Федеральное государственное автономное образовательное учреждение
высшего образования
«Московский физико-технический институт
(национальный исследовательский университет)»
Физтех-школа прикладной математики и информатики
Кафедра интеллектуальных систем

Олейник Михаил Сергеевич

ДИСТИЛЛЯЦИЯ ЗНАНИЙ В ГЛУБОКИХ СЕТЯХ С ПРИМЕНЕНИЕМ МЕТОДОВ ВЫРАВНИВАНИЯ СТРУКТУР МОДЕЛЕЙ

01.03.02 — Прикладная математика и информатика

Выпускная квалификационная работа бакалавра

Научный руководитель:
Бахтеев Олег Юрьевич
канд. физ.-мат. наук

Москва — 2024

Аннотация

В данной работе рассматривается задача дистилляции знаний в глубоких сетях. Методы дистилляции в основном не учитывают разнородность обучаемой модели, так называемого «ученика» и обучающей модели, «учителя». Разнородность моделей ведет к снижению эффективности методов дистилляции и неспособности производить дистилляцию знаний между промежуточными слоями моделей. При этом даже при похожих архитектурах, но при разном количестве слоёв, существующие методы не учитывают большое количество информации, которой располагает учитель, что ведёт не к лучшему качеству модели ученика при дистилляции знаний. Целью работы является предложить метод дистилляции, который будет учитывать больше информации от учителя, сможет работать с разными архитектурами сетей и показывать лучшее качество, по сравнению с классическими методами. Предлагается проводить дистилляцию с помощью максимизации взаимной информации между всеми слоями ученика и учителя. Вводится вариационное распределение, с помощью которого можно максимизировать взаимную информацию, предлагается его вид для свёрточных и линейных слоёв моделей. Были проведены эксперименты, показавшие эффективность предложенного метода, протестированы методы подбора гиперпараметров для данной задачи.

Содержание

1 Введение

Глубокие нейронные сети достигли больших успехов в задачах машинного зрения, обработки естественного языка и других. Однако, лучшие результаты достигают модели с большим количеством параметров, из-за этого их трудно встроить в системы с небольшими вычислительными мощностями, например, мобильные телефоны. Если подобрать размер модели под целевую платформу, уменьшив количество параметров, то сильно потеряем и в качестве.

Одним из подходов, которые позволяют не теряя сильно в качестве, получить модель с меньшим количеством параметров, является дистилляция знаний. Этот подход использует большую предобученную на необходимой задаче модель, называемую учителем, данные о слоях которой переносятся определенным образом в модель меньшего размера, называемую учеником. Перенос чаще всего выражается в дополнительном слагаемом в функции потерь ученика.

Первым, и самым классическим методом дистилляции знаний является дистилляция Хинтона, представленная в статье [1]. Она заключается в том, что одновременно с обучением ученика основной задаче, например, классификации изображений, ученик еще старается повторить логиты предобученного учителя, который предобучен на такую же задачу. У такого подхода много плюсов — он понятный, не требует большого количества дополнительных ресурсов на вычисления, и может работать с разными архитектурами, так как на одной и той же задаче, вне зависимости от архитектуры, размерности логитов будут совпадать. Однако, много информации, которые накопила в себе модель учителя, в таком случае теряется. Можно сказать, что при увеличении размера моделей такая дистилляция теряет эффективность.

Подходы, которые учитывают большее количество информации от учителя, как правило, показывают лучшее качество [2]. Однако чаще всего они требовательны к архитектуре сетей, и в некоторых задачах их будет сложно или невозможно применить. Также возникают сложности в процессе дистилляции, если количество параметров в слое учителя сильно больше,

чем в соответствующем слое ученика, как показано в статье [3].

В настоящее время тема дистилляции знаний активно изучается в мировом сообществе [2], появляются новые методы, в том числе основанные на механизме внимания [4]. Наибольший интерес представляют подходы, которые можно применить, если учитель и ученик имеют разные архитектуры. Так, в статье [5] происходит дистилляция с помощью моделирования информационного потока в учителе, и этот поток в процессе обучения старается имитировать ученик. В статье [6] происходит дистилляция знаний для моделей с одинаковым числом слоёв. Попарно между соответствующими слоями учителя и ученика происходит максимизация взаимной информации. В основе этого метода используется вариационный подход [7].

