

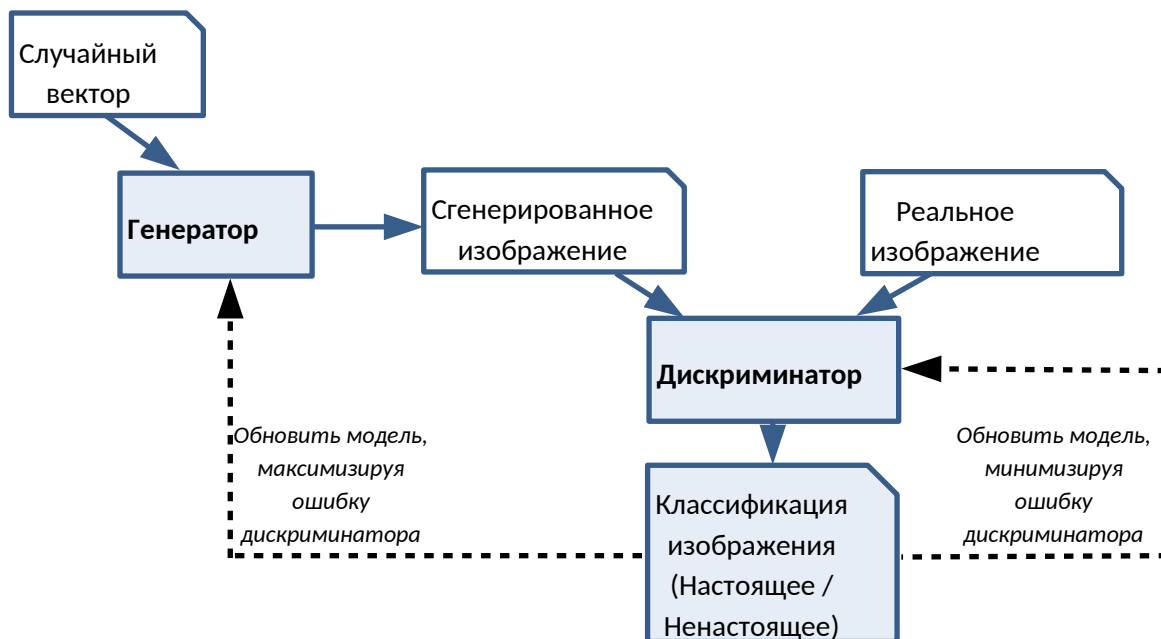
Задача: передать в GAN-сеть информацию о типе изображения. Эта информация должна передаваться как в генератор, так и в дискриминатор. Это позволит:

- генератору - генерировать изображения указанного типа;
- дискриминатору - учитывать тип изображения при распознавании.

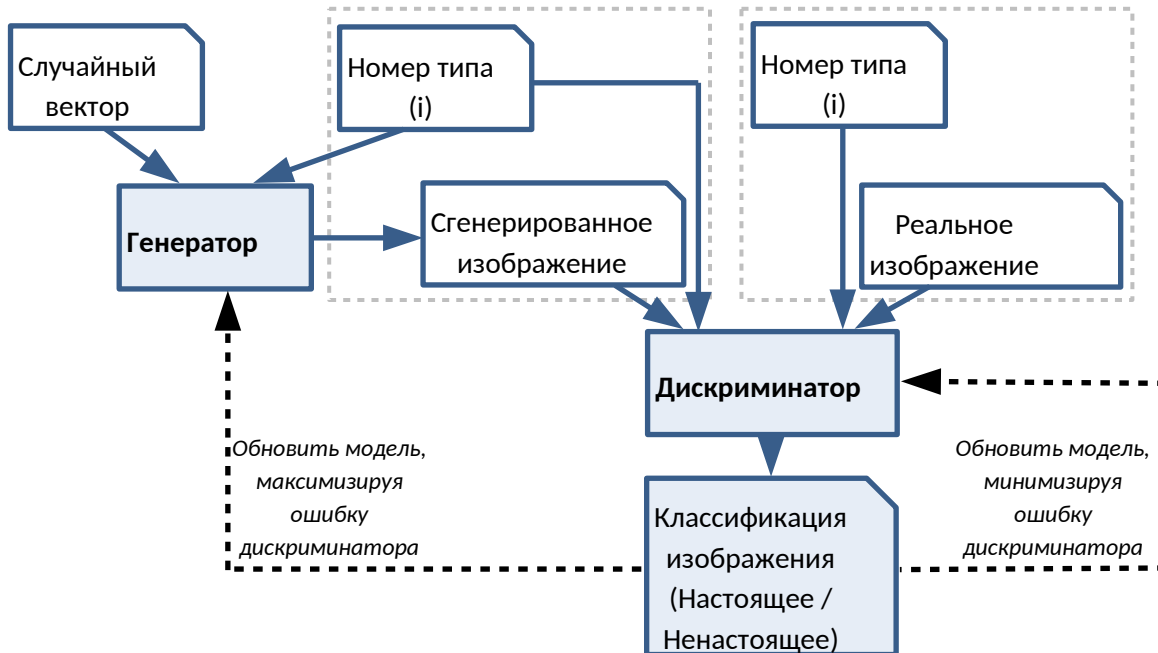
Количество типов изображений ( $N$ ) - это целое положительное число. В обучающей выборке должны присутствовать изображения каждого типа.

