StyleGAN by NVIDIA

Генеративное-Состязательные сети (GAN) - это принцип, позволяющий обучить генератор генерировать хорошие похожие на настоящие данные.
Он состоит из двух сетей: генератора, который генерирует новые выборки, и дискриминатора, который обнаруживает ложные примеры. Генератор пытается обмануть дискриминатор, в то время как дискриминатор пытается отличить примеры, синтезированные генератором, от настоящих.
После обучения генератор может быть использован для создания новых примеров. Обычно на вход генератору подается некоторый шум, из которого уже синтезируется изображение. Таким образом, у таких моделей полностью отсутствует возможность контролировать признаки сгенерированного изображения. Кроме того, обычно GAN-ы не могут генерировать высокоточные изображения и исследователи в последнее время активно пытаются избавиться от этой проблемы. В основном, они изменяют дискриминатор, делая его более сложным, так как если дискриминатор начнет отличать фейк от реальных данных лучше, то и генератору придется генерировать более правдоподобные данные.

В StyleGAN архитектура генератора переделывается так, что открываются новые способы контролировать процесс генерации изображения, дискриминатор же они не трогают. По сути архитектура пытается отделить высокоуровневые атрибуты изображения такие как положение лица, личность человека от случайных факторов таких как прическа и прочее.

Mapping network
У данной архитектуры кроме входа, из которого синтезируется изображение, есть еще один вход, в который подается шум z. Этот z при помощи обучаемой mapping network, преобразуется в вектор w, значения которого отвечают за разные визуальные концепции. Проблема, которую авторы пытаются решать таким методом, называется features entanglement. Она состоит в том, что входной вектор у GAN-а должен подчиняться распределению обучающих данных. Например, если в датасете большое количество людей с черными волосами, то большинство входных векторов будет мапиться в данный признак. Однако, при использовании отдельной нейросети для преобразования входного вектора z модель может генерировать векторы, которые не обязаны подчиняться закону распределения обучающих данных.
Данная mapping network состоит из последовательности Dense слоев с нелинейностями (всего 8 слоев) и возвращает на выходе вектор w такой же размерности, что и вход z. Позже полученный вектор w будет использоваться в каждом из сверхточных слоев в генераторе для контроля операции нормализации.

Нормализация входов на сверточных слоях
Каждый вход сверточного слоя нормализуется с помощью операции AdaIN (adaptive instance normalization) с использованием вектора “стиля” w. При такой нормализации информация о стиле проходит в генератор. Операция состоит из нескольких частей. Сначала каждый канал изображения нормализуется. После этого вектор “стиля” w трансформируется при помощи полносвязного слоя в масштаб (scale) и смещение (bias) для каждого канала. Таким образом, получим вектор y = [y_s, y_b], размерность которого равна удвоенному количеству каналов (на каждый канал по смещению и масштабу). Полученный вектор масштабов и смещений сдвигает каждый канал тензора, полученного на выходе предыдущей свертки.

$$AdaIN(x_i, y) = y_{s,i}\frac{x_i - \mu(x_i)}{\sigma(x_i)} + y_{b,i}$$

Отсутствие шума на входе
В отличие от остальных GAN-ов, вместо шума на вход генератору здесь подается обучаемая константная матрица размера 4x4x512. Авторы заметили, что это нисколько не ухудшает генерацию, так как все высокоуровневые фичи и так уже зашиты в вектор стиля w.

Добавление случайного шума в генератор.
Также в генератор встраивается дополнительный шум для генерации случайных элементов на изображениях. Этот шум — просто одноканальное изображение, состоящее из некоррелированного гауссова шума. Шум подается перед каждой операцией AdaIN на каждый слой. Более того, существует фактор масштаба для шума, который обучается для каждого канала и шум домножается на масштаб перед добавлением к каналу изображения. Авторы показывают, что без добавления шума картинки тоже получаются неплохими. Однако при добавлении шума на сгенерированных изображениях появляется всё больше новых мелких и хорошо прорисованных деталей, а также улучшается качество изображения. Здесь стоит пояснить, что ранее проводились исследования, которые показывают, что именно пространственно-инвариантные статистики (такие матрица Грамма, среднее по каналу, дисперсия и прочее) кодируют стиль изображения, в то время как пространственно меняющиеся функции кодируют конкретный экземпляр.

Mixing regularization
На каждом слое в генератор подается один и тот же вектор стиля w, в которому применяется лишь одно афинное преобразование (dense слой). Таким образом, сеть может выучить, что векторы стиля коррелируют между собой. Чтобы избежать этого при обучении некоторый процент картинок генерируется с помощью двух разных латентных векторов z. Для этого сначала рандомно

генерируются z1 и z2 , которые мапятся в w1 и w2. Далее сначала до некоторого момента времени в генератор подается w1, после него - w2.

Трюк усечения в пространстве W (truncation trick)
Рассматривая распределение обучающих данных можно понять, что районы с маленькой плотностью плохо представляются и генератору сложно их выучить. Поэтому изображения, полученные из латентного вектора z, имеющего маленькую плотность, будут не очень реалистичны. Чтобы избежать этой проблемы латентные вектора при генерации часто берутся из усеченного пространства. Авторы статьи предложили метод усечения, который работает в пространстве W. Идея метода проста. Подсчитаем сначала центра масс пространства W : $\overline{w} = E_{z \sim P(z)} f(z)$. Данные центр масс является средним лицом - лицом, которое похоже на многие изображения из обучающих данных. После этого имея вектор w, масштабируем его отклонение от центра : $w' = \overline{w} + \phi(w - \overline{w})$, где , где $\phi < 1$.

Результаты
Модель является state-of-the-art результатом на двух датасетах. Сгенерированные картинки получаются очень реалистичными и, более того, теперь получилась возможность и контролировать фичи изображения путем семплирования подходящих векторов w. Это также позволяет делать классные видео, где человеческие лица плавно перетекают друг в друга.