараметров.

Теорема 1. Если функция $\mathcal{L}_{train}(\mathbf{w}, \boldsymbol{\lambda})$ является гладкой и выпуклой, и ее гессиан $\mathbf{H} = \nabla^2_{\mathbf{w}}\mathcal{L}_{\text{train}}$ является единичной матрицей, $\mathbf{H} = \mathbf{I}$, а также если параметры $\mathbf{w}$ равны $\mathbf{w}^*$, где $\mathbf{w}^*$ — точка локального минимума для текущего значения $\boldsymbol{\lambda}$, тогда жадный алгоритм (4) находит оптимальное решение двухуровневой задачи. Если существует область $\mathcal{D} \in \mathbb{R}^2$ в пространстве метапараметров, такая что градиент метапараметров может быть аппроксимирован константой, то оптимизация является линейной по метапараметрам.

Доказательство. В работе [11] была выведена формула для $\nabla_{\boldsymbol{\lambda}}\mathcal{L}_{\text{val}} = \nabla_{\boldsymbol{\lambda}}\mathcal{L}_{\text{val}}(U(\mathbf{w}, \boldsymbol{\lambda}))$ в случае, если $\mathcal{L}_{\text{train}}(\mathbf{w}, \boldsymbol{\lambda})$ является гладкой и выпуклой, и найдена $\mathbf{w}^*$ — точка локального минимума для текущего значения $\boldsymbol{\lambda}$:

$$\nabla_{\boldsymbol{\lambda}}\mathcal{L}_{\text{val}}(\boldsymbol{\lambda}) = \nabla_{\boldsymbol{\lambda}}\mathcal{L}_{\text{val}} - (\nabla^2_{\mathbf{w},\boldsymbol{\lambda}}\mathcal{L}_{\text{train}})^\top(\nabla^2_{\mathbf{w}}\mathcal{L}_{\text{train}})^{-1}\nabla_{\mathbf{w}}\mathcal{L}_{\text{val}}.$$

Эта формула упрощается исключением первого слагаемого, так как функция $\mathcal{L}_{\text{val}}$ явно не зависит от метапараметров:

$$\nabla_{\boldsymbol{\lambda}}\mathcal{L}_{\text{val}}(\boldsymbol{\lambda}) = -(\nabla^2_{\mathbf{w},\boldsymbol{\lambda}}\mathcal{L}_{\text{train}})^\top(\nabla^2_{\mathbf{w}}\mathcal{L}_{\text{train}})^{-1}\nabla_{\mathbf{w}}\mathcal{L}_{\text{val}}.$$

Если $\nabla^2_{\mathbf{w}}\mathcal{L}_{\text{train}}$ равен единичной матрице, то жадный алгоритм дает оптимум двухуровневой задачи в том случае, если его шаг выражается следующей формулой [8]:

$$\boldsymbol{\lambda}_{t+1} = \boldsymbol{\lambda}_t + \eta_1(\nabla^2_{\mathbf{w},\boldsymbol{\lambda}}\mathcal{L}_{\text{train}})^\top\nabla_{\mathbf{w}}\mathcal{L}_{\text{val}}.$$

Также заменим $\nabla^2_{\mathbf{w}}\mathcal{L}_{\text{train}}$ на единичную матрицу.

Вернемся к упрощенной формуле градиента:

$$\nabla_{\boldsymbol{\lambda}}\mathcal{L}_{\text{val}}(\boldsymbol{\lambda}) = -(\nabla^2_{\mathbf{w},\boldsymbol{\lambda}}\mathcal{L}_{\text{train}})^\top\nabla_{\mathbf{w}}\mathcal{L}_{\text{val}}.$$

Предположим, что существует область $\mathcal{D}$, в которой $\nabla_{\boldsymbol{\lambda}}\mathcal{L}_{\text{val}}(\boldsymbol{\lambda})$ равен константному вектору

$$\nabla_{\boldsymbol{\lambda}}\mathcal{L}_{\text{val}}(\boldsymbol{\lambda}) \approx \begin{pmatrix} a_1 \\ a_2 \end{pmatrix}.$$

Тогда в $\mathcal{D}$ шаг оптимизации можно представить в виде

$$\boldsymbol{\lambda}_{t+1} = \boldsymbol{\lambda}_t - \gamma_{\boldsymbol{\lambda}} \begin{pmatrix} a_1 \\ a_2 \end{pmatrix},$$

и имеет вид, аналогичный (5).

