Дистилляция моделей и данных

Баринов Никита
МФТИ

Филатов Андрей
МФТИ

Abstract

Во многих задачах машинного обучения точность предсказания модели зависит от её размера. При этом, чтобы получить хорошее качество, требуется много данных, что в свою очередь увеличивает время обучения. Дистилляция моделей позволяет уменьшить их размер, при этом не сильно потеряв в качестве, а дистилляция данных позволяет существенно снизить время обучения. В статье мы предлагаем использовать модель-эксперт, которая дистиллируется. Это позволяет ускорить сходимость моделей при дистилляции данных и качество моделей, обученных на этих данных. Такой процесс называется дистилляцией моделей и данных, и в статье мы предлагаем одно из решений, а для сравнения и обработки результатов провели эксперименты на выборке CIFAR10

Keywords Deep Learning · Distilling the Knowledge · Dataset Distillation · Model Compression

1 Introduction

Глубокое обучение добилось огромного успеха за последние несколько лет в различных областях, таких как компьютерное зрение([1]), обработка естественного языка([2]) и распознавание речи([3]). Но всё это требует больших вычислительные и временные ресурсы. Например, как говорится в оригинальной статье [4], GPT-2 с 1,5 миллиардом параметров было использовано 40 терабайт текстовых данных, и модель обучалась на суперкомпьютерных кластерах в течение нескольких недель.

Со временем начали появляться методы «сжатия» моделей без сильной потери качества. Появилась концепция дистилляции знаний(knowledge distillation) – это способ обучения нейросетевых моделей машинного обучения, направленный на передачу знаний от модели-учителя к модели-ученику. В широком диапазоне практически значимых задач машинного обучения точность предсказания модели существенно зависит от её размера. При этом зачастую данная зависимость выглядит очень просто: последовательное увеличение размеров модели позволяет последовательно улучшать точность её предсказаний. Однако такой безграничный рост приводит к ряду проблем, связанных с практическим применением итоговых моделей. Первой статьёй, в которой можно встретить дистилляцию знаний в современном виде является [5]. В ней предлагается сжать ансамбль моделей в одну модель, тем самым значительно уменьшив её размер.

Поскольку проблема есть и с большим объёмом денных, появляется дистилляция данных(dataset distillation) - это существенное уменьшение выборки, путём создания искусственных объектов (синтетических данных), которые агрегируют полезную информацию, хранящуюся в данных, и позволяют настраивать алгоритмы машинного обучения не менее эффективно, чем на всех данных. Если мы имеем лишь несколько достаточно хорошо дистиллированных изображений, мы можем гораздо эффективнее обучить нейронную сеть на целом наборе данных, по сравнению с традиционным обучением, при котором часто используются десятки тысяч шагов градиентного спуска. Каждый элемент синтетических данных содержит в себе больше информации, чем отдельный элемент исходной выборки.

В данной работе предлагается новый подход: одновременная дистилляция модели и данных методами [5] и [6]. Выборку CIFAR10 мы сократили до десяти изображений на каждый из десяти классов, в качестве модели-учителя выступила ResNet50, а в качестве модели-ученика - ConvNet. Мы привели несколько методов одновременной дистилляции и сравнили их эффективность на тестовой части выборки CIFAR10.

2 Related works

Сегодня существует несколько решений проблемы дистилляции моделей или данных в отдельности.

2.1 Дистилляция моделей

В статье [7] говорится, что промежуточные веса или особенности слоев могут также использоваться для обучения меньшей модели. Ещё один способ дистилляции, описанный в статье [8], он основан на том, что меньшая модель(ученик) обучается аналогично большей(учитель), тем самым получается конкурентоспособная производительность. В [5] применили дистилляцию, чтобы сжать ансамбль в одну модель. Одной из последних работ является [9]. В ней описывается дистилляция в онлайн-режиме: модель и ученик совместно оптимизируются на каждой итерации. Также существует кросс-модельная дистилляция(передача знаний между промежуточными моделями), одним из сценариев котороя является [10]: имеется граф взаимоотношений между моделями, а передача знаний осуществляется при помощи предложенной функции потерь, сохраняющей локальность.

