Практическое задание 1. Создание изображений с помощью генеративно-состязательных нейронных сетей

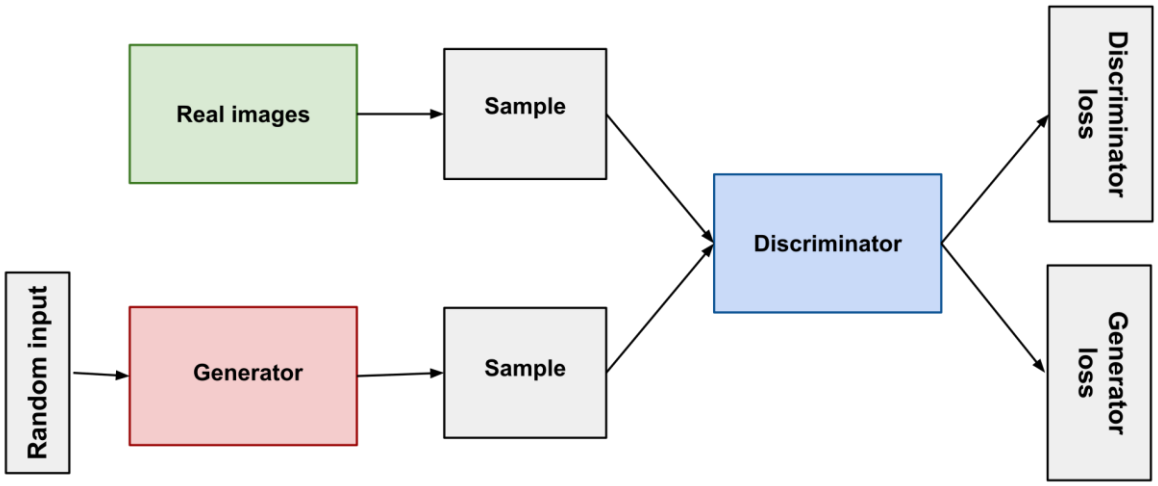
Цели задания

- ознакомление с принципами работы генеративно-состязательных сетей;
- применение языка Python и изучение одного из фреймворков машинного обучения;
- получение навыков создания и обучения GAN'ов, предназначенных для создания изображений;
- выбор оптимальной архитектуры для генерации изображений заданного домена;
- освоение различных методов для оценки качества получаемых изображений;

Введение

Генеративно состязательные нейронные сети

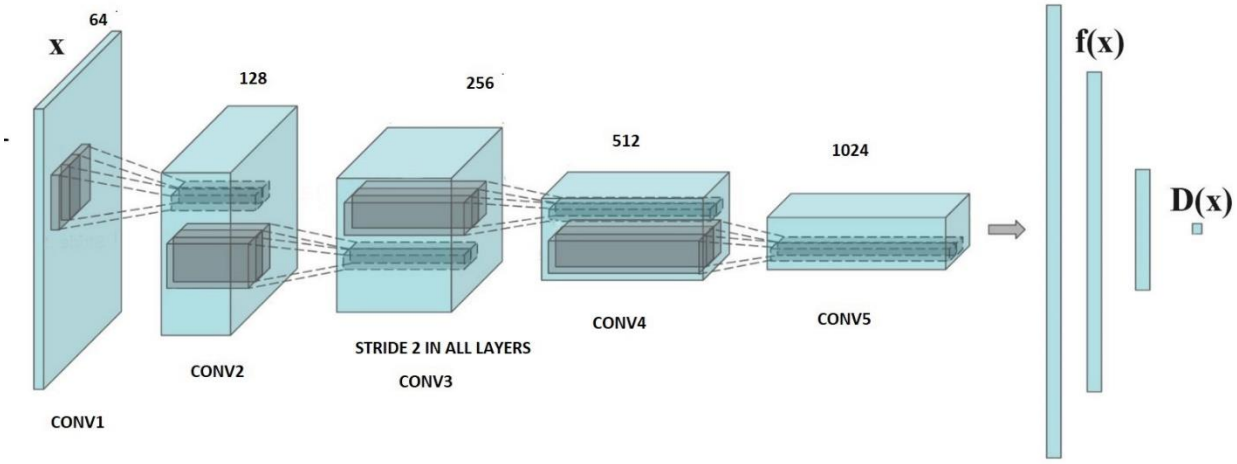
Генеративно-состязательная нейросеть (Generative adversarial network, GAN) — архитектура, состоящая из двух независимых нейронных сетей, настроенных на работу друг против друга: **генератора** и **дискриминатора**. Задача генератора – создавать изображения из случайного шума, подаваемого на вход сети. Задача дискриминатора – отличать реальные изображения от изображений, которые создал генератор. В процессе обучения генератор учится создавать изображения всё более похожие на реальные, в то время как дискриминатор старается находить новые признаки для того, чтобы их различать.



