Дистилляция данных.

Латышев Александр

Дистилляция данных

Дистилляция данных.

Сейчас для многих задач сложно обучить модели, которые способны дать хороший результат на тестовой выборке. На это есть разные причины.

Одна из таких причин это слишком большие массивы данных.

Решение - дистилляция данных.

Дистилляция данных.

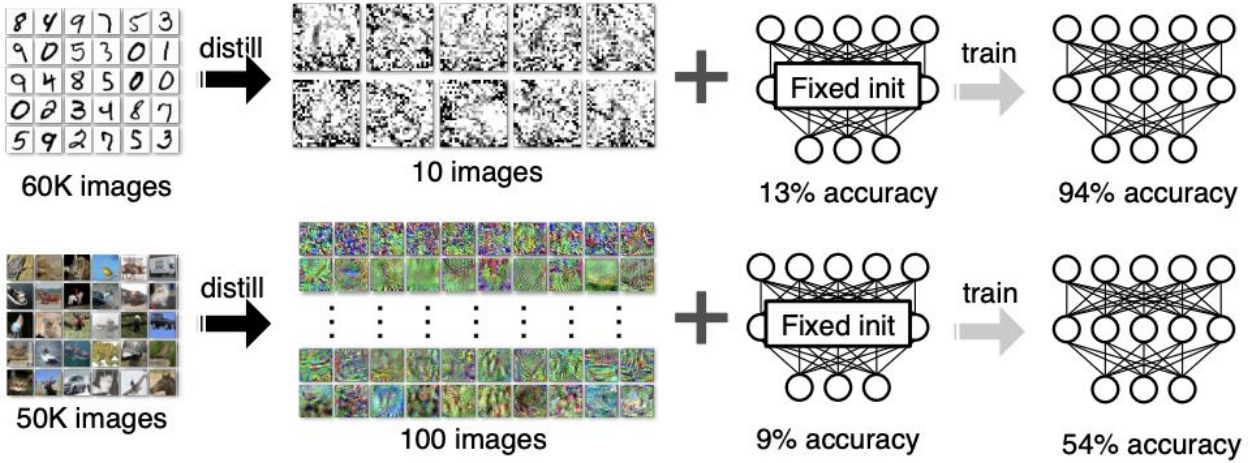
Слово дистилляция(distillation) означает очистку, мы и будем очищать наши данные.

В статье дистилляция данных разбирается на примере нескольких массивов картинок.

Дистилляцию данных можно применять для двух задач, для проверки качества самих данных или для задачи классификации изображений.

Дистилляция данных.

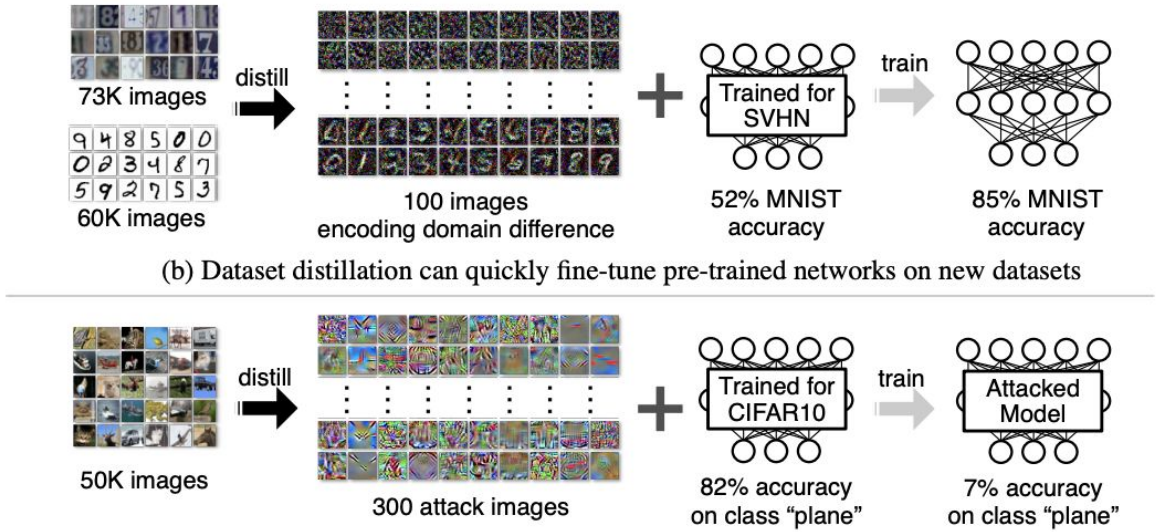
Например в задаче классификации можно уменьшить набор данных, чтобы обучение прошло намного быстрее.



(a) Dataset distillation on MNIST and CIFAR10

Дистилляция данных.