4. Вычислительный эксперимент

Целью эксперимента является оценка качества предложенного метода дистилляции и анализ полученных моделей и их метапараметров. Метод оценивался на синтетической выборке, а также выборках CIFAR-10 и Fashion-MNIST.На выборке CIFAR-10 было проведено два вида экспериментов: на всей выборке, $|\mathfrak{D}_{\text{train}}| = 50000$, и на уменьшенной обучающей выборке, $|\mathfrak{D}_{\text{train}}| = 12800$.

Были проанализированы следующие методы оптимизации метапараметров:

1) оптимизация без дистилляции;

2) оптимизация со случайной инициализацией метапараметров. Метапараметры порождаются из равномерного распределения

$$\lambda_1 \sim \mathcal{U}(0;1), \quad T \sim \mathcal{U}(0.1, 10).$$

3) оптимизация с "наивным" назначением метапараметров:

$$\lambda_1 = 0.5, T = 1;$$

4) градиентная оптимизация;

5) предложенный метод с $e_1 = e_2 = 10$.

6) оптимизация с помощью вероятностной модели. Для данного типа оптимизации использовалась библиотека hyperopt [3], в которой реализована оптимизация с помощью метода парзеновского окна. Для этого метода проводилось 5 запусков перед итоговым предсказанием метапараметров.

Для методов 1-3 использовалась вся обучающая выборка $\mathfrak{D}$. Для методов 4-6 выборка разбивалась на обучение, валидацию, контроль $\mathfrak{D} = \mathfrak{D}_{\text{train}} \sqcup \mathfrak{D}_{\text{val}} \sqcup \mathfrak{D}_{\text{test}}$.

В качестве внешнего критерия качества была использована метрика accuracy:

$$\text{accuracy} = \frac{1}{m} \sum_{i=1}^{m} [\mathbf{g}(\mathbf{x}_i, \mathbf{w}) = y_i],$$

Для всех экспериментов порождение начальных значений метапараметров происходило следующим образом:

$$\lambda_1 \sim \mathcal{U}(0,1), \quad \log_{10} T \sim \mathcal{U}(-1,1).$$

Для каждого эксперимента проводилось 10 запусков, затем результаты усреднялись. Код эксперимента доступен в [15].

Итоговые результаты представлены в Таблице 2. Зависимость точности от номера итерации на синтетической выборке и уменьшенной версии CIFAR-10 изображена на Рис. 4.

4.1. Эксперимент на синтетической выборке

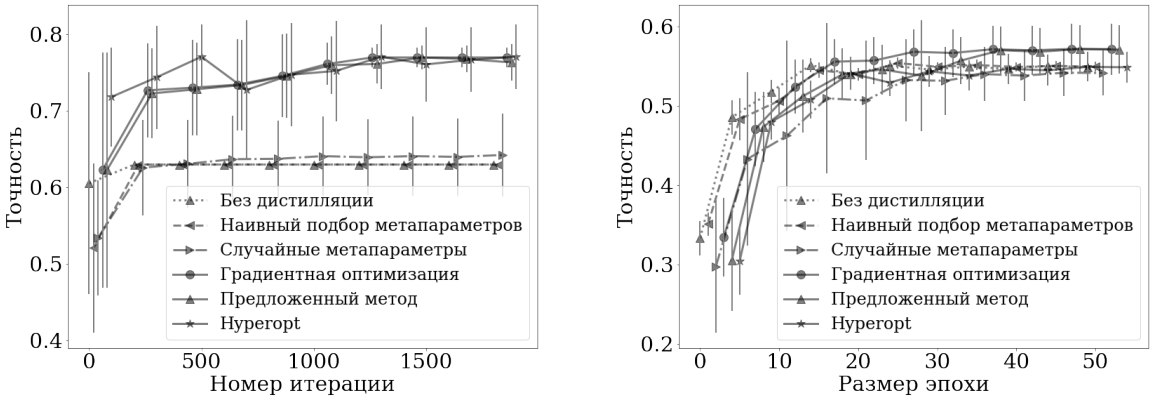
Для оценки полученного метода был проведен эксперимент на синтетической выборке:

$$\mathfrak{D} = \{(\mathbf{x}_i, y_i)\}_{i=1}^{m}, \quad x_{ij} \in \mathcal{N}(0,1),\ j = 1,2, \quad x_{i3} = [\text{sign}(x_{i1}) + \text{sign}(x_{i2}) > 0],$$
$$y_i = \text{sign}(x_{i1} \cdot x_{i2} + \delta),$$