2.2 Дистилляция данных

В статье [11] сначала данные инициализируются случайным шумом, а затем при помощи градиентного спуска происходит обновление синтетических данных. Описанный метод имеет явный недостаток: он ограничен числом эпох обучения. Использование теоремы о неявной функции в [12] помогает избавиться от такого недостатка. В [13] в качестве функции ошибки используется расстояние между градиентами этой ошибки по параметрам ученика, которые получаются при обучении на обычных и дистиллированных данных. Альтернативным вариантом может быть введение генеративной модели(может создавать новые данные, которые похожи на те, что были использованы для ее обучения), способной из шума и меток класса создавать необходимые для обучения синтетические изображения, этот подход подробно описан в [14]. Статья [6] предлагает метод дистилляции путем создания выборки, на которой динамика обучения такая же, как и на исходной.

3 Постановка проблемы

В этом разделе описывается формальная постановка проблем дистилляции данных, дистилляции моделей и предлагаемое решениеодновременной дистилляции моделей и данных.

3.1 Дистилляция данных

Пусть $\mathcal{D}_{real} = \{(\mathbf{x_i}, y_i)\}_{i=1}^{N}$ — исходная выборка. Наша задача — создать меньшую выборку $\mathcal{D}_{syn} = \{(\hat{\mathbf{x}}_\mathbf{i}, \hat{y}_i\}_{i=1}^{M}$, где $M \ll N$ и такой, что качество модели, обученной на нём сопоставимо с качеством при обучении на исходных данных. Наш метод дистилляции предполагает создание экспертных траекторий обучения τ^*, под которыми понимается последовательность параметров $\{\theta_t^*\}_{t=0}^{T}$, полученных во время обучения нейронной сети на $\mathcal{D}_{real}$. Чтобы получить экспертные траектории, предлагается обучить большое количество нейронных сетей на $\mathcal{D}_{real}$ и сохраним их параметры на каждой эпохе. Также определим $\hat{\theta}_t$ - параметры модели-студента, обученной на $\mathcal{D}_{syn}$ на шаге обучения t. На каждом шаге обучения мы будем выбирать случайно θ_t^*, инициализировать этим значением парметры модели-студента $\theta_t^* := \theta_t^*$. Установим верхнюю границу T^{max} на число t, чтобы игнорировать ту часть обучения, где параметры меняются незначительно.

Пусть $l(\mathcal{A}(\mathcal{D}_{syn}), \theta_t)$ - дифференцируемая функция потерь, $\mathcal{A}$ - дифференцируемая техника аугментации данных(это метод, используемый в машинном обучении, для увеличения размера обучающего набора данных путем создания новых примеров на основе существующих данных.) [7]. После инициализации параметров модели-студента мы совершим N шагов градиентного спуска по параметрам $\hat{\theta}_t$:

$$\hat{\theta}_{t+n+1} = \hat{\theta}_{t+n} - \alpha \nabla l(\mathcal{A}(\mathcal{D}_{syn}), \hat{\theta}_{t+n}), \tag{1}$$

где α - шаг обучения модели-студента, используемый для обновления её параметров. После обучения градиентного спуска для конкретной траектории $\tau^* \in \{\tau_i^*\}$ считаем

$$\mathcal{L} = \frac{\| \hat{\theta}_{t+N} - \theta_{t+M}^* \|_2^2}{\| \theta_t^* - \theta_{t+M}^* \|_2^2}, \tag{2}$$

где $\mathcal{L}$ - функция потерь между конечными параметрами студента и учителя, нормированная на пройденное учителем расстояние, что помогает получать информацию о более поздних стадиях его обучения, где параметры меняются не сильно. В конце мы обновляем $\mathcal{D}_{syn}$ в соответствии с обучаемым параметром α и посчитанной функцией $\mathcal{L}$. Итоговый алгоритм выглядит так:

Algorithm 1: Data Distillation

Data: $\{\tau_i^*\}$ – множество параметров учителей, обученных на $\mathcal{D}_{real}$
Data: M - число обновлений между стартовыми и целевыми параметрами учителя
Data: N - число обновлений студента за один шаг дистилляции
Data: $\mathcal{A}$ – дифференцируемая функция аугментации
Data: $T^{max} < T$ – максимальная стартовая эпоха
Result: Дистиллированный набор $\mathcal{D}_{syn}$ и α
$\mathcal{D}_{syn} \leftarrow \mathcal{D}_{real}$;
$\alpha \leftarrow \alpha_0$;
for $step : 1 \,..\, N$ do
  $\tau^* \sim \{\tau_i^*\}, \tau^* = \{\theta_t^*\}_0^T$ - выбираем траекторию обучения;
  $t \leq T^*$ - случайно выбираем начальную эпоху;
  $\theta_t^* := \theta_t^*$ - инициализируем веса студента параметрами учителя;
  for $n : 0 \,..\, N-1$ do
    $b_{t+n} \sim \mathcal{D}_{syn}$ - выбрать мини-батч из $\mathcal{D}_{syn}$;
    $\hat{\theta}_{t+n+1} \leftarrow \hat{\theta}_{t+n} - \alpha \nabla l(\mathcal{A}(\mathcal{D}_{syn}), \hat{\theta}_{t+n})$;
  end
  $\mathcal{L} \leftarrow \| \hat{\theta}_{t+N} - \theta_{t+M}^* \|_2^2 \,/\, \| \theta_t^* - \theta_{t+M}^* \|_2^2$;
  Изменить $\mathcal{D}_{syn}$ и α в зависимости от $\mathcal{L}$;
end