Также можно быстро завершить обучение уже готовой на какую-то часть модели или проверить, не является наша модель просто переобученной под неверные данные.



Ключевые моменты

Ключевые моменты

- В статье не оптимизируются гиперпараметры во время градиентного спуска
- Для хороших результатов дистилляции желательно хорошо понимать внутреннее строение тренировочных данных
- Первыми подходами в дистиллировании были попытки выбрать подмассив исходных данных, в котором бы содержалась вся “суть”

Подход к решению

Фиксированная инициализация

Первым рассматривается самый простой вариант обучения, с константной базовой инициализацией нейронной сети и одним градиентным шагом для обучения.

Как всегда надо минимизировать функцию ошибки, которая в нашем случае выглядит так.

$$\theta^* = \arg\min_\theta \frac{1}{N} \sum_{i=1}^{N} \ell(x_i, \theta) \triangleq \arg\min_\theta \ell(\mathbf{x}, \theta),$$

Случайная инициализация

Для случайных гиперпараметров нейронной сети не подойдет обычная дистилляция данных, тк дистиллированные данные содержат информацию и о данных, и о гиперпараметрах сети.

Решение: создать множество дистиллированных данных для разных вариантов инициализаций, если они все из какой-то выборки. И минимизировать по выборке.

$$\tilde{\mathbf{x}}^*, \tilde{\eta}^* = \arg\min_{\tilde{\mathbf{x}}, \tilde{\eta}} \mathbb{E}_{\theta_0 \sim p(\theta_0)} \mathcal{L}(\tilde{\mathbf{x}}, \tilde{\eta}; \theta_0),$$

Случай линейной регрессии

Дальше заменяется нейронная сеть на случай линейной регрессии и решается вопрос ограничения снизу на размер дистиллированных данных. Тк в данном случае (разбирается квадратичная функция потерь, один шаг градиентного спуска) оптимальное решение можно найти точно, то из него выводится, что размер должен быть хотя бы равен размерности исходных данных.

При этом размерность данных может быть очень большой(например картинки), а также для многих случаев данная оценка неточна.

Больше слоев

Следующим шагом является использование большего числа эпох сети и больше градиентных шагов для обучения каждого.

На каждом слое используются одни и те же дистиллированные данные.

Скорость обучения не фиксирована.

Используется back-gradient optimization чтобы избежать долгих вычислений.

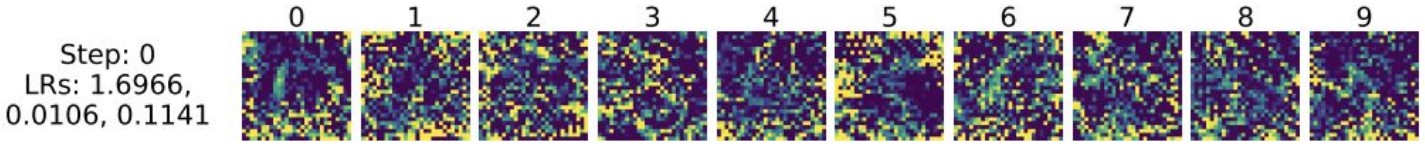
Результаты применения

Применение

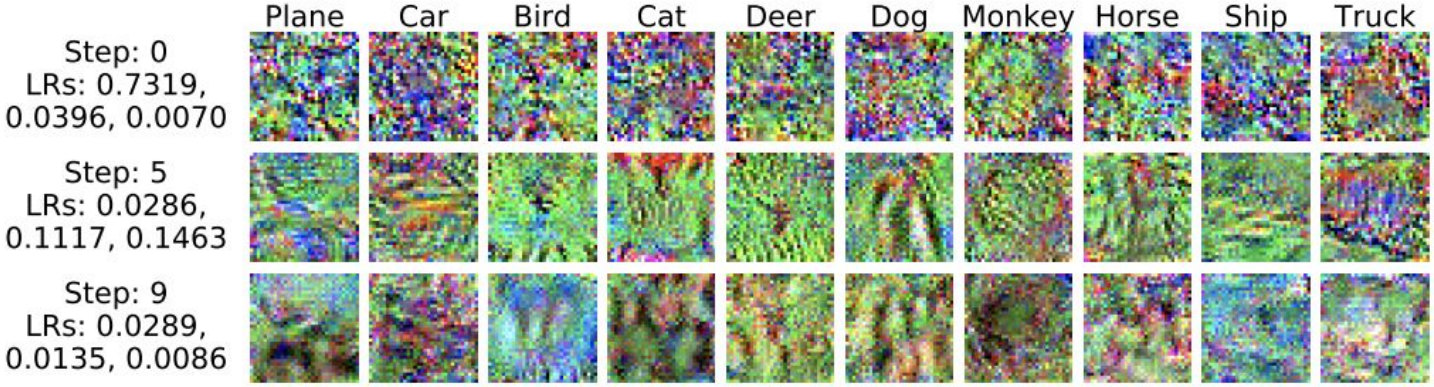
Эксперименты проводились на 4 выборках картинок.