где $\delta \in \mathcal{N}(0, 0.5)$ — это шум. Размер выборки модели-ученика значительно меньше размера выборки модели-учителя и $\mathfrak{D}_{\text{train}}$. Для корректной демонстрации предложенного метода в этом эксперименте выборка была поделена на 3 части: обучающая выборка для модели-учителя, состоящая из 200 объектов, обучающая выборка для модели-ученика, состоящая из 15 объектов, и валидационная выборка, которая также является тестовой, $\mathfrak{D}_{\text{val}} = \mathfrak{D}_{\text{test}}$. Она также состоит из 200 объектов. Визуализация выборки изображена на Рис. 5. Модель-учитель была обучена на протяжении 20000

Таблица 2: Результаты эксперимента. Числа в скобках являются максимальным полученным значением точности в конкретном эксперименте.

Метод	Синтетическая выборка	Fashion-MNIST	Уменьшенный CIFAR-10	CIFAR-10
Без дистилляции	0.63 (0.63)	0.87 (0.88)	0.55 (0.56)	0.65 (0.66)
Наивные метапараметры	0.63 (0.63)	0.87 (0.88)	0.55 (0.56)	0.66 (0.67)
Случайные метапараметры	0.64 (0.72)	0.79 (0.88)	0.54 (0.57)	0.64 (0.67)
Градиентная оптимизация	**0.77** (0.78)	**0.88** (0.89)	**0.57** (0.61)	**0.70** (0.72)
Hyperopt	**0.77** (0.78)	0.87 (0.88)	0.55 (0.58)	0.65 (0.69)
Предложенный метод	0.76 (0.78)	**0.88** (0.89)	**0.57**	**0.70** (0.72)



а) б)

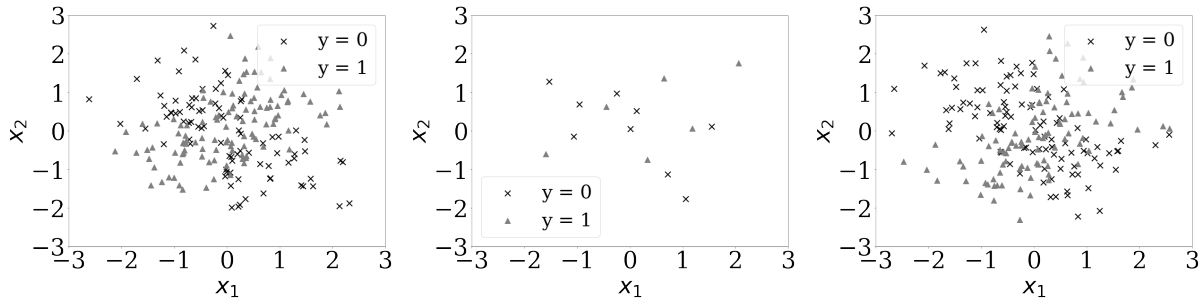
Рис. 4: Точность модели на выборках: а) синтетической, б) уменьшенной CIFAR-10. Здесь и далее точки незначительно смещены относительно оси абсцисс для лучшей читаемости графиков.

итераций методом стохастического градиентного спуска с длиной шага, равной 10^{-2}. Для ее обучения было использовано модифицированное признаковое пространство:

$$x_{i3} = [\text{sign}(x_{i1}) + \text{sign}(x_{i2}) + 0.1 > 0].$$

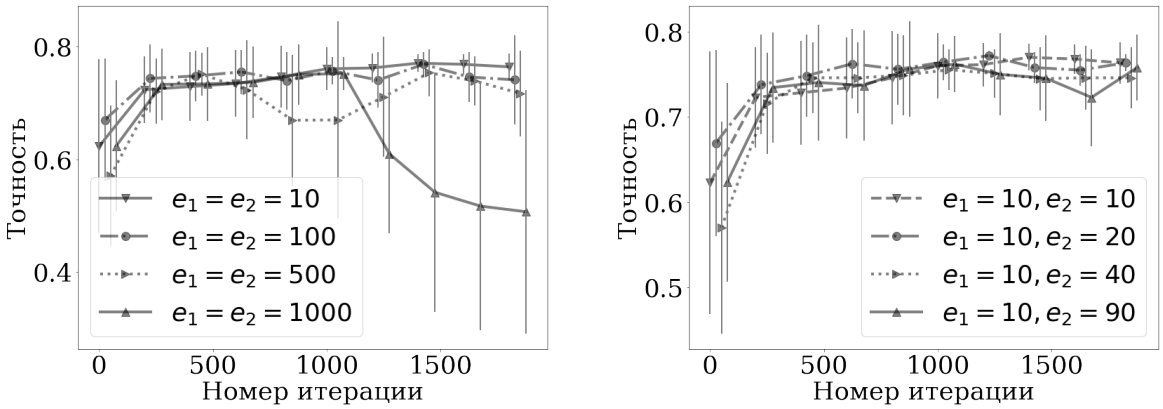
Данная модификация не позволяет модели-учителю безошибочно предсказывать обучающую выборку. В данном случае, для обучения модели-ученика предпочтительно использование только слагаемого дистилляции, $\lambda_1 = 0$. Обучение модели-ученика происходило на протяжении 2000 итераций методом стохастического градиентного спуска с длиной шага, равной 1.0 и $T_{\text{val}} = 0.1$.

Была проведена серия экспериментов для определения наилучших значений e_1 и e_2. На Рис. 6.а приведен график точности для различных e_1 с e_2 равным 10. На Рис. 6.б изображена точность для различных значений e_2. Можно заметить, что с возрастанием e_1 и e_2 качество аппроксимации траектории обновления метапараметров уменьшается.



а) б) в)

Рис. 5: Визуализация выборки для а) модели-учителя; б) модели-ученика; в) тестовой выборки



а) б)

Рис. 6: Точность модели со значениями e_1 и e_2: а) $e_1 = e_2$; б) подбор e_2 при $e_1 = 10$.

На рис. 4.а изображена точность модели для различных методов. Наилучшие результаты были получены для оптимизированных значений метапараметров и предложенного метода. Можно заметить, что предложенный метод хорошо аппроксимирует оптимизацию метапараметров в данном эксперименте.

4.2. Эксперименты на выборках CIFAR-10 и Fashion-MNIST

Обе выборки были разделены в пропорции 9:1 для обучения и валидации. Для оптимизации параметров модели был использован метод стохастического градиентного спуска с начальной длиной шага, равной 1.0. Длина шага умножалась на 0.5 каждые 10 эпох. Значение T_{val} задано равным 1.0.

Для эксперимента на выборке CIFAR-10 была использована предобученная модель ResNet из [10] в качестве модели-учителя. В качестве модели-ученика была использована модель CNN с тремя сверточными слоями и двумя полносвязными слоями.

Для экспериментов на уменьшенной выборке длина шага для оптимизации метапараметров была равна 0.25 и модель обучалась 50 эпох. Для эксперимента на полной выборке была использована дина шага, равная 0.1. Модель обучалась 100 эпох.

Для эксперимента на выборке Fashion-MNIST использовались архитектуры

модели-ученика и модели-учителя, аналогичные архитектурам в эксперименте на выборке CIFAR-10. Для оптимизации метапараметров была использована длина шага, равная 0.1 и модель обучалась 50 эпох.

Из результатов в Таблице 2 видно, что предложенный метод и градиентные методы дают высокое значение точности. Однако, недостаток градиентных методов заключается в «застревании» в точках локального минимума, из-за чего дисперсия результатов получается гораздо выше, чем у остальных методов. Этот эффект можно заметить на Рис. 4 и в Таблице 2.