Итого первая оптимизационная задача, которая решает дистилляцию данных, выглядит так:

$$\hat{\mathbf{X}}, \alpha = \arg\min_{\mathbf{x}, \alpha} \mathcal{L}(\mathbf{x}, \alpha, \theta),\ \mathcal{D}_{syn} = \bigcup_{\hat{\mathbf{x}} \in \hat{\mathbf{X}}} \hat{\mathbf{x}}. \tag{3}$$

3.2 Дистилляция моделей

Далее стоит задача обучения нейросети на дистиллированных данных $\mathcal{D}_{syn}$ и дистилляция модели.

Def 1 : Дистилляция модели - снижение сложности модели путем выбора модели в множестве более простых моделей на основе анализа пространства параметров и предсказаний целевой перменной более сложной фиксированной модели.

Def 2 : Учитель - фиксированная модель, ответы которй используются при выборе модели-ученика.

Def 3 : Ученик - модель, которая выбираемся согласно заданному критерию качества учителя.

Итак, решается задача класификации:

$$\mathcal{D} = \{(\hat{\mathbf{x}}_\mathbf{i}, y_i)\}_{i=1}^R,$$

где $y_i \in \mathbb{Y} = 1, 2, ..., R$, R - число классов, $\hat{\mathbf{x}}_\mathbf{i} \in \mathbb{R}^n$.

В дистилляции Хинтона [5] рассматривается параметрическое семейство функций:

$$\mathcal{G} = \{\mathbf{g} \mid \mathbf{g} = softmax(\mathbf{z}(\mathbf{x})/T),\ \mathbf{z} : \mathbb{R}^n \to \mathbb{R}^R\}, \tag{4}$$

где $\mathbf{z}$ - дифференцируемая параметрическая функция заданной структуры, T - параметр температуры. В качестве модели-учителя рассматривается функция $\mathbf{f}$ из множества:

$$\mathcal{F} = \{\mathbf{f} \mid \mathbf{f} = softmax(\mathbf{v}(\mathbf{x})/T,\ \mathbf{v} : \mathbb{R}^n \to \mathbb{R}^R\}, \tag{5}$$

где $\mathbf{z}$ - дифференцируемая параметрическая функция заданной структуры, T - параметр температуры.

При этом температура T имеет свойства:

1. при T $\to 0$ получаем вектор, в котором один из классов имеет единичную вероятность;
2. при T $\to \infty$ получаем вектор, в котором все классы равновероятны.

Функция потерь $\mathcal{L}$ учитывает перенос инфорации от модели-учителя $\mathbf{f}$ к ученику $\mathbf{g}$ и имеет вид:

$$\mathcal{L}(\mathbf{g}) = -\sum_{i=1}^{m}\sum_{r=1}^{R} y_i^r \log \mathbf{g}(\mathbf{x_i})\Big|_{T=1} - \sum_{i=1}^{m}\sum_{r=1}^{R} \mathbf{f}(\mathbf{x_i})\Big|_{T=T_0} \log \mathbf{g}(\mathbf{x_i})\Big|_{T=T_0}, \tag{6}$$

где первое слаагемое отвечает за исходную функцию потерь, а второе - за дистилляцию. Итого получаем оптимизационную задачу:

$$\hat{\mathbf{g}} = \arg\min_{\mathbf{g}\in\mathcal{G}} \mathcal{L}(\mathbf{g}). \tag{7}$$

3.3 Дистилляция моделей и данных

В нашей работе мы предлагаем модифицированный способ дистилляции: в функцию потерь, которая используется для дистилляции данных, добавим слагаемое, отвечающее за дистилляцию знаний от модели-учителя. Тем самым, новая функция потерь будет иметь вид:

4 Эксперимент

4.1 Базовый эксперимент

Проведем базовый эксперимент на выборке CIFAR10, с которым будем сравнивать результаты в дальнейшем: дистилялция моделей. Для этого в качестве модели-учител используем ResNet50, в качестве модели-ученика ConvNet.