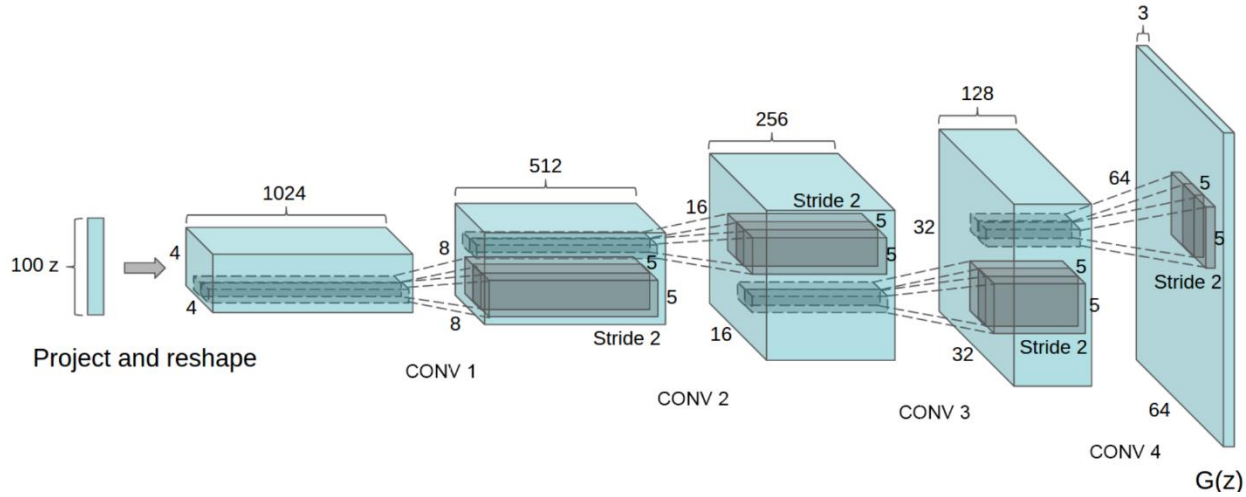
Архитектуры генератора и дискриминатора

В настоящий момент одной из наиболее популярных архитектур для создания изображений является глубокая свёрточная генеративно-состязательная сеть (DCGAN). Как следует из названия, генератор и дискриминатор в такой архитектуре являются глубокими свёрточными сетями.

Дискриминатор получает на вход изображение и при помощи свёрточных слоёв извлекает признаки изображения, которые затем используются для обучения обычного полносвязного слоя (или нескольких слоёв).



Генератор же должен из одномерного вектора создать объёмное изображение, а потому использует слои транспонированной свёртки (transposed convolution), также называемые деконволюционными (deconvolution layer), благодаря чему на выходе сети получается 3-D набор чисел, формирующих изображение.



Методы оценки качества модели

После того, как генеративно-состязательная сеть обучена, необходимо оценить качество генерируемых изображений. Самым простым методом является оценка внешнего вида изображений на предмет вариативности и

реалистичности, однако это довольно субъективная оценка. Для того, чтобы получить количественную оценку в настоящий момент наиболее часто используется метрика *Frechet Inception Distance (FID)*.

Данная метрика была специально разработана для оценки производительности генеративно-состязательных сетей и активно используется в различных соревнованиях. Данная метрика оценивает, насколько близки два набора изображений и выдаёт одно число. Поскольку метрика является расстоянием, то чем полученное значение меньше, тем сильнее похожи наборы изображений, а значит тем лучше модель генератора. Подробнее про данную метрику и её реализацию на языке Python3 можно посмотреть в статье.

Примеры изображений из предлагаемого набора данных

Постановка задачи

Скачать набор данных с изображениями. Выбрать архитектуру для генератора и дискриминатора, а также другие гиперпараметры. Создать генеративно-состязательную сеть, обучить её на скачанных изображениях, а затем оценить её качество с помощью FID метрики и визуально на предмет реалистичности/наличие артефактов.

Основные требования к реализации задания

- язык программирования: Python3;
- использование Typing и Docstring в коде;
- рекомендуем придерживаться руководства PEP8 при написании кода;
- набор изображений: Cat faces dataset;

Ожидаемые выходные данные

- архив, содержащий 10000 сгенерированных изображений;
- файл модели генератора для воспроизведения результатов (или полный скрипт, позволяющий выполнить генерацию изображений, если не планируется использование Keras моделей);
- .py скрипт или Jupyter notebook для создания и обучения сети с requirements.txt и описанием проделанной работы и сделанными выводами;

Внимание! Размер изображений довольно велик, а потому обучение моделей может длиться довольно долго. Особенно если вы планируете обучать модели на CPU. Настоятельно рекомендуем по возможности производить обучение на видеокарте.

Пожалуйста, не откладывайте выполнение задания на последний день, так как, даже быстро выбрав первую попавшуюся модель, вы можете не успеть обучить её и отправить результаты.

Полезные ссылки:

- GAN — What is Generative Adversarial Networks GAN
- Generative Adversarial Network(GAN) using Keras
- Deep Convolutional Generative Adversarial Network
- How to Train a GAN? Tips and tricks to make GANs work
- FID метрика и как её реализовать
- Генеративно-состязательная нейросеть (GAN). Руководство для новичков
- Gan loss functions