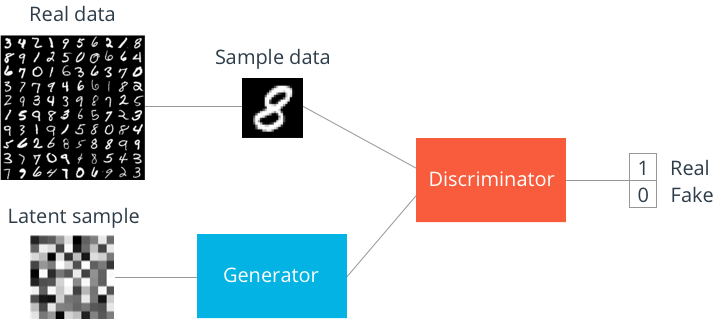
Отчёт по индивидуальному проекту

GAN – генерация изображений «рукописных» цифр

Снапков Максим

Задача нейронной сети заключается в генерации изображений с «рукописными» цифрами.

Изображения, использующиеся для обучения сети, являются частью датасета MNIST, состоящего из серых полутоновых изображений 28×28 пикселей рукописных цифр. Этот датасет содержит 60000 примеров для обучения и 10000 примеров для тестирования.

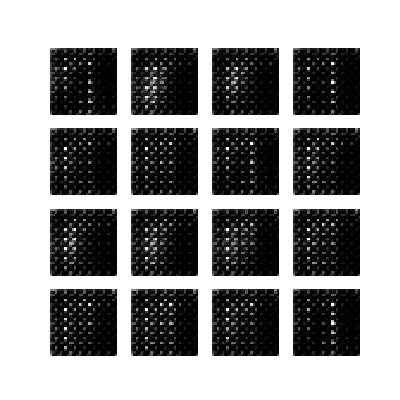
В GAN две модели учатся одновременно в состязательном процессе:  
Генератор учится создавать изображения, которые похожи на рукописные цифры, а Дискриминатор учится отличать реальные изображения от искусственных. Процесс достигает равновесия, когда Дискриминатор уже не может отличить реальные изображения от подделок. 

Архитектура подсетей

|  |  |  |  |
| --- | --- | --- | --- |
| Генератор: | | | |
| Dense | | Полносвязный слой, количество нейронов = 7\*7\*256, без вектора смещения, размерность входного пространства – (\*,100) | |
| BatchNormalization | | Слой, нормализующий свои входы | |
| LeakyReLu | | Активационный слой | |
| Reshape | | Реформатирует выход в форму:  (None, 7, 7, 256) | |
| Conv2DTranspose | | Транспонированный свёрточный слой,   количество карт признаков = 128,  высота и ширина ядра свёртки = 5,  шаги свёртки по высоте и ширине = 1,  на выходе получается изображение с разрешением исходного изображения,  без вектора смещения | |
| BatchNormalization | | Слой, нормализующий свои входы | |
| LeakyReLu | | Активационный слой | |
| Conv2DTranspose | | Транспонированный свёрточный слой,  количество карт признаков = 64,  высота и ширина ядра свёртки = 5,  шаги свёртки по высоте и ширине = 2,  на выходе получается изображение с разрешением исходного изображения,  без вектора смещения | |
| BatchNormalization | | Слой, нормализующий свои входы | |
| LeakyReLu | | Активационный слой | |
| Conv2DTranspose | | Транспонированный свёрточный слой,  количество карт признаков = 1,  высота и ширина ядра свёртки = 5,  шаги свёртки по высоте и ширине = 2,  на выходе получается изображение с разрешением исходного изображения,  без вектора смещения, функция активации - tanh | |
| Дискриминатор | | |
| Conv2D | Свёрточный слой, количество карт признаков = 64,  высота и ширина ядра свёртки = 5,  шаги свёртки по высоте и ширине = 2,  на выходе получается изображение с разрешением исходного изображения,  без вектора смещения, размерность входного пространства – (28, 28, 1) | |
| LeakyReLu | Активационный слой | |
| Dropout | Отключает нейроны с вероятностью = 0.3 | |
| Conv2D | Свёрточный слой, количество карт признаков = 128,  высота и ширина ядра свёртки = 5,  шаги свёртки по высоте и ширине = 2,  на выходе получается изображение с разрешением исходного изображения,  без вектора смещения | |
| LeakyReLu | Активационный слой | |
| Dropout | Отключает нейроны с вероятностью = 0.3 | |
| Flatten | Слой изменят форму тензора: (None, 7, 7, 128) -> (None, 6272) | |
| Dense | Полносвязный слой, количество нейронов = 1 | |

Генератор и Дискриминатор определяются с помощью API Keras Sequential.  
Генератор использует tf.keras.layers.Conv2DTranspose слои повышающей дискретизации для получения изображения из seed (случайного шума).

Следующая анимация показывает серию изображений, похожих на данные из MNIST, вырабатываемых Генератором, который был обучен в течение 50 эпох. Изображения начинаются как случайный шум (произвольный вектор) и со временем все больше напоминают рукописные цифры.



Сеть обучалась в течении 50 эпох. Размер батча: 256. Среднее время обработки эпохи: 333,5 секунды. В качестве оптимизатора был выбран Adam.

Результат работы сети:

|  |  |
| --- | --- |
| Случайные входные вектора для Генератора: | Полученные изображения, спустя 50 эпох обучения: |
|  |  |