Модель	Accuracy
ConvNet	——%
ResNet50	74.83%

Таблица 1: Точность моделей в отдельности

Зависимость функции потерь и точности от эпохи обучения для каждой модели в отдельности можно видеть на графиках:

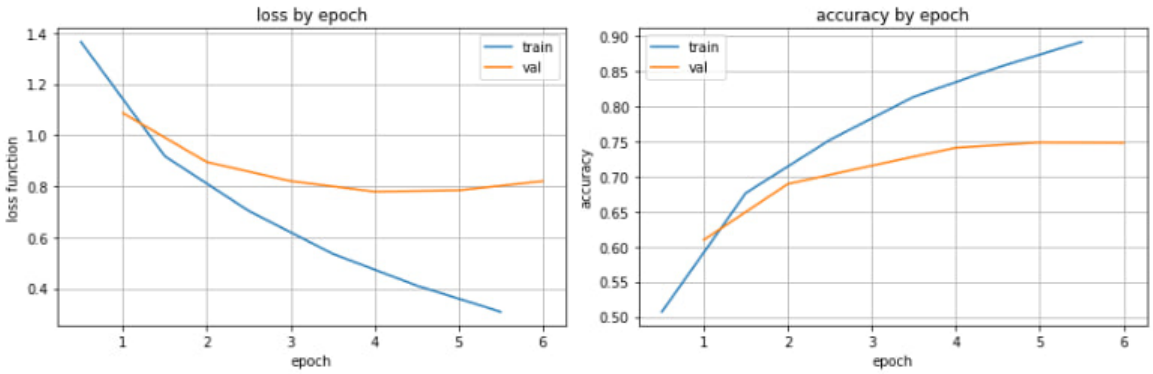

Рис. 1: Процесс обучения ResNet50

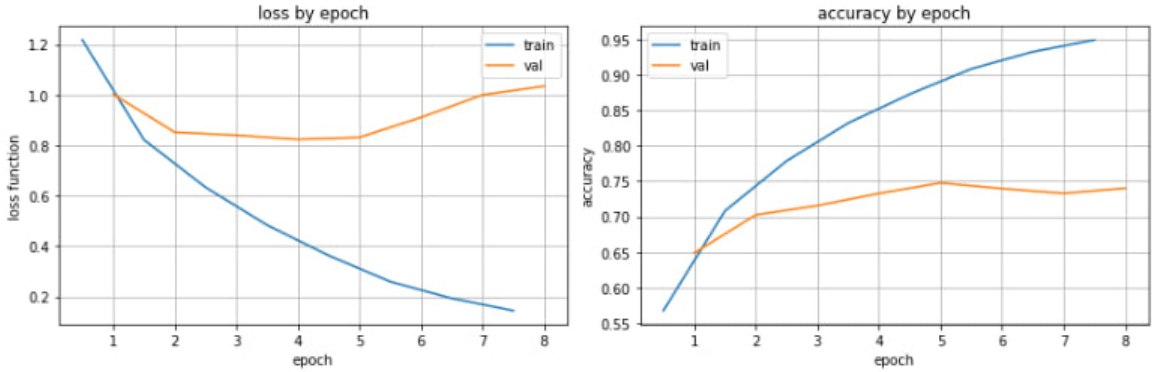

Рис. 2: Процесс обучения ConvNet

Теперь используем дистилляцию моделей. Тут вводится понятие distillation weight - коэффициент влияния модели-ученика. Обозначим его α, тогда функция потерь ученика примет вид:

$$(1-\alpha)\cdot H(p,q)+\alpha\cdot D_{KL}(p|q),$$

где кросс-энтропия вычисляется по формуле:

$$H(p,q)=-\sum_x p(x)\log q(x),$$

а дивергенция Кульбака-Лейблера:

$$D_{KL}(p|q)=\sum_x p(x)\log\frac{p(x)}{q(x)}.$$

Мы использовали оптимизатор *SGD* и 6 эпох градиентного спуска. Результаты представлены в таблице:

α	Функция потерь	Accuracy
0.25	$0.75\cdot H(p,q)+0.25\cdot D_{KL}(p\|q)$	73.30%
0.5	$0.5\cdot H(p,q)+0.5\cdot D_{KL}(p\|q)$	71.71%
0.75	$0.25\cdot H(p,q)+0.75\cdot D_{KL}(p\|q)$	69.43%

Таблица 2: Дистилляция моделей

Вывод: дистилляция моделей не сильно ухудшила качество, причем качество немного падает с уменьшением влияния учителя.

