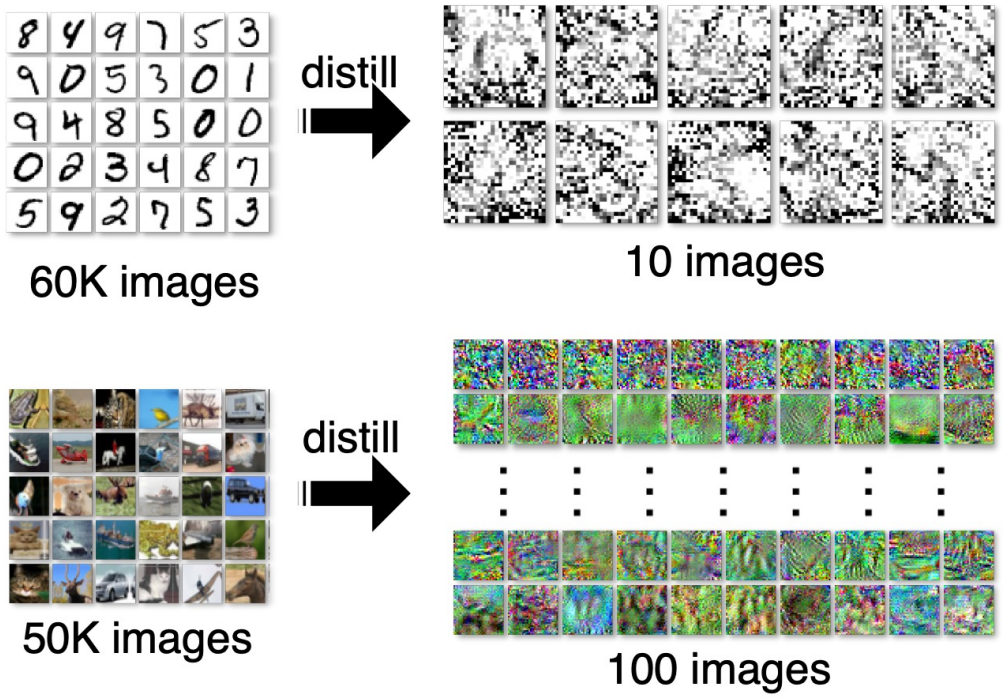
DATASET DISTILLATION

Орлов Александр, 191

Виды дистилляции

- Дистилляция моделей
  - Уменьшить размер модели
  - Ускорить обучение
  - Интерпретируемость
- Дистилляция данных
  - Уменьшить объем датасета
  - Ускорить обучение



Dataset Distillation

Задача:

Сгенерировать небольшой набор синтетических данных, при обучении на которых модель будет за небольшое число шагов градиентного спуска выдавать давать хороший результат на реальных данных

Похожие задачи

- Knowledge distillation:

  Дистилляция моделей, ансамблирование моделей
- Dataset pruning, core-set construction:

  Выбор подвыборки из всего датасета
- Understanding datasets

Идея

- Хотим сгенерировать набор данных $\tilde{\mathbf{x}} = \{\tilde{x}_i\}_{i=1}^{M}$, такой что за один шаг градиентного спуска на этих данных мы получим минимальный лосс
- Для этого представим веса модели после одного шага, как функцию от lr и данных

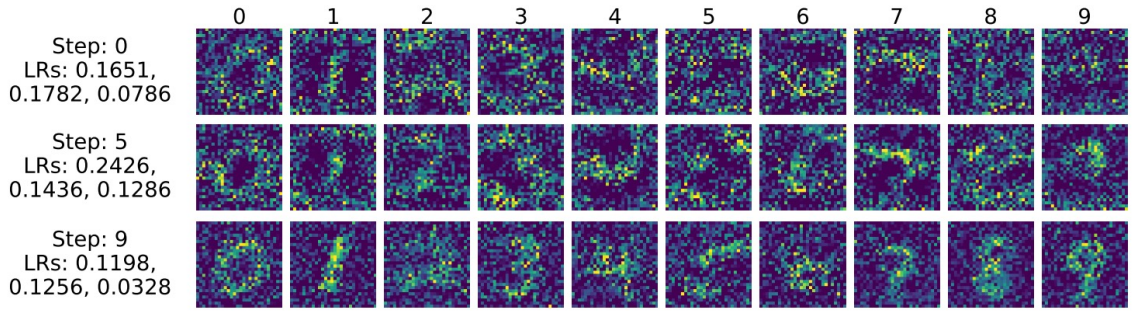
$$\theta_1 = \theta_0 - \tilde{\eta}\nabla_{\theta_0}\ell(\tilde{\mathbf{x}}, \theta_0)$$