В некоторых работах [8] проводится дополнительный анализ и оптимизация гиперпараметров, которые возникают в задаче дистилляции. Часто используются методы, которые позволяют уменьшить сложность двухуровневой оптимизации, как например в статье [9].

В настоящей работе предлагается улучшение метода, рассмотренного в статье [6], которое состоит в добавлении возможности проведения дистилляции при разном количестве слоев у учителя и ученика. Схема методов, нашего и базового, изображена на рисунке 1. Актуальность работы заключается в создании метода, который будет использовать большое количество информации от учителя в процессе дистиляции, будет довольно гибким, подходя под разные архитектуры моделей ученика и учителя, и будет показывать лучшее качество.

2 Постановка задачи

Рассмотрим задачу классификации на K классов:

$$\mathfrak{D} = \{(\mathbf{x}_i, y_i)\}_{i=1}^{m},\ \mathbf{x}_i \in \mathbb{R}^n,\ y_i \in \mathbb{Y} = \{1, \ldots, K\},$$

где $\mathfrak{D}$ это доступная выборка объектов, y_i — целевая переменная, а $\mathbf{x}_i$ — данные, описывающие объект, взятые из распределения $p(\mathbf{x})$.

В задаче дистилляции нам необходима кроме обучаемой модели, так называемого ученика, еще модель учителя, которая уже предобучена на такой же задаче, и параметры которой не меняются в процессе обучения. Пропуская входные данные $\mathbf{x}$ через модель, мы можем после каждого слоя получить его активации. Обозначим активации после i-го слоя модели учителя как $\mathbf{t}_i$, а активации после j-го слоя модели ученика как $\mathbf{s}_j$.

Взаимная информация [6] пары $(\mathbf{t}, \mathbf{s})$ определена как:

$$I(\mathbf{t}; \mathbf{s}) = H(\mathbf{t}) - H(\mathbf{t}|\mathbf{s}) = -\mathbb{E}_{\mathbf{t}}[\log p(\mathbf{t})] + \mathbb{E}_{\mathbf{t},\mathbf{s}}[\log p(\mathbf{t}|\mathbf{s})],$$

где энтропия $H(\mathbf{t})$ и условная энтропия $H(\mathbf{t}|\mathbf{s})$ получены из совместного распределения $p(\mathbf{t},\mathbf{s})$. Определение $I(\mathbf{t}; \mathbf{s})$ можно понимать, как уменьшение неопределенности в знаниях учителя, которые закодированны в его слое $\mathbf{t}$, когда известен слой $\mathbf{s}$ ученика.

Теперь, можем определить функцию потерь для модели ученика, минимизируя которую ученик будет не только обучаться для задачи классификации, но и будет максимизироваться взаимная информация между слоями учителя и ученика:

$$\mathcal{L} = \beta \mathcal{L}_{\text{CE}} - (1 - \beta) \sum_{i=1}^{T} \sum_{j=1}^{S} \lambda_{i,j} I(\mathbf{t}_i, \mathbf{s}_j),$$

где $\mathcal{L}_{\text{CE}}$ — кросс-энтропия, T — количество слоёв учителя, S — количество слоёв ученика, $\beta \in (0; 1)$ — гиперпараметр, отвечающий за баланс между минимизацией кросс-энтропии и максимизацией взаимной информации между слоями учителя и ученика, $\lambda_{i,j} \in [0; 1]$ — гиперпараметр, отвечающий за важность связи i-го слоя учителя и j-го слоя ученика.

Схема нашего метода, в сравнении с уже упомянутым методом Sungsoo Ahn[6], изображена на рисунке 1.

