Flow Matching with Minibatch OT

Яковчук Николай*
Higher School of Economics
@glavapride

Шаламкова Алиса*
Higher School of Economics
@merray

Abstract

Обучение модели Flow Matching в двух режимах: без и с конструированием пар с помощью решения задачи OT между батчами шума и данных [2]. Сравнение дисперсии лосса/градиента, кривых и скорости обучения на задачах генерации. Используемый датасет: CIFAR-10 [3]. Используемая модель: EDM.

* - Мы сидели на созвоне и все вместе делали (п. 5)

1 Background

Как обычно, мы будет работать с задачей генерации с точки зрения плотностей. Нашей задачей будет, имея распределения над $\mathbb{R}^d$, построить f, которая будет переводить плотность q_0 в q_1. Имеем: пусть $x \sim q_0$, тогда $f(x) \sim q_1$. Мы будем пользоваться несколькими теоремами и предложениями, доказательства которых приведены в статье Tong A., Fatras K., Malkin N: Improving and Generalizing Flow-Based Generative Models with Minibatch Optimal Transport [2]. Приведем некоторые из них в более простой формулировке.

1. Пусть динамика $p_t(x) = \int p_t(x|z)q(z)dz$, где $q(z)$ - какое-то распределение над condition переменными. Если $p_t(x|z)$ генерируется $u_t(x|z)$ из изначальных состояний $p_0(x|z)$, тогда векторное поле $u_t(x) := \mathbb{E}_{q(z)}\frac{u_t(x|z)p_t(x|z)}{p_t(x)}$ генерирует динамику $p_t(x)$. Теорема: $u_t(x)$ генерирует динамику $p_t(x)$ из $p_0(x)$.

2. Предложение: u_t решает задачу динамического оптимального транспорта между q_0 и q_1, где q_0 и q_1 - маргинальные распределения $q(z) = q(x_0, x_1), q(z) := \pi(x_0, x_1)$, где $\pi(\cdot, \cdot)$ - план.

2 Introduction

2.1 Flow Matching

Пусть есть $p_t(x)$ - динамика и $u_t(x)$ - векторное поле, которое ее генерирует. Пусть $v_\theta(\cdot, \cdot) : [0, 1] \times \mathbb{R}^d \to \mathbb{R}^d$ - векторное поле / наша нейронная сеть, параметризованная весами θ. Тогда есть суть Flow Matching:

$$\mathbf{L}_{FM}(\theta) := \mathbb{E}_{t\sim U(0,1), x\sim p_t(x)}||v_\theta(t, x) - u_t(x)||^2.$$

Аналогично формулируется случай Conditional Flow Matching:

$$\mathbf{L}_{CFM}(\theta) := \mathbb{E}_{t\sim U(0,1), x\sim p_t(x|z)}||v_\theta(t,x) - u_t(x|z)||^2.$$

В статье [2] дополнительно приводится теорема, которая говорит, что если плотность существует для любой точки $\mathbb{R}^d$ то функционалы FM и CFM эквивалентны.

Псевдокод для CFM:

Algorithm 1 Conditional Flow Matching

```
if Дано: q(z), p_t(x|z)u_t(x|z), v_θ - нейросеть then
    while Training do
        z ~ q(z); t ~ U(0,1); x ~ p_t(x|z)
        L_CFM(θ) ← ||v_θ(t,x) − u_t(x|z)||^2
        θ ← Update(θ, ∇_θ L_CFM(θ))
return v_θ
```

2.2 Minibatch OT

Для больших данных подсчет транспортного плана π может быть сложным, поэтому мы вводим minibatch OT аппроксимацию. Minibatch OT может привести к ошибке и не дать достаточно точное приближение, но сильно упрощает время подсчета. Для каждого батча $(\{x_0^{(i)}\}_{i=1}^B, \{x_1^{(i)}\}_{i=1}^B)$ будем семплировать пары из совместного распределения π_{batch}, заданного планом между сорс и таргет точками в батче. Когда OT batch size равен "размеру"(q_0, q_1), мы восстанавливаем точное значение OT, следовательно по предложению (2) из Background выучиваем решение. Далее полагая, что интерполянт - линейный:

Algorithm 2 Conditional Flow Matching with Minibatch OT

```
if Дано: q_o, q_1, σ, batch_size, v_θ - нейросеть then
    while Training do
        Sample batches of size batch_size i.i.d from datasets
        x_0 ~ q_0(x_0); x_1 ~ q_1(x_1)
        π ← OT(x_1, x_0)
        (x_0, x_1) ~ π
        t ~ U(0,1)
        μ_t ← t x_1 + (1 − t) x_0
        x ~ N(μ_t^2, σI)
        L_CFM(θ) ← ||v_θ(t,x) − (x_1 − x_0)||^2
        θ ← Update(θ, ∇_θ L_CFM(θ))
return v_θ
```

3 Experiments

3.1 General

Мы будем использовать датасет CIFAR-10 с 10 классами животных, транспорта и т.д., 50 тысяч изображений. Разрешение картинок во всех экспериментах используется 32x32. Из препроцессинга использовали нормализацию. Выбранный батч сайз - 64/32, обучение моделей проводилось на V100/2xT4/T4. Выбранная модель - EDM (ссылка на репозиторий непосредственно в описании проекта).

3.2 Conditional Flow Matching

Как было сказано ранее, использовалась модель EDM, оптимайзер - Adam с $lr = 5 * 10^{-4}$, как было показано в исходной статье. Дополнительно был взят $weight_decay = 0.001$, поскольку где-то на 300 эпохе лосс взрывался. Сэмплирование с помощью эйлера: $eps = 10^{-3}$. Обучение продлилось около 6.5 часов. Мы дополнительно обучили модель на данных без нормализации, получалось добиться большей стабильности лосса и меньших его значений, но качество семплов при этом было хуже.