$$\tilde{\mathbf{x}}^*, \tilde{\eta}^* = \arg\min_{\tilde{\mathbf{x}}, \tilde{\eta}} \mathcal{L}(\tilde{\mathbf{x}}, \tilde{\eta}; \theta_0) = \arg\min_{\tilde{\mathbf{x}}, \tilde{\eta}} \ell(\mathbf{x}, \theta_1) = \arg\min_{\tilde{\mathbf{x}}, \tilde{\eta}} \ell(\mathbf{x}, \theta_0 - \tilde{\eta}\nabla_{\theta_0}\ell(\tilde{\mathbf{x}}, \theta_0))$$

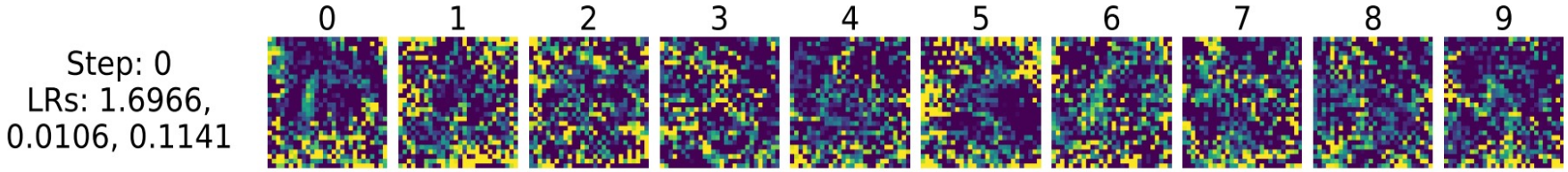
Проблема

- Наше решение зависит от начальной инициализации параметров
- Поэтому мы будем генерировать данные, которые смогут работать с инициализациями из определенного распределения

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



(a) MNIST. These distilled images unknown random initializations to $79.50\% \pm 8.08\%$ test accuracy.



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

Итоговый алгоритм

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 $\tilde{\eta}$

Нижняя граница на размер дистиллированного датасета

- Попытаемся оценить снизу размер синтетического датасета, который будет давать такое же качество, как полный датасет
- Для простоты рассмотрим случай линейной регрессии с квадратичной функцией потерь

$$\ell(\mathbf{x}, \theta) = \ell((\mathbf{d}, \mathbf{t}), \theta) = \frac{1}{2N}\|\mathbf{d}\theta - \mathbf{t}\|^2$$

$$\theta_1 = \theta_0 - \tilde{\eta}\nabla_{\theta_0}\ell(\tilde{\mathbf{x}}, \theta_0) = \theta_0 - \frac{\tilde{\eta}}{M}\tilde{\mathbf{d}}^T(\tilde{\mathbf{d}}\theta_0 - \tilde{\mathbf{t}}) = (\mathbf{I} - \frac{\tilde{\eta}}{M}\tilde{\mathbf{d}}^T\tilde{\mathbf{d}})\theta_0 + \frac{\tilde{\eta}}{M}\tilde{\mathbf{d}}^T\tilde{\mathbf{t}}$$

- Заметим, что такой датасет всегда существует:

$$\tilde{\mathbf{t}} = N \cdot \theta^*$$
$$\tilde{\mathbf{d}} = N \cdot \mathbf{I}$$

Нижняя граница на размер дистиллированного датасета

- Минимум достигается на: $\mathbf{d}^T\mathbf{d}\theta^* = \mathbf{d}^T\mathbf{t}$

$$\mathbf{d}^T\mathbf{d}(\mathbf{I} - \frac{\tilde{\eta}}{M}\tilde{\mathbf{d}}^T\tilde{\mathbf{d}})\theta_0 + \frac{\tilde{\eta}}{M}\mathbf{d}^T\mathbf{d}\tilde{\mathbf{d}}^T\tilde{\mathbf{t}} = \mathbf{d}^T\mathbf{t}.$$

- Для того, чтобы $\tilde{\mathbf{x}} = (\tilde{\mathbf{d}}, \tilde{\mathbf{t}})$ удовлетворял такому условию для любой начальной инициализации, должно выполняться:

$$\mathbf{I} - \frac{\tilde{\eta}}{M}\tilde{\mathbf{d}}^T\tilde{\mathbf{d}} = \mathbf{0},$$