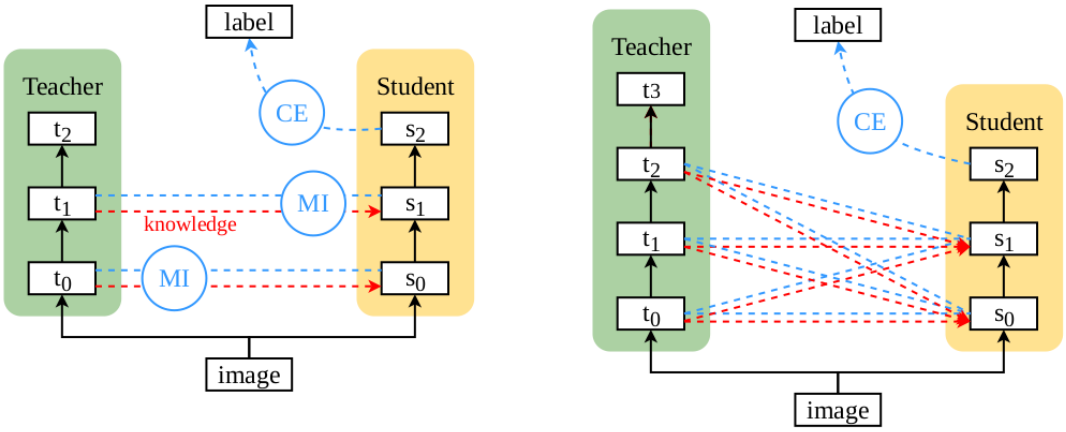

Рис. 1: Схема метода дистилляции Sungsoo Ahn [6] (слева) и предлагаемого нами метода (справа)

3 Вариационная максимизация информации

Функция потерь должна быть минимизирована относительно параметров модели ученика. Однако, сделать это будет сложно, так как трудно вычислить взаимную информацию.

Вместо этого используется вариационная нижняя граница для каждого члена взаимной информации $I(\mathbf{t};\mathbf{s})$, в которой определяется вариационное распределение $q(\mathbf{t}|\mathbf{s})$, которое аппроксимирует $p(\mathbf{t}|\mathbf{s})$:

$$\begin{aligned} I(\mathbf{t};\mathbf{s}) = H(\mathbf{t}) - H(\mathbf{t}|\mathbf{s}) = H(\mathbf{t}) + \mathbb{E}_{\mathbf{t},\mathbf{s}}[\log p(\mathbf{t}|\mathbf{s})] \\ + H(\mathbf{t}) + \mathbb{E}_{\mathbf{t},\mathbf{s}}[\log q(\mathbf{t}|\mathbf{s})] + \mathbb{E}_{\mathbf{s}}[D_{\mathrm{KL}}(p(\mathbf{t}|\mathbf{s})||q(\mathbf{t}|\mathbf{s}))] \\ \geq H(\mathbf{t}) + \mathbb{E}_{\mathbf{t},\mathbf{s}}[\log q(\mathbf{t}|\mathbf{s})]. \quad (1) \end{aligned}$$

Данная техника известна как вариационная максимизация информации [7]. Применяя её к каждому члену взаимной информации в функции потерь, получим:

$$\mathcal{L} = \beta\mathcal{L}_{\mathrm{CE}} - (1-\beta)\sum_{i=1}^{T}\sum_{j=1}^{S}\lambda_{i,j}\mathbb{E}_{\mathbf{t},\mathbf{s}}[\log q(\mathbf{t}^{(i)}|\mathbf{s}^{(j)})].$$

Эта новая функция потерь минимизируется относительно параметров модели ученика и вариационного распределения $q(\mathbf{t}|\mathbf{s})$. Стоит обратить внимание, что энтропия $H(\mathbf{t})$ убрана из выражения, так как не зависит от

оптимизируемых параметров. Также второй член в функции потерь можно интерпретировать как максимизацию условной вероятности соответствия активациям выбранных слоев из модели учителя. Таким образом, сеть учеников получает сжатые знания, необходимые для восстановления активаций выбранных слоев в модели учителя.

Мы выбираем вариационное распределение $q(\mathbf{t}|\mathbf{s})$, как нормальное распределение с средним $\boldsymbol{\mu}(\cdot)$ и стандратным отклонением $\boldsymbol{\sigma}$. При этом, $\boldsymbol{\mu}(\cdot)$ является функцией от $\mathbf{s}$, а $\boldsymbol{\sigma}$ — нет. Параметризация $\boldsymbol{\mu}(\cdot)$ и $\boldsymbol{\sigma}$ зависит от типа слоя, которому соответствует $\mathbf{t}$, и в процессе дистилляции эти параметры оптимизируются вместе с параметрами модели ученика.