Информация о типе изображения - это номер типа ( $i$ ) - положительно целое число в диапазоне от 1 до  $N$ .

Общая схема обычной GAN-сети



Общая схема Conditional GAN-сети



Возникает вопрос - как объединить информацию? Для генератора нужно объединять случайный вектор и одно число  $n$ , для дискриминатора - изображение и одно число  $n$ .

Лучшие практики рекомендуют использовать слой Embedding. Слой Embedding ставит в соответствие числам (номерам типов) плотные вектора размерностью  $(1 \times V)$ .  $V$  - это параметр модели, целое положительное число. Компоненты вектора содержат веса, которые тренируются в процессе обучения.

Таким образом, Embedding хранит в себе матрицу размерностью  $(N \times V)$ , где  $N$  - общее количество типов,  $V$  - размерность вектора

Номера типов \ номера компонент вектора	1	2	3	...	$V-1$	$V$
вектор номер 1						
вектор номер $i$	0.2	0.6	-0.1	...	0.3	0.1
вектор номер N						

Когда мы передаем в слой Embedding конкретный номер класса ( $i$ ), делается срез и оставляется только тот вектор, который соответствует запрошенному классу. Размерность этого вектора  $1 \times V$ .

Теперь нам нужно объединить этот вектор с изображением.

Рассмотрим, как происходит объединение отдельно для дискриминатора и для генератора.

### 1. Дискриминатор

Для дискриминатора нужно объединить одно число с полным изображением размерностью  $M \times M$ , где  $M$  - количество пикселей как по горизонтали, так и по вертикали. В нашем случае  $M=32$ .

1.1. Идея объединения: Число преобразуем в вектор размерностью  $1 \times V$ , который затем преобразуется в матрицу размерностью  $M \times M$ . После чего исходное изображение и полученная матрица объединяются в тензор размерностью  $M \times M \times 2$ .

1.2. Реализация: У дискриминатора два входа - изображение и номер типа.

1.2.1) Изображение передаем как есть, в виде матрицы  $32 \times 32$ .

1.2.2) Номер типа (одно число) передаем на слой Embedding (его компоненты обозначим буквой  $E$ ). За слоем Embedding следует полносвязный слой (Dense, его компоненты обозначим буквой  $D$ ), который в рассматриваемом случае должен содержать  $32 \times 32 = 1024$  нейрона. На выходе полносвязного слоя получаем вектор размерностью 1 на 1024. Затем выход полносвязного слоя преобразуется в матрицу размерностью  $(32 \times 32)$ .



1.3) Объединение. Исходное изображение (компоненты обозначим буквой  $I$ , в индексе - номер пикселя) и преобразованный выход слоя Dense (из п.1.2.2) объединяются методом Concatenate в тензор размерностью  $32 \times 32 \times 2$  (т.е. в тензоре содержатся две матрицы размерностью  $32 \times 32$ ):

$I_1$	$I_2$	$I_3$	...	$I_{31}$	$I_{32}$
$I_{33}$	$I_{34}$	$I_{35}$	...	$I_{63}$	$I_{64}$
...	...	...	...	...	...
$I_{961}$	$I_{962}$	$I_{963}$	...	$I_{991}$	$I_{992}$
$I_{993}$	$I_{994}$	$I_{995}$	...	$I_{1023}$	$I_{1024}$

$D_1$	$D_2$	$D_3$	...	$D_{31}$	$D_{32}$
$D_{33}$	$D_{34}$	$D_{35}$	...	$D_{63}$	$D_{64}$
...	...	...	...	...	...
$D_{961}$	$D_{962}$	$D_{963}$	...	$D_{991}$	$D_{992}$
$D_{993}$	$D_{994}$	$D_{995}$	...	$D_{1023}$	$D_{1024}$

Полученный объединенный тензор передается на конволюционные слои и обрабатываются дальше как в обычной GAN.

## 2. Генератор

Для генератора нужно объединить одно число с шумом размерностью  $1 \times Z$ , где  $Z$  - это параметр сети.

2.1. Идея объединения:

2.1.1) Шум размерностью  $1 \