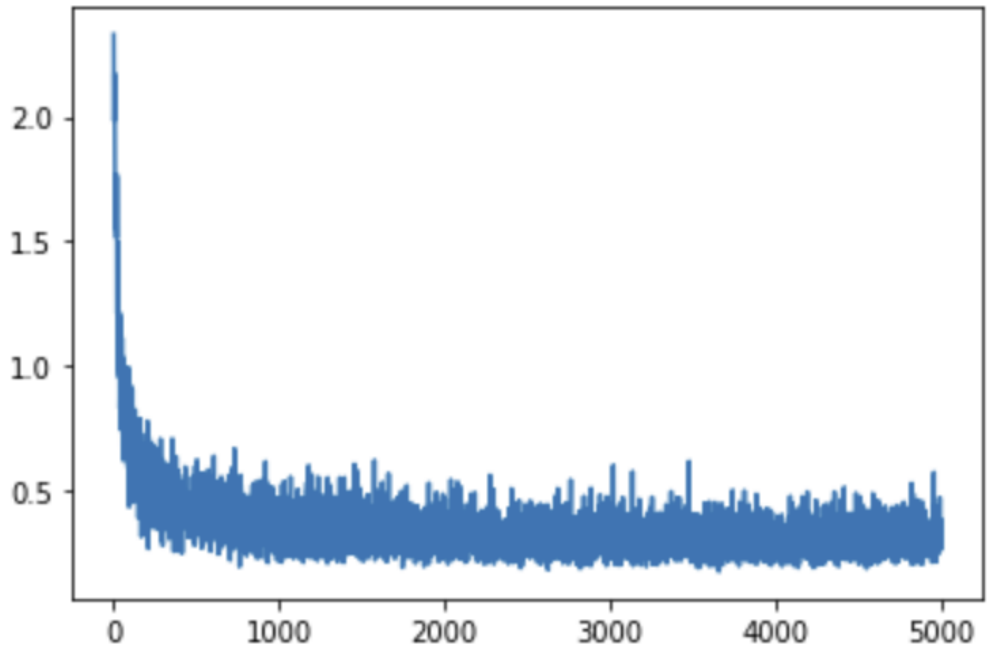

Рис. 1: CFM Loss

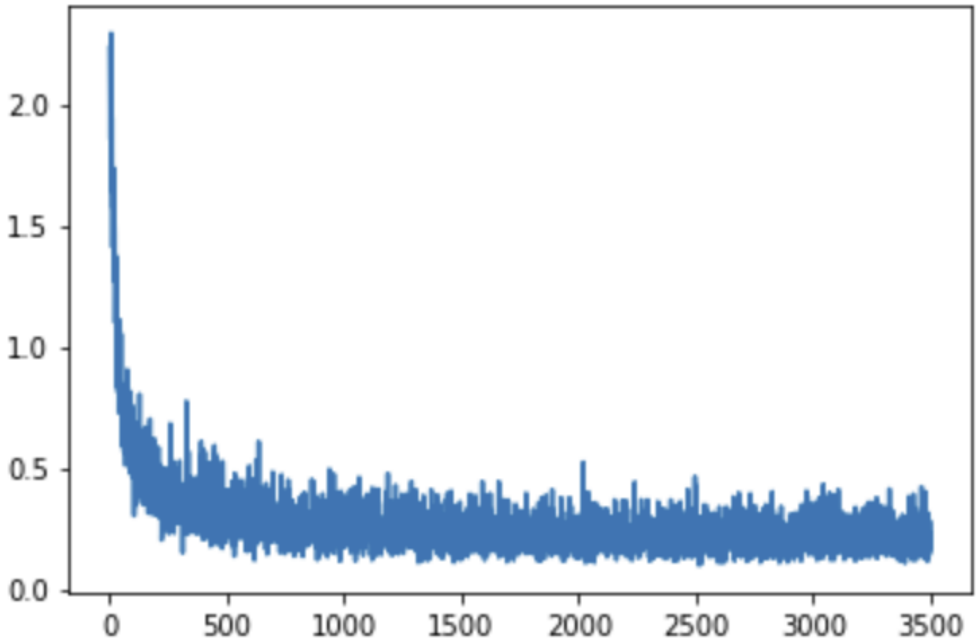

Рис. 2: CFM Loss w/o norm

Рис. 3: CFM samples 1

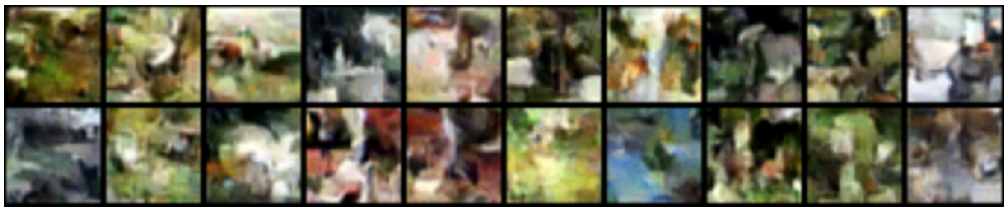

Рис. 4: CFM samples 2

В разных комбинациях ставились эксперименты:
1) кол-во эпох 500, 1500, 5000
2) без/с weight_decay. В оригинальной статье справились без него, но у них и вычислительных ресурсов было больше - они обучали модель с batch_size=128, что наверняка уменьшало нестабильность модели. Нам же нужно было защищаться от этого по-другому
3) с/без нормализацией и переворотом картинок по горизонтали. Показалось, что нормализация убивает некоторые особенности картинок, практически смешивая их в черный фон

Рис. 5: CFM samples 3 (w/o norm)

4) почти сразу и везде использовали клипирование градиентов (clip_grad_norm с параметром max_norm=1.0)) опять же в надежде предотвратить взрыв градиентов 5) для уменьшения дисперсии лосса и стабилизации обучения в целом написали обёртку Exponential Moving Average. Разницы в сэплах особой не заметили, но построили графики лоссов для самой модели и её EMA пары, получили вот такие результаты:

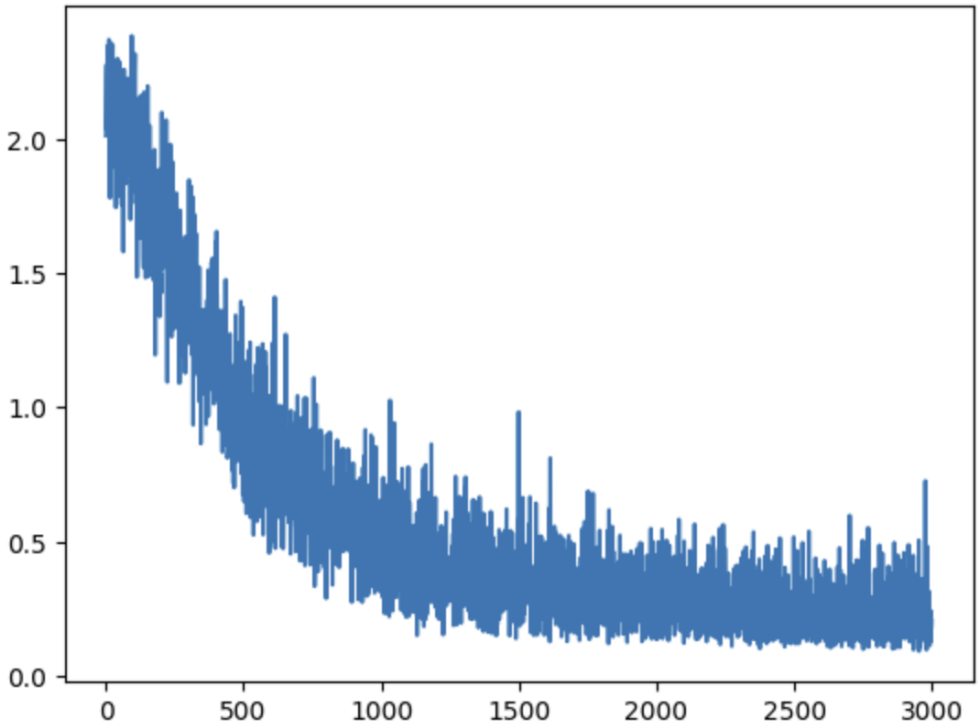

Рис. 6: EMA loss

По этом графику понятно, что до 1500 тысяч итераций лосс уверенно падает и обучение надо продолжать.

3.3 OT - Conditional Flow Matching

Как было сказано ранее - использовалась модель EDM. Батчсайз 32, обучение на 2xT4, $weight_decay = 0.001$, оптимайзер - Adam с $lr = 5 * 10^{-4}$. Обучали около 2х часов.