Если $\mathbf{t}$ соответствует свёрточному слою сети учителя с размерностями активаций, обозначающими канал, высоту и ширину соответственно, то есть $\mathbf{t} \in \mathbb{R}^{\times H \times W}$, вариационное распределение имеет вид:

$$
\begin{aligned}
-\log q(\mathbf{t}|\mathbf{s}) = -\sum_{c=1}^{C}\sum_{h=1}^{H}\sum_{w=1}^{W}\log q(t_{c,h,w}|\mathbf{s}) = \\
= \sum_{c=1}^{C}\sum_{h=1}^{H}\sum_{w=1}^{W}\log\sigma_c + \frac{(t_{c,h,w}-\mu_{c,h,w}(\mathbf{s}))^2}{2\sigma_c^2} + constant. \quad (2)
\end{aligned}
$$

Рассмотрим все обозначения в данном выражении. Под $t_{c,h,w}$ обозначается скалярный компонент $\mathbf{t}$, взятый по индексу (c,h,w). Кроме того, $\boldsymbol{\mu}(\mathbf{s})$ представляет собой нейронную сеть из нескольких свёрточных слоёв, на вход которой подаются активации $\mathbf{s}$ слоя ученика, а выход имеет размерность, совпадающую с размерностью $\mathbf{t}$ слоя учителя. Стандартное отклонение задаём таким образом, чтобы оно было положительное, в нашем случае:

$$\sigma_c^2 = \log\left(1 + e^{\alpha_c}\right) + \epsilon,$$

где $\alpha_c \in \mathbb{R}$ — обучаемый параметр и ϵ — минимальное значение отклонения, заданное для численной устойчивости.

Если $\mathbf{t}$ соответствует линейному слою сети учителя с размерностью N, то есть $\mathbf{t} \in \mathbb{R}^N$, тогда вариационное распределение имеет вид:

$$-\log q(\mathbf{t}|\mathbf{s}) = -\sum_{n=1}^{N} \log q(t_n|\mathbf{s}) =$$

$$= \sum_{n=1}^{N} \log \sigma_n + \frac{(t_n - \mu_n(\mathbf{s}))^2}{2\sigma_n^2} + constant. \quad (3)$$

В данном выражении обозначения t_n и σ_n аналогичны тем, что представлены выше, а $\boldsymbol{\mu}(\mathbf{s})$ представляет собой линейную модель, так же отображающую активации после слоя $\mathbf{s}$ студента в вектор размерности N.

3.1 Двухуровневая задача оптимизации

До этого момента мы рассматривали задачу как обычную задачу оптимизации с гиперпараметрами. Однако, в таком подходе возникают сложности с подбором значений гиперпараметров. Они могут быть заданы какими-то наивными соображениями, быть подобраны с помощью полного перебора, с использованием вероятностных моделей [10] или с помощью градиентных методов [8]. Опишем постановку задачи двухуровневой оптимизации.

Разобьём нашу выборку на тренировочную и валидационную: $\mathfrak{D} = \mathfrak{D}_{\text{train}} \bigsqcup \mathfrak{D}_{\text{test}}$. Обозначим $\mathcal{L}_{\text{train}}$ как функцию потерь $\mathcal{L}$, вычисляемую на выборке $\mathfrak{D}_{\text{train}}$, а $\mathcal{L}_{\text{val}}$ — как функцию потерь $\mathcal{L}$, вычисляемую на выборке $\mathfrak{D}_{\text{val}}$.

Определим вектор $\boldsymbol{\lambda}$ из всех гиперпараметров задачи:

$$\boldsymbol{\lambda} = [\lambda_{0,0}, \ldots, \lambda_{i,j}, \ldots, \beta].$$

Определим все обучаемые параметры модели ученика и обучаемые параметры взаимной информации как $\mathbf{w}$.

Заметим, что функции потерь $\mathcal{L}_{\text{train}}$ и $\mathcal{L}_{\text{val}}$ зависят и от $\boldsymbol{\lambda}$, и от $\mathbf{w}$. Цель — найти $\hat{\boldsymbol{\lambda}}$, которое минимизирует функцию потерь на валидационной выборке $\mathcal{L}_{\text{val}}(\hat{\mathbf{w}}, \hat{\boldsymbol{\lambda}})$, где параметры $\mathbf{w}$ получаются в результате минимизации функции потерь на тренировочной выборке $\hat{\mathbf{w}} = \arg\min_{\mathbf{w}} \mathcal{L}_{\text{train}}(\mathbf{w}, \hat{\boldsymbol{\lambda}})$.

