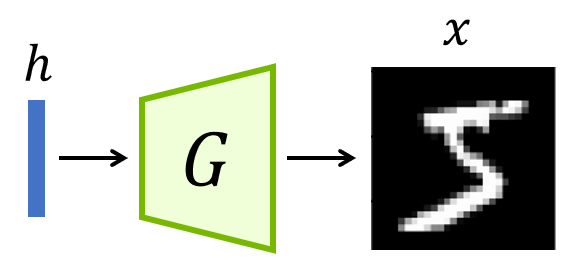
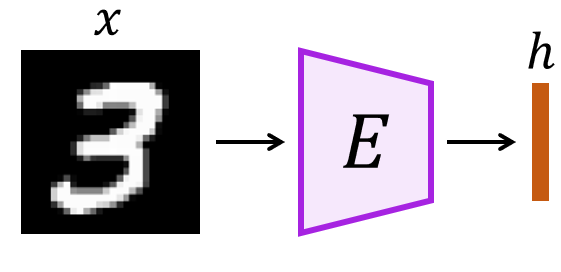
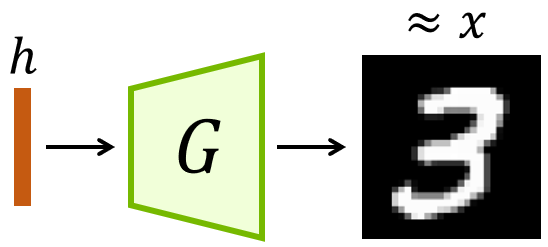
**[Часть 1]** Обучите свёрточную генеративную модель (с использованием **DCGAN**) на изображениях цифр из датасета **MNIST**. Генератор ***G*** должен синтезировать случайные изображения цифр ***x*** по некоторому входному латентному вектору ***h***, который семплируется из нормального распределения.



**[Часть 2]** Создайте процедуру ***E*** получения латентного вектора ***h*** по входному изображению с цифрой ***x*** (при условии полученного ранее фиксированного генератора ***G***)



Вектор ***h*** должен получиться таким, чтобы при подаче на вход в генератор ***G***, он выдавал изображение изначально заданной цифры ***x***.



Продемонстрируйте, что это условие выполняется.

Реализуйте **два** варианта решения **Части 2**:

**[2.1]** Через создание универсальной Feed-Forward модели ***h*** *=* ***E****(****x****)*

**[2.2]** Через оптимизационный поиск ***h*** по данному ***x***.

***Пояснение***: Необходимо создать функцию, которая на вход принимает ***x*** (изображение цифры из датасета MNIST) и возвращает вектор ***h***. Полученный вектор ***h*** должен удовлетворять следующему условию: если мы подаем ***h*** в полученный ранее (фиксированный) генератор ***G***, то мы должны получить изображение, близкое к исходному ***x***. Внутри функции, которую необходимо создать, вектор ***h*** должен получаться с помощью некоторого оптимизационного (итерационного) процесса (процесс поиска ***h*** запускается для каждого входного ***x***), а не с помощью Feed-Forward модели, как в пункте **[2.1]**

Оформите и пришлите решение в виде **Ipython Notebook** (не размещайте его в публичный доступ). Подойдите ответственно к оформлению и подробно задокументируйте ваш код (что где происходит). Используйте **PyTorch**.