Рис. 7: OT-CFM samples

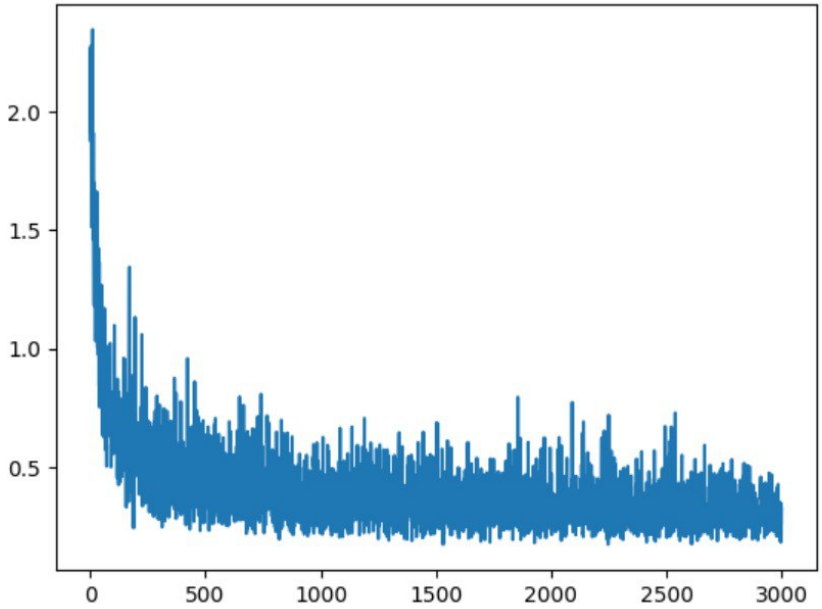

Рис. 8: OT-CFM Loss

На этапе обучение на 1500 итерации семплы из OT-CFM были очень сглаженными по сравнению с сэплами из CFM. Ранее мы наблюдали увеличение сглаженности сэмпла при увеличении количества эпох при обучении CFM, так что мы аккуратно сделали вывод, что OT-CFM учится быстрее - на более раннем этапе сэмплы обладают качеством, присущим более поздним эпохам. Из этого можно сделать вывод, что если цель задачи - ускорить процесс обучения, то стоит выбирать OT-CFM. В скорости семплирования особой разницы замечено не было.
Отметим также (рис. 9), что при сравнении графиков OT-CFM и CFM видно, что дисперсия лосса OT-CFM выше. Это подтверждает слова из статьи о том, что из-за использования minibatch'ей обучение может быть менее стабильным, но зато более быстрым.

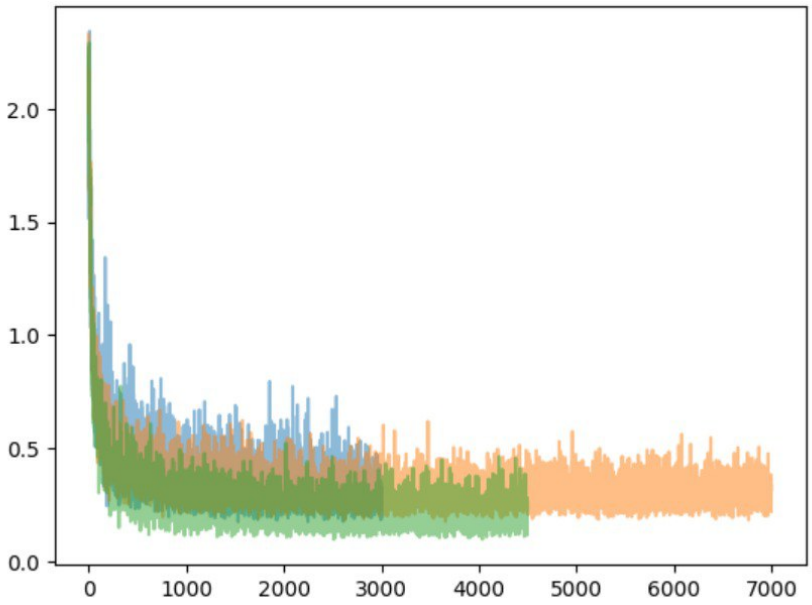

Рис. 9: Oran - cfm, green - cfm w/o norm, blue - OT-cfm

3.4 Общие наблюдения о сэмплировании

В работе мы изучили альтернативные самописному способу сэмплирования, в частности dopri5. Мы заметили, что при обучении CFM на 2000 итерациях и сэмплировании данным методом картинки обладают артефактами, но общие идеи изображений улавливаются. Так, на рис. 10 изображены машины, что видно на самой первой картинке.
При сэммплировании по схеме эйлера найти однозначные соответствия между лейблом и сэмплом удавалось с трудом. Тем не менее, часто общие свойства объекта совпадали с изображенным на картинке (например, лягушка была зеленой, что видно в примерах ниже). При сэмплировании из одинакого шума заметных различий в работе методов обнаружено не было.

3.5 Примечание

Предполагаем, что для достижения принципиально новых результатов и более аккуратного сэмплинга необходимо дольше обучать модель, что было затруднительным при наших ресурсах. Мы бы также попробовали адаптивные learning rate для уточненного обучения.

4 Доп. задача для переноса стиля

В ноутбуке paired_5_6.ipynb сделали бонусное задание о задаче переноса стиля, выбрали сделать из пятёрки шестёрку. Результаты оказались неплохими: модель понимает, что какой-то конец цифры нужно закрутить. При решении задачи воспользовались идеей paired dataset из 8 семинара.