1. Случайно выбираем картинки для категорий.
2. Аналогично выбираем, только смотрим на 20% лучших картинок.
3. Делим картинки на кластеры и центры кластеров создают выборки.
4. Средняя картинка.

Применение



(a) MNIST. Theses distilled images train a fixed initializations from 12.90% test accuracy to 93.76%.



(b) CIFAR10. These distilled images train a fixed initialization from 8.82% test accuracy to 54.03%.

Применение

Заметим, что результаты получились лучше при одинаковом числе шагов оптимизации.

	Ours		Baselines					
	Fixed init.	Random init.	Used as training data in same number of GD steps				Used in K-NN classification	
			Random real	Optimized real	k-means	Average real	Random real	k-means
MNIST	**96.6**	79.5 ± 8.1	68.6 ± 9.8	73.0 ± 7.6	76.4 ± 9.5	77.1 ± 2.7	71.5 ± 2.1	$\mathbf{92.2 \pm 0.1}$
CIFAR10	**54.0**	$\mathbf{36.8 \pm 1.2}$	21.3 ± 1.5	23.4 ± 1.3	22.5 ± 3.1	22.3 ± 0.7	18.8 ± 1.3	29.4 ± 0.3

Применение

А теперь сравниваем качество модели при попытках адаптировать ее с одного массива данных на другой(M-MNIST, U-USPS, S-SVHN).

	Ours w/ fixed pre-trained	Ours w/ random pre-trained	Random real	Optimized real	k-means	Average real	Adaptation Motiian et al. (2017)	No adaptation	Train on **full** target dataset
$\mathcal{M} \to \mathcal{U}$	$\mathbf{97.9}$	$\mathbf{95.4 \pm 1.8}$	94.9 ± 0.8	95.2 ± 0.7	92.2 ± 1.6	93.9 ± 0.8	$\mathbf{96.7 \pm 0.5}$	90.4 ± 3.0	97.3 ± 0.3
$\mathcal{U} \to \mathcal{M}$	$\mathbf{93.2}$	$\mathbf{92.7 \pm 1.4}$	87.1 ± 2.9	87.6 ± 2.1	85.6 ± 3.1	78.4 ± 5.0	89.2 ± 2.4	67.5 ± 3.9	98.6 ± 0.5
$\mathcal{S} \to \mathcal{M}$	$\mathbf{96.2}$	85.2 ± 4.7	84.6 ± 2.1	85.2 ± 1.2	$\mathbf{85.8 \pm 1.2}$	74.9 ± 2.6	74.0 ± 1.5	51.6 ± 2.8	98.6 ± 0.5

Пример алгоритма

Algorithm 1 Dataset Distillation

Input: $p(\theta_0)$: distribution of initial weights; M: the number of distilled data
Input: α: step size; n: batch size; T: the number of optimization iterations; $\tilde{\eta}_0$: initial value for $\tilde{\eta}$

Initialize $\tilde{\mathbf{x}} = \{\tilde{x}_i\}_{i=1}^{M}$ randomly, $\tilde{\eta} \leftarrow \tilde{\eta}_0$
for each training step $t = 1$ to T **do**
    Get a minibatch of real training data $\mathbf{x}_t = \{x_{t,j}\}_{j=1}^{n}$
    Sample a batch of initial weights $\theta_0^{(j)} \sim p(\theta_0)$
    for each sampled $\theta_0^{(j)}$ **do**
        Compute updated parameter with GD: $\theta_1^{(j)} = \theta_0^{(j)} - \tilde{\eta}\nabla_{\theta_0^{(j)}}\ell(\tilde{\mathbf{x}}, \theta_0^{(j)})$
        Evaluate the objective function on real training data: $\mathcal{L}^{(j)} = \ell(\mathbf{x}_t, \theta_1^{(j)})$
    end for
    Update $\tilde{\mathbf{x}} \leftarrow \tilde{\mathbf{x}} - \alpha\nabla_{\tilde{\mathbf{x}}}\sum_j \mathcal{L}^{(j)}$, and $\tilde{\eta} \leftarrow \tilde{\eta} - \alpha\nabla_{\tilde{\eta}}\sum_j \mathcal{L}^{(j)}$
end for

Output: distilled data $\tilde{\mathbf{x}}$ and optimized learning rate $\tilde{\eta}$

Источники

- https://arxiv.org/pdf/1811.10959.pdf
- https://arxiv.org/pdf/1912.07768.pdf