И это определяет задачу двухуровневой оптимизации[9]:

$$\begin{aligned} &\min_{\boldsymbol{\lambda}} \quad \mathcal{L}_{\text{val}}(\hat{\mathbf{w}}(\boldsymbol{\lambda}), \boldsymbol{\lambda}), \\ &\text{s.t.} \quad \hat{\mathbf{w}}(\boldsymbol{\lambda}) = \arg\min_{\mathbf{w}} \mathcal{L}_{\text{train}}(\mathbf{w}, \boldsymbol{\lambda}). \end{aligned} \quad (4)$$

4 Вычислительный эксперимент

Цель вычислительного эксперимента — проверить качество предлагаемого метода дистилляции, проанализировать получившиеся модели и их зависимость от гиперпараметров.

Все методы оцениваются на датасетах для классификации изображений: CIFAR-10 [11] и Fashion-MNIST [12]. CIFAR-10 содержит в себе цветные изображения, разбитые на 10 разных классов. Fashion-MNIST содержит в себе изображения в оттенках серого, разбитые на 10 классов одежды и обуви.

Сравниваются и анализируются в ходе эксперимента следующие методы:

1. Оптимизация без дистилляции.
2. Оптимизация с дистилляцией Хинтона [1].
3. Оптимизация с дистилляцией Sungsoo Ahn [6], то есть:

$$\lambda_{i,j} = 1 \quad \text{если} \quad i = j,$$

$$\lambda_{i,j} = 0 \quad \text{если} \quad i \neq j.$$

4. Предлагаемый метод дистилляции, со всеми связями, то есть $\lambda_{i,j} = 1$.
5. Предлагаемый метод дистилляции, со случайной инициализацией гиперпараметров λ.
6. Предлагаемый метод дистилляции, оптимизируем гиперпараметры, используя вероятностные модели. Для этого метода используем библиотеку Optuna [10].

Каждый датасет был разделён на тренировочную и тестовую часть. Во всех методах значение гиперпараметра $\beta = 0.5$. Метрикой качества была выбрана точность.

Как модель учителя использовалась предобученная свёрточная модель под именем Tiny, которая состоит из трёх свёрточных слоёв, и двух

Tiny	VeryTiny
Свёрточный 2D слой (3×3, 4 фильтра)	Свёрточный 2D слой (3×3, 8 фильтров)
Свёрточный 2D слой (3×3, 8 фильтров)	Свёрточный 2D слой (3×3, 16 фильтров)
Свёрточный 2D слой (3×3, 16 фильтров)	Свёрточный 2D слой (3×3, 32 фильтра)
Полносвязный слой (64 нейрона)	Полносвязный слой (64 нейрона)
Полносвязный слой (10 нейронов)	Полносвязный слой (10 нейронов)

Таблица 1: Архитектура моделей Tiny и VeryTiny, слои заданы поочередно, сверху вниз

линейных. В качестве ученика использовалась модель под именем VeryTiny, которая отличается от Tiny уменьшенным в два раза количеством каналов в свёрточных слоях. Архитектура моделей изображена на рисунке 1.

Во всех алгоритмах шаг градиентного спуска был равен 0.1, обучение длилось 50 эпох.

Метод	CIFAR-10	Fashion-MNIST
Без дистилляции	0.541	0.839
Дистилляция Хинтона	0.563	0.849
Дистилляцией Sungsoo Ahn	0.591	0.852
Наш метод, все связи	0.590	0.853
Наш метод, случайные гиперпараметры	0.595	0.859
Наш метод, вероятностная оптимизация	0.608	0.857

Таблица 2: Значение качества экспериментов на датасетах CIFAR-10 и Fashion-MNIST

Финальные результаты продемонстрированы в таблице 2. Зависимость качества ученика от эпохи для CIFAR-10 изображена на рисунке 2.