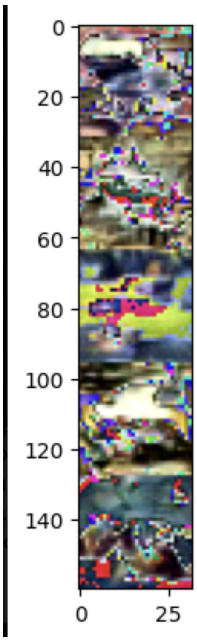

Рис. 10: dopri5 sampling

Список литературы

[1] Ракитин Д.: Генеративные модели на основе диффузии (2024)

[2] Tong A., Fatras K., Malkin N: Improving and Generalizing Flow-Based Generative Models with Minibatch Optimal Transport (2024)

[3] Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009

5 Так называемая Сота

Здесь мы прикрепляем самые милые семплы, которые удалось получить, вашему вниманию:

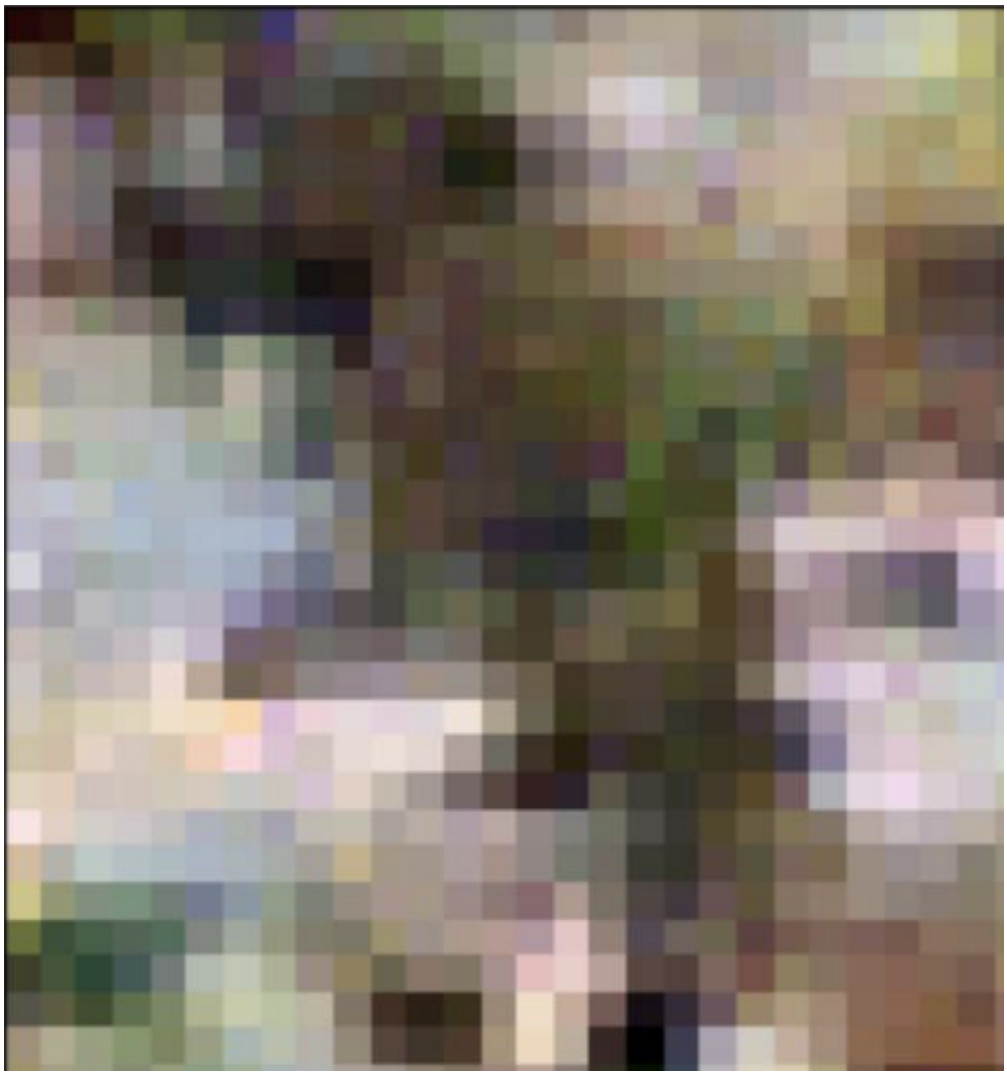

Рис. 11: Собака

Рис. 12: Лягушка

Рис. 13: Олень

Рис. 14: Котик

Рис. 15: Котик 3 в плаще (Truck)