5. Заключение

Была исследована задача оптимизации параметров модели глубокого обучения. Было предложено обобщение методов дистилляции, заключающееся в градиентной оптимизации метапараметров. На первом уровне оптимизируются параметры модели, на втором — метапараметры, задающие вид оптимизационной задачи. Был предложен метод, уменьшающий вычислительную сложность оптимизации метапараметров для градиентной оптимизации. Были исследованы свойства оптимизационной задачи и методы предсказания траектории оптимизации метапараметров модели. Под метапараметрами модели понимаются параметры оптимизационной задачи дистилляции. Предложенное обобщение позволило производить дистилляцию модели с лучшими эксплуатационными характеристиками и за меньшее число итераций оптимизации. Данный подход был проиллюстрирован с помощью вычислительного эксперимента на выборках CIFAR-10 и Fashion-MNIST, и на синтетической выборке. Вычислительный эксперимент показал эффективность градиентной оптимизации для задачи выбора метарапараметров функции потерь дистилляции. Проанализирована возможность аппроксимировать траекторию оптимизации метапараметров локально-линейной моделью. Планируется дальнейшее исследование оптимизационной задачи и анализ качества аппроксимации траектории оптимизации метапараметров более сложными прогностическими моделями.

СПИСОК ЛИТЕРАТУРЫ

1. *Bakhteev O.Y., Strijov V.V.* Comprehensive analysis of gradient-based hyperparameter optimization algorithms // Ann. Oper. Res, 2020. Vol. 289. No. 1. P. 51–65.

2. *Bergstra J., Bengio Y.* Random search for hyper-parameter optimization. // Journal of machine learning research, 2012. Vol. 13. No. 2.

3. *Bergstra J., Yamins D., Cox D.* Making a science of model search: Hyperparameter optimization in hundreds of dimensions for vision architectures // International conference on machine learning. – 2013. P. 115–123.

4. *Bishop C.M.* Pattern recognition and machine learning (information science and statistics). – 2006.

5. *Hinton G.E., Vinyals O., Dean J.* Distilling the knowledge in a neural network // CoRR, 2015. Vol. abs/1503.02531. URL: http://arxiv.org/abs/1503.02531.

6. *Krizhevsky A., et al.* Learning multiple layers of features from tiny images, 2009.

7. *Liu H., Simonyan K., Yang Y.* Darts: Differentiable architecture search // arXiv preprint arXiv:1806.09055, 2018.

8. *Luketina J., Berglund M., Greff K., Raiko T.* Scalable gradient-based tuning of continuous regularization hyperparameters // CoRR, 2015. Vol. abs/1511.06727. URL: http://arxiv.org/abs/1511.06727.

9. *Maclaurin D., Duvenaud D., Adams R.P.* Gradient-based hyperparameter optimization through reversible learning // CoRR, 2015. Vol. abs/1502.03492. URL: http://arxiv.org/abs/1502.03492.

10. *Passalis N., Tzelepi M., Tefas A.* Heterogeneous knowledge distillation using information flow modeling // Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. – 2020.

11. *Pedregosa F.* Hyperparameter optimization with approximate gradient // CoRR, 2016. Vol. abs/1602.02355. URL: http://arxiv.org/abs/1602.02355.

12. *Rasley J., Rajbhandari S., Ruwase O., He Y.* Deepspeed: System optimizations enable training deep learning models with over 100 billion parameters // Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining. – 2020. P. 3505–3506.

13. *Vatanen T. et al.* Pushing stochastic gradient towards second-order methods – backpropagation learning with transformations in nonlinearities // International Conference on Neural Information Processing. – Springer, Berlin, Heidelberg, 2013. – P. 442–449.

14. *Xiao H., Rasul K., Vollgraf R.* Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms // CoRR, 2017. Vol. abs/1708.07747. URL: http://arxiv.org/278abs/1708.07747.

15. Код вычислительного эксперимента. URL: https://github.com/Intelligent-Systems-Phystech/MetaOptDistillation. Дата обращения: 14.06.2022.

Горпинич М., *Московский физико-технический институт (государственный университет), студент, Долгопрудный*, gorpinich4@gmail.com

Бахтеев О.Ю., *Вычислительный центр имени А.А. Дородницына Федерального исследовательского центра «Информатика и управление» Российской академии наук, к.ф.-м.н., Москва*, bakhteev@phystech.edu

Стрижов В.В., *Вычислительный центр имени А.А. Дородницына Федерального исследовательского центра «Информатика и управление» Российской академии наук, д.ф.-м.н., Москва*, strijov@ccas.ru