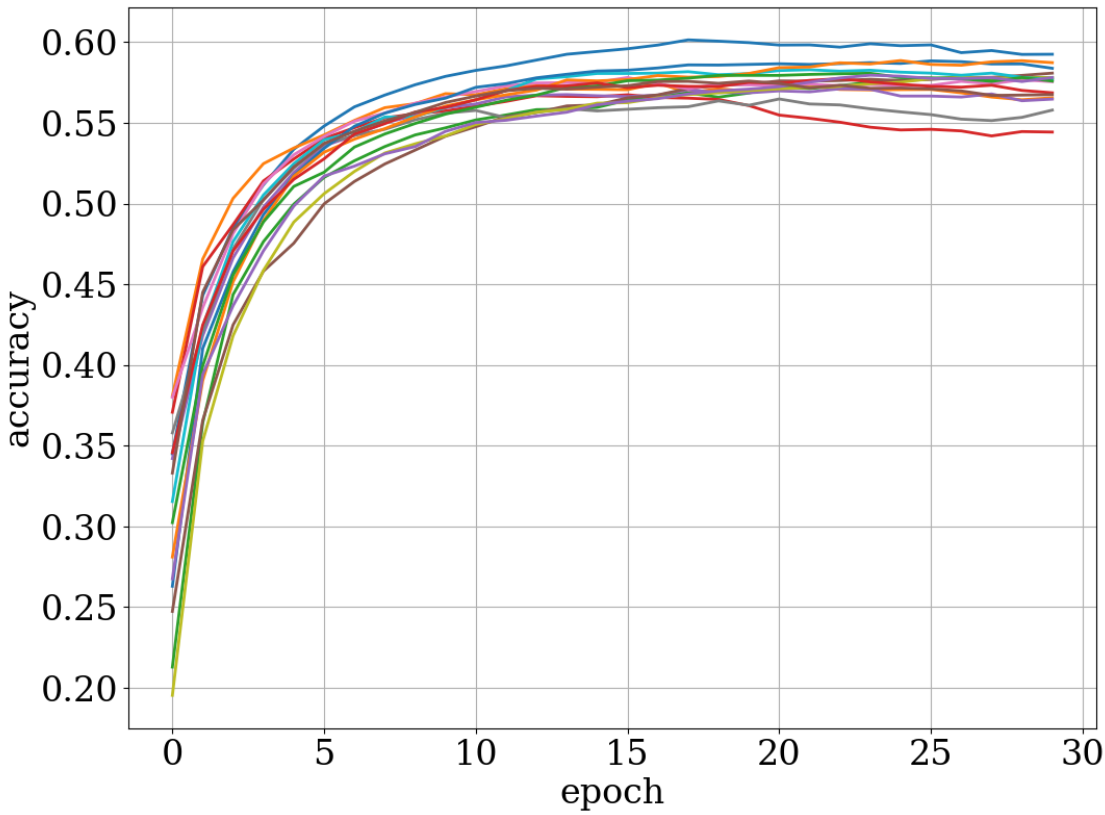

Рис. 2: Зависимость качества модели ученика от эпохи на датасете CIFAR-10

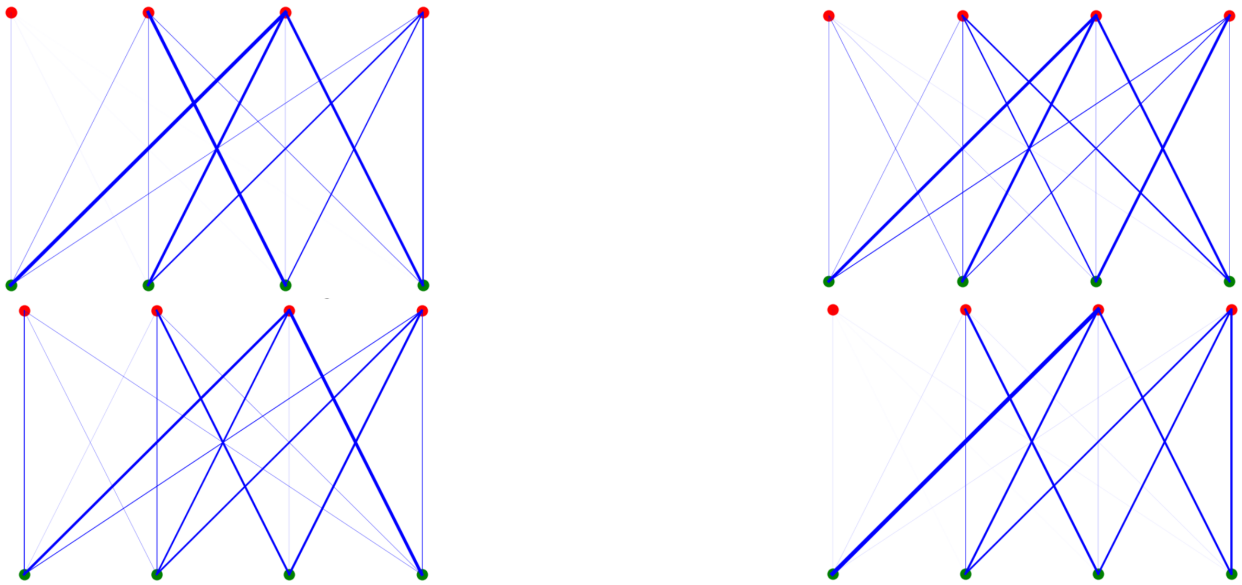

Рис. 3: Иллюстрация коэффициентов у четырех лучших моделей по качеству в эксперименте с случайным подбором гиперпараметров на датасете CIFAR-10. Зелёные точки — слои ученика, красные — слои учителя. Чем толще линия, тем больше коэффициент λ у соответствующей связи.

Интерес для дальнейших исследований представляет собой связь гипер-

параметров $\lambda_{i,j}$ с итоговым качеством модели. На рисунке 3 схематически изображены значения коэффициентов $\lambda_{i,j}$ в четырех лучших запусках в эксперименте с случайным подбором гиперпараметров, на датасете CIFAR-10. Видно некоторую закономерность, что связи, которые ведут к третьему слою учителя, у которого наибольшее число параметров, больше, чем остальные. В дальнейших исследованиях планируется подробнее изучить связь гиперпараметров и вывести условия или законы, которые связывают их с качеством модели ученика при дистилляции.

5 Заключение

В данной работе исследовалась задача дистилляции в глубоких сетях. Предложен метод дистилляции, максимизирующий взаимную информацию между отдельными слоями ученика и учителя. Продемонстрировано, что предложенный метод достигает лучшего качества, чем базовый метод дистилляции знаний. Дистилляция проводилась с помощью максимизации взаимной информации между всеми слоями ученика и учителя. Были проведены эксперименты, показавшие эффективность предложенного метода. Поставлены задачи для дальнейших исследований, а именно выяснить связь между значениями гиперпараметров и качеством модели ученика.

Список литературы

[1] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2015.

[2] Jianping Gou, Baosheng Yu, Stephen J. Maybank, and Dacheng Tao. Knowledge distillation: A survey. *International Journal of Computer Vision*, 129(6):1789–1819, March 2021. ISSN 1573-1405. doi:10.1007/s11263-021-01453-z. URL http://dx.doi.org/10.1007/s11263-021-01453-z.