4.2 Основной эксперимент

ResNet50, обученный на всех данных, добавляется в лосс дистилляции данных.

Мы получили дистиллированные изображения для $T=0.5,1,3$ и $\alpha=0.3,0.7$

Итого, функция потерь, по которой происходит дистилляция моделей и данных:

$$l()=(1-\alpha)\cdot H(p,q)+\alpha\cdot D_{KL}(p|q)$$

T	Accuracy
0.25	73.30%
0.5	71.71%
0.75	69.43%

Таблица 3: Дистилляция моделей и данных, $\alpha=0.3$

Первый столбец	Второй столбец
2*Строка 1	Строка 1, часть 1 Строка 1, часть 2
Строка 2	Строка 2, часть 1 Строка 2, часть 2

Первый столбец	Второй столбец
* Строка 1, часть 1	Строка 1, часть 2
* Строка 2, часть 1	Строка 2, часть 2

Обучаем модель ученика:

$Loss = (1 - \alpha) * CE + \alpha * KLloss$

5 Вывод

Список литературы

[1] Meng-Hao Guo, Tian-Xing Xu, Jiang-Jiang Liu, Zheng-Ning Liu, Peng-Tao Jiang, Tai-Jiang Mu, Song-Hai Zhang, Ralph R Martin, Ming-Ming Cheng, and Shi-Min Hu. Attention mechanisms in computer vision: A survey. Computational Visual Media, 8(3):331–368, 2022.

[2] Vineet Raina, Srinath Krishnamurthy, Vineet Raina, and Srinath Krishnamurthy. Natural language processing. Building an Effective Data Science Practice: A Framework to Bootstrap and Manage a Successful Data Science Practice, pages 63–73, 2022.

[3] Aswin Shanmugam Subramanian, Chao Weng, Shinji Watanabe, Meng Yu, and Dong Yu. Deep learning based multi-source localization with source splitting and its effectiveness in multi-talker speech recognition. Computer Speech & Language, 75:101360, 2022.

[4] Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. Language models are unsupervised multitask learners. OpenAI blog, 1(8):9, 2019.

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

[6] George Cazenavette, Tongzhou Wang, Antonio Torralba, Alexei A Efros, and Jun-Yan Zhu. Dataset distillation by matching training trajectories. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 4750–4759, 2022.

[7] Adriana Romero, Nicolas Ballas, Samira Ebrahimi Kahou, Antoine Chassang, Carlo Gatta, and Yoshua Bengio. Fitnets: Hints for thin deep nets. arXiv preprint arXiv:1412.6550, 2014.

[8] Jimmy Ba and Rich Caruana. Do deep nets really need to be deep? Advances in neural information processing systems, 27, 2014.

[9] Inseop Chung, SeongUk Park, Jangho Kim, and Nojun Kwak. Feature-map-level online adversarial knowledge distillation. In International Conference on Machine Learning, pages 2006–2015. PMLR, 2020.

[10] Hanting Chen, Yunhe Wang, Chang Xu, Chao Xu, and Dacheng Tao. Learning student networks via feature embedding. IEEE Transactions on Neural Networks and Learning Systems, 32(1):25–35, 2020.

[11] Tongzhou Wang, Jun-Yan Zhu, Antonio Torralba, and Alexei A Efros. Dataset distillation. arXiv preprint arXiv:1811.10959, 2018.

[12] Jonathan Lorraine, Paul Vicol, and David Duvenaud. Optimizing millions of hyperparameters by implicit differentiation. In International Conference on Artificial Intelligence and Statistics, pages 1540–1552. PMLR, 2020.

[13] Bo Zhao, Konda Reddy Mopuri, and Hakan Bilen. Dataset condensation with gradient matching. arXiv preprint arXiv:2006.05929, 2020.

[14] Felipe Petroski Such, Aditya Rawal, Joel Lehman, Kenneth Stanley, and Jeffrey Clune. Generative teaching networks: Accelerating neural architecture search by learning to generate synthetic training data. In International Conference on Machine Learning, pages 9206–9216. PMLR, 2020.