

РАЗРАБОТКА ГЕНЕРАТИВНО-СОСТЯЗАТЕЛЬНОЙ НЕЙРОСЕТИ ДЛЯ ГЕНЕРАЦИИ КТ-ИЗОБРАЖЕНИЙ ГРУДНОЙ КЛЕТКИ

Кузнецов И.Е.

Томский политехнический университет, студент гр. 8К93, e-mail: iek7@tpu.ru

Введение

В настоящее время искусственные нейронные сети широко используются при решении самых разнообразных задач особенно там, где обычные алгоритмические решения оказываются неэффективными или вовсе невозможными [1]. Одним из таких бурно развивающихся направлений применения нейросетей является генерация изображений с помощью архитектуры GAN.

Генеративно-состязательная нейросеть (Generative adversarial network, GAN) — архитектура нейросети, состоящая из генератора и критика, настроенных на работу друг против друга [2].

Основной принцип GAN заключается она в том, чтобы использовать хорошо обученный классификатор, дабы различать сгенерированное изображение и реальное изображение. При наличии такого классификатора, можно создать и обучить сеть-генератор, пока она не сможет производить изображения, которые могут полностью обмануть классификатор [2].

Целью моей исследовательской работы являлось создание генеративно-состязательной нейросети для генераций изображений грудной клетки подобно тем, что получаются с помощью компьютерного томографа. Полученные с помощью генератора изображения можно использовать для наполнения датасетов, используемых уже при обучении классифицирующих или сегментирующих нейросетей.

Описание алгоритма

Самой простой версией GAN является архитектура, состоящая из полносвязных слоев. Подобная нейросеть едва ли способна генерировать простейшие изображения, поэтому для достижения поставленной цели, были изучены и реализованы более сложные архитектуры генеративно-состязательных нейросетей, которые будут описаны ниже.

Deep convolutional generative adversarial network или DCGAN – это генеративно-состязательная нейросеть, в которой вместо полносвязных слоев используются сверточные слои. Работа сверточной нейронной сети обычно интерпретируется как переход от конкретных особенностей изображения к более абстрактным деталям. При этом сеть сама вырабатывает необходимую иерархию абстрактных признаков, фильтруя маловажные детали и выделяя существенное.

Принцип работы генератора DCGAN состоит в следующем: на вход нейросети подается n -мерный вектор, который затем проходит через несколько сверточных слоев, постепенно увеличивающих размер изображения, пока он не достигает нужного разрешения.

Дискриминатор DCGAN обладает обратной структурой, в которой сверточные слои понижают разрешение подаваемого на вход реального или сгенерированного изображения, пока на выходе после прохождения сигмоидной функции активации не получается вектор.

При обучении DCGAN в качестве функции потерь используется бинарная кросс-энтропия. Пусть $D(x)$ будет выходом дискриминатора, который представляет собой вероятность того, что x является реальным изображением, а $G(z)$ будет выходом нашего генератора. Дискриминатор аналогичен бинарному классификатору, поэтому цель дискриминатора — максимизировать функцию потерь. С другой стороны, цель генератора минимизировать шансы дискриминатора сделать правильное определение, поэтому он старается минимизировать функцию потерь. Следовательно, окончательная функция потерь будет минимаксной игрой между двумя классификаторами (1):

$$\min_G \max_D L = \log(D(x)) + \log(1 - D(G(z))) \quad (1)$$

Изображения генерируемые DCGAN после обучения хоть и повторяли силуэт грудной клетки, но при этом не отображали мелких детали и были очень расплывчатыми. Таким образом, чтобы улучшить качество создаваемых изображений была реализована продвинутая архитектура генеративно-состязательной нейросети под названием ProGAN.

Progressive generative adversarial network или ProGAN – это сверточная генеративно-состязательная сеть, получившая свое название, благодаря своему основному принципу – прогрессивному или постепенному добавлению новых сверточных блоков и обучению генератора и критика (дискриминатора) на изображениях, увеличивающегося размера [3].

Для генератора задача сразу сгенерировать изображение большого разрешения может быть достаточно сложной, поэтому было предложено обучать его постепенно. Так для начала генератор обучается создавать изображения малого разрешения, а критик распознавать их. Затем, когда генератор достаточно натренирован, мы учим его генерировать изображения большего размера и так далее вплоть до нужного разрешения.

Одно из преимуществ данного подхода это сокращенное время обучения. При прогрессивно растущих GAN большинство итераций выполняется с меньшим разрешением, и сопоставимое качество результата часто достигается в несколько раз быстрее.

Еще одним большим отличием ProGAN от DCGAN является метод обучения моделей и вид функции потерь. Пусть P_g и P_r вероятностные распределения генерируемых и реальных изображений. Для генерации реалистичных изображений нужно чтобы они были как можно более близки.

Новую функцию потерь можно представить следующим образом (2):

$$E_x(x \sim P_r) [f(x)] - E_x(x \sim P_g) [f(x)] \quad (2)$$

В левой части выражения x – это тензор, получаемый на выходе дискриминатора, который в ProGAN в официальной литературе называют критиком. В правой же части выражения x – тензор значений, который создает генератор. E означает нахождение среднего значения данных выходных сигналов.