[3] Seyed Iman Mirzadeh, Mehrdad Farajtabar, Ang Li, Nir Levine, Akihiro Matsukawa, and Hassan Ghasemzadeh. Improved knowledge distillation via teacher assistant. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pages 5191–5198, 2020.

[4] Peyman Passban, Yimeng Wu, Mehdi Rezagholizadeh, and Qun Liu. Alp-kd: Attention-based layer projection for knowledge distillation, 2020.

[5] Nikolaos Passalis, Maria Tzelepi, and Anastasios Tefas. Heterogeneous knowledge distillation using information flow modeling. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2339–2348, 2020.

[6] Sungsoo Ahn, Shell Xu Hu, Andreas Damianou, Neil D. Lawrence, and Zhenwen Dai. Variational information distillation for knowledge transfer. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.

[7] David Barber and Felix Agakov. The im algorithm: a variational approach to information maximization. *Advances in neural information processing systems*, 16(320):201, 2004.

[8] M Gorpinich, O Yu Bakhteev, and VV Strijov. Gradient methods for optimizing metaparameters in the knowledge distillation problem. *Automation and Remote Control*, 83(10):1544–1554, 2022.

[9] Hanxiao Liu, Karen Simonyan, and Yiming Yang. DARTS: Differentiable architecture search. In *Proc. ICLR*, 2019.

[10] Takuya Akiba, Shotaro Sano, Toshihiko Yanase, Takeru Ohta, and Masanori Koyama. Optuna: A next-generation hyperparameter optimization framework. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2019.

[11] Alex Krizhevsky. Learning multiple layers of features from tiny images. 2009. URL https://api.semanticscholar.org/CorpusID:18268744.

[12] Han Xiao, Kashif Rasul, and Roland Vollgraf. Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms. *arXiv preprint arXiv:1708.07747*, 2017.