- Отсюда следует, что $M \geq D$

Несколько шагов градиентного спуска

- Метод можно обобщить на произвольное число шагов

$$\theta_{i+1} = \theta_i - \tilde{\eta}_i \nabla_{\theta_i} \ell(\tilde{\mathbf{x}}_i, \theta_i),$$

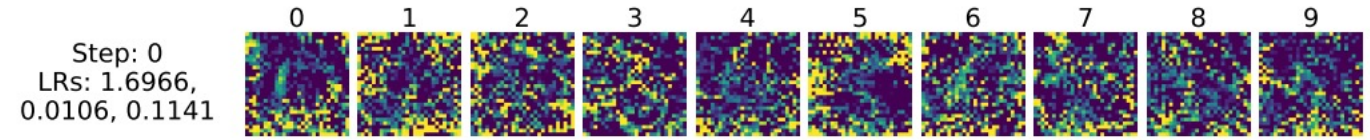
- Бэкпроб по данным и lr через все шаги

Malicious data poisoning

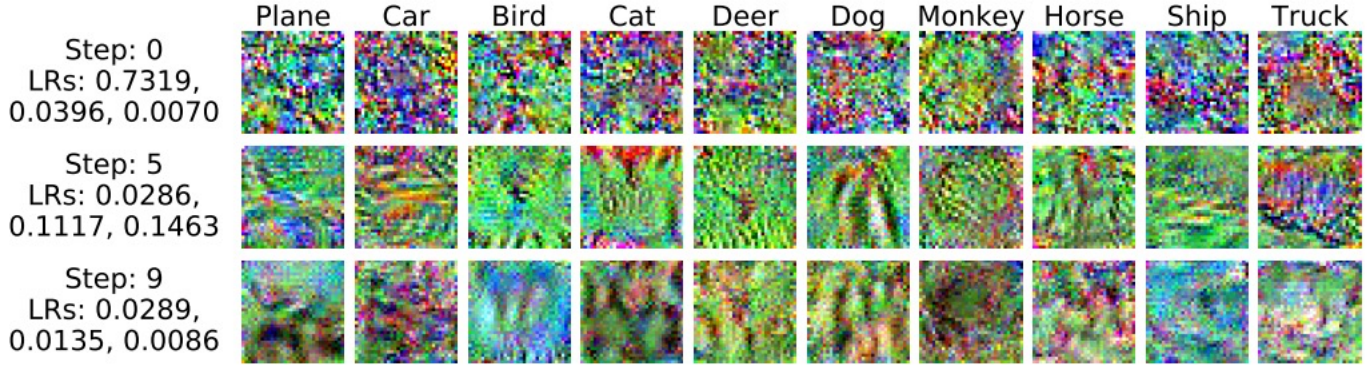
- Немного изменим наш алгоритм, чтобы предобученный классификатор за один шаг градиентного спуска переставал работать на одном из классов
- Для этого мы можем придумать такой лосс, который будет побуждать модель выдавать класс K за класс T

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

Эксперименты



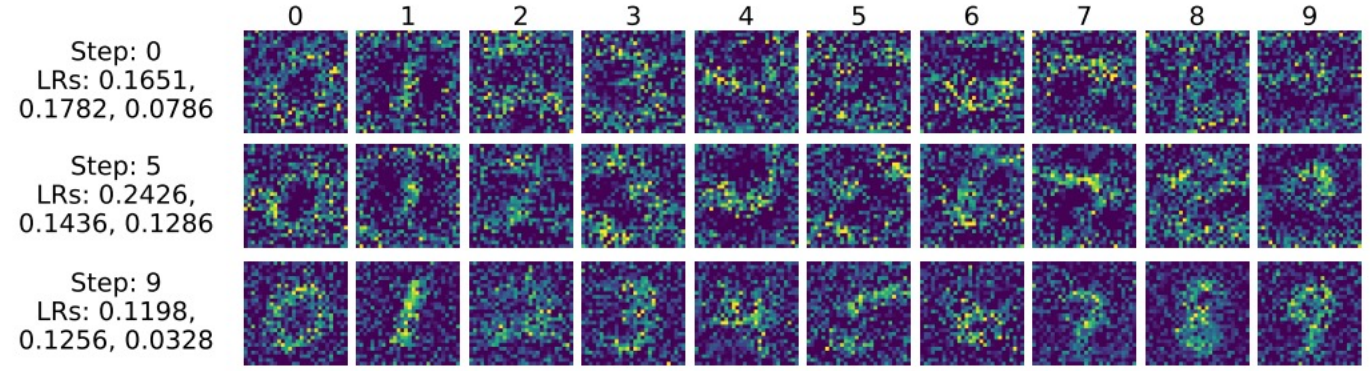
(a) MNIST. Theses distilled images train a fixed initializations from 12.90% test accuracy to 93.76%.



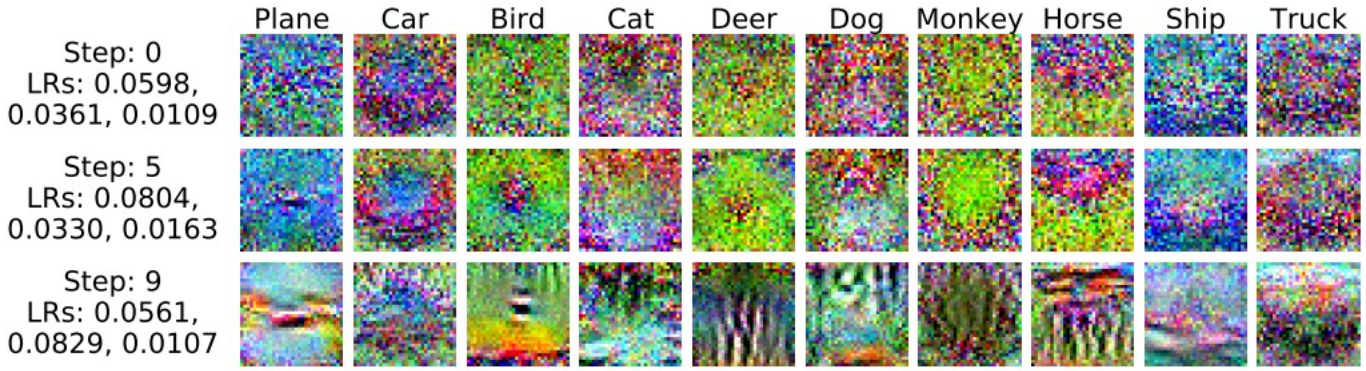
(b) CIFAR10. These distilled images train a fixed initialization from 8.82% test accuracy to 54.03%.

Figure 2: Dataset distillation for fixed initializations: MNIST distilled images use one GD step and three epochs (10 images in total). CIFAR10 distilled images use ten GD steps and three epochs (100 images in total). For CIFAR10, only selected steps are shown. At left, we report the corresponding learning rates for all three epochs.

Эксперименты



(a) MNIST. These distilled images unknown random initializations to $79.50\% \pm 8.08\%$ test accuracy.



(b) CIFAR10. These distilled images unknown random initializations to $36.79\% \pm 1.18\%$ test accuracy.

Figure 3: Distilled images trained for *random initialization* with ten GD steps and three epochs (100 images in total). We show images from selected GD steps and the corresponding learning rates for all three epochs.

Дистилляция данных на практике

- Подбор архитектуры сети
- Ускорение обучения
- Перенос знаний о данных между моделями

Подбор архитектуры нейросети

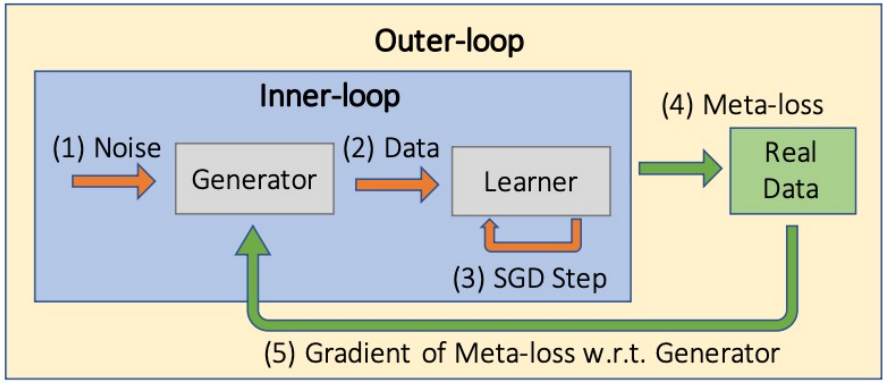
- Обучать модель, предсказывающую, насколько хорошо будет работать новая модель, путем экстраполяции ранее обученных архитектур
- Использовать общие веса для разных архитектур (shared weights)

Generative Teaching Networks

- Процедура для генерации батчей синтетических данных
- Генерирует данные, подходящие под задачу
- Состоит из двух вложенных циклов:
  - Внутреннего цикла, в котором происходит обучение модели
  - Внешнего, где происходит генерация синтетических данных
- Генерирует синтетические данные, которые зависят от архитектуры сети

Generative Teaching Networks

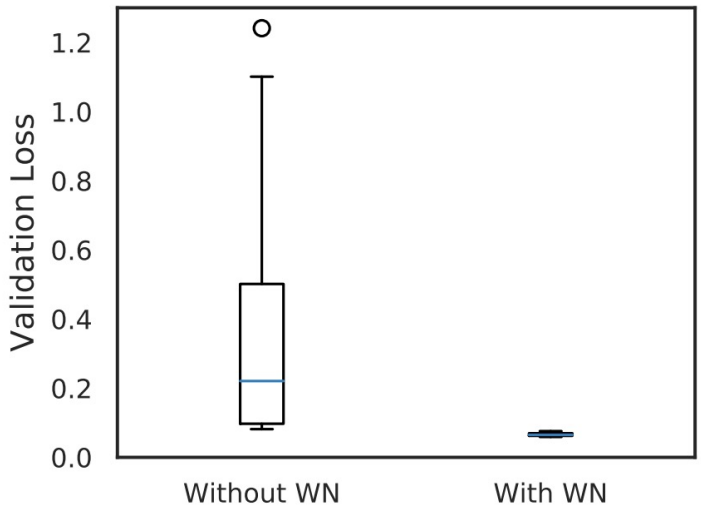
1. GTN на вход подается шум
2. Модель делает шаг градиентного спуска на батче данных от генератора
3. Считается “мета-лосс” модели на реальном датасете
4. Бэкпроб и оптимизация генератора по “мета-лоссу”



(a) Overview of Generative Teaching Networks

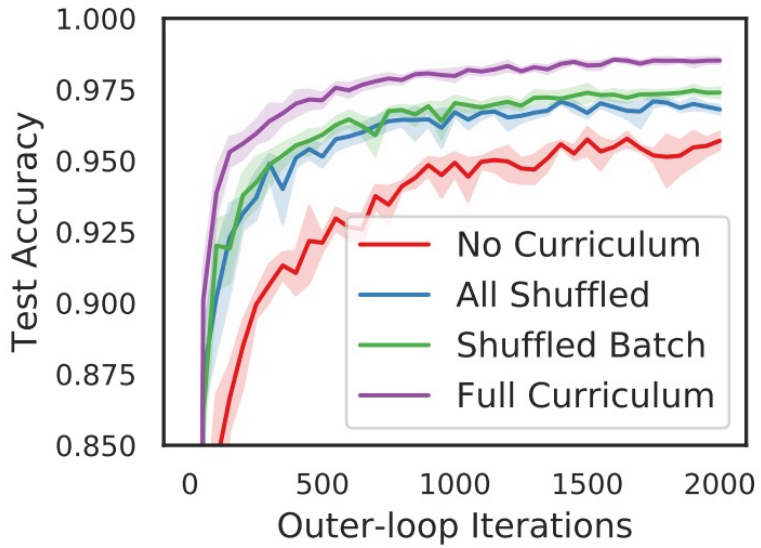
Weight Normalization

- Оптимизация через мета-лосс нестабильна
- Спустя некоторое количество шагов внутреннего цикла алгоритм начинает расходиться
- Спасает нормализация весов моделей $W = g \cdot V/\|V\|$

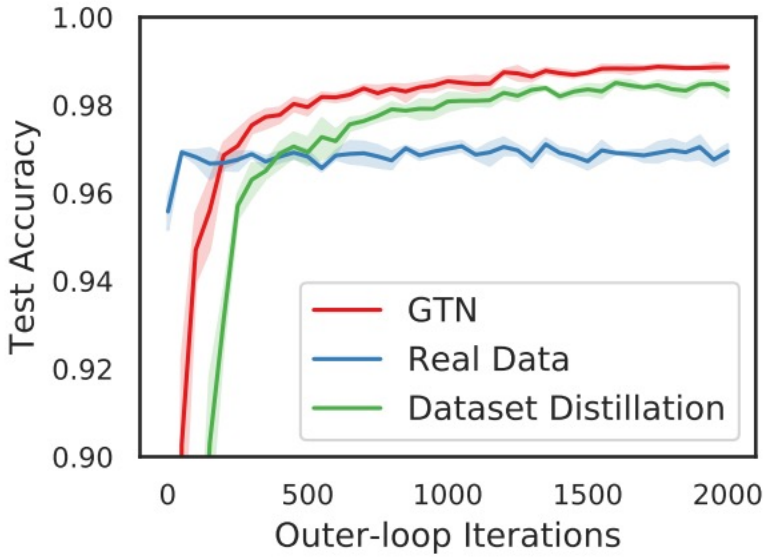


Curriculum learning

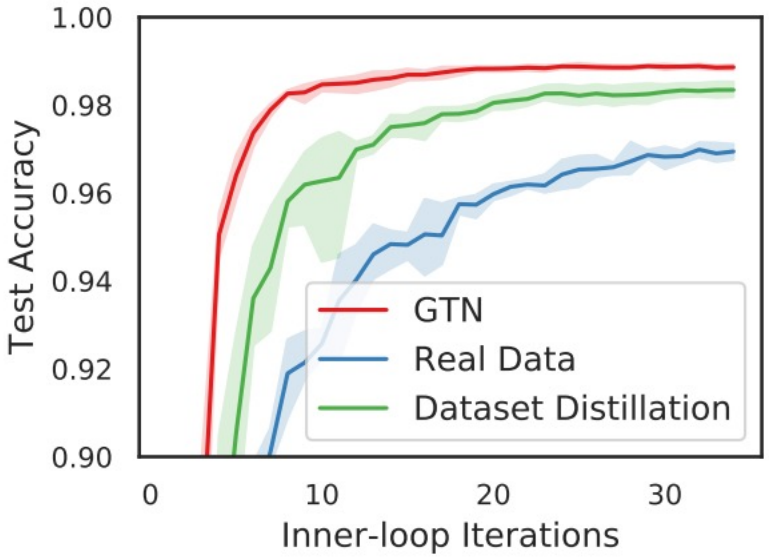
- Авторы предлагают обучать не только веса генератора, но и его входы
- За счет этого генератор переходит от абстрактных синтетических данных к более конкретным примерам



Эксперименты



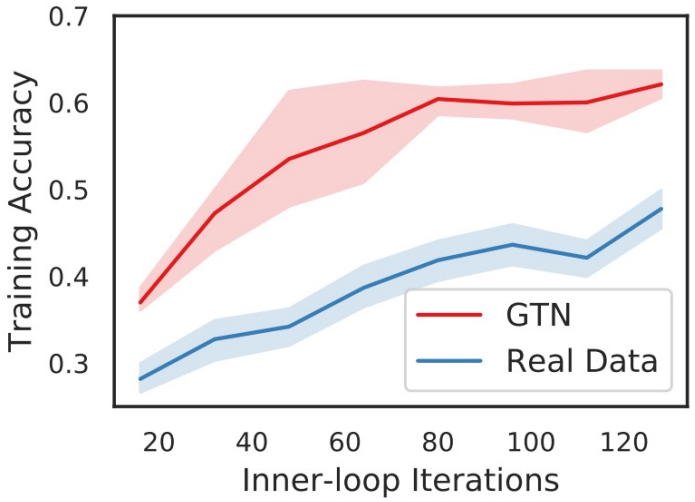
(a) Meta-training curves



(b) Training curves

(c) GTN-generated samples

Эксперименты



(a) CIFAR10 inner-loop training

(b) CIFAR10 GTN samples

Эксперименты

Model	Error(%)	#params	GPU Days
Random Search + GHN (Zhang et al., 2018)	4.3 ± 0.1	5.1M	0.42
Random Search + Weight Sharing (Luo et al., 2018)	3.92	3.9M	0.25
Random Search + Real Data (baseline)	3.88 ± 0.08	12.4M	10
Random Search + GTN (ours)	**3.84** ± 0.06	8.2M	0.67
Random Search + Real Data + Cutout (baseline)	3.02 ± 0.03	12.4M	10
Random Search + GTN + Cutout (ours)	**2.92** ± 0.06	8.2M	0.67
Random Search + Real Data + Cutout (F=128) (baseline)	2.51 ± 0.13	151.7M	10
Random Search + GTN + Cutout (F=128) (ours)	**2.42** ± 0.03	97.9M	0.67

Спасибо за внимание!