В ходе обучения нейросети критик стремится как можно больше разделить эти значения, чтобы суметь их различать, поэтому он стремится к максимизации данного выражения. В то же время задачей генератора наоборот является свести эти значения к одному модулю, дабы обмануть критика, в связи с чем он стремится к минимизации данного выражения.

В этом принципе также заключается еще одно преимущество ProGAN перед предыдущей структурой, поскольку значение функции потерь позволяет понять успешность тренировки нейросетей. Чем эти значения в конце тренировки ближе к нулю, тем лучше генератор справляется со своей задачей.

Нейросеть в подобной конфигурации обучалась в течении 9 часов, по 30 эпох на каждое разрешение. Результат обучения можно оценить по рисункам ниже.



Рис. 1. График значений функции потерь нейросети ProGAN

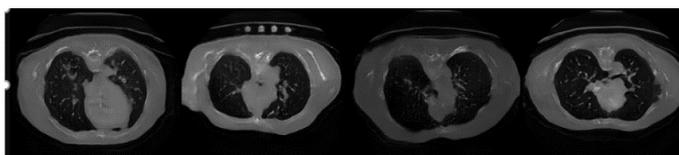


Рис. 2 Изображения, создаваемые генератором ProGAN

Как видно из рисунка 2 изменение архитектуры нейросети привело к значительному улучшению качества генерируемых изображений. Однако на получаемых таким образом изображениях все еще трудно отличать различные ткани, поскольку изображения получаются однотонными. На это есть несколько причин. Во-первых, загрузчик данных библиотеки Pytorch, с помощью которой были реализованы как сами модели генератора и критика, так построено процесс обучения, не может напрямую работать с особым форматом изображений, используемым в медицине, *dicom* (*dcm*). Поэтому для обучения описанных ранее нейросетей приходилось предварительно с помощью сторонней библиотеки конвертировать изображения из обучающей в выборке в формат PNG. Такое преобразование могло приводить к потере качества. Во-вторых, в оригинальных трансформациях библиотеки Pytorch отсутствует возможность точной настройки контраста изображений, что как раз и приводит к неразличимости мелких деталей у изображений из обучающей выборке.

Для решения описанных выше проблем была использована библиотека MONAI, спроектированная специально для решений задач машинного обучения при работе с медицинскими изображениями. Данная библиотека была использована на этапе загрузки и предварительной трансформации изображений, при этом структуры генератора и критика остались прежними. Лишь в алгоритм обучения были внесены небольшие изменения.

Поскольку строение моделей генератора и критика остались неизменными, то сам процесс обучения нейросети так же занял 9 часов. Результаты обучения представлены на рисунках 3 и 4.

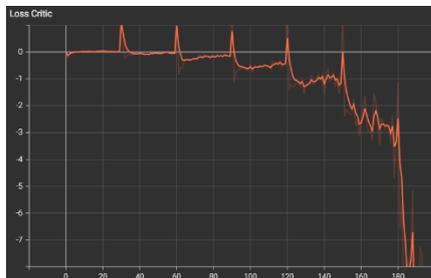


Рис. 3. График значений функции потерь нейросети ProGAN_MONAI

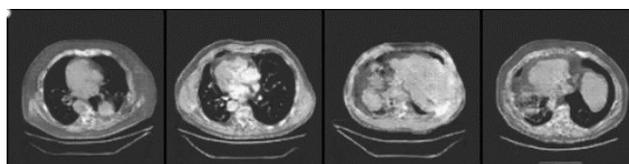


Рис. 4. Изображения, создаваемые генератором ProGAN_MONAI

При сравнении графиков на рисунках 1 и 3 явно видно, что значения функции потерь нейросети ProGAN без использования MONAI больше по модулю, а значит генератор второй версии нейросети создает более реалистичные изображения. Также, по субъективному мнению, изображения, создаваемые ProGAN с MONAI (рис. 4) получаются более детализированными по сравнению с прошлой версией (рис 2.).

Заключение

В ходе исследовательской работы было изучено и реализовано несколько архитектур генеративно-сопоставительных нейросетей, каждая со своими особенностями. Создаваемые после обучения нейросетей изображения, а также графики позволяют наглядно проследить эволюция генератора, с каждой версией создающего все более детальные и качественные изображения. Так изображения, получаемые на выходе генератора ProGAN, обученном с помощью предобработанных посредством библиотеки MONAI, являются достаточно качественными, чтобы использовать их при обучении других нейросетей.

Список использованных источников

1. Цаунит, А. Н. Перспективы развития и применения нейронных сетей / А. Н. Цаунит. — Текст : непосредственный // Молодой ученый. — 2021. — № 23 (365). — С. 114-117.
2. Создаем GAN с помощью PyTorch. [Электронный ресурс]. — URL: <https://habr.com/ru/company/otus/blog/569858/> (дата обращения 10.12.2022).
3. Unsupervised Representation Learning with Deep Convolutional Generative Adversarial Networks. [Электронный ресурс]. — URL: <https://arxiv.org/abs/1511.06434> (дата обращения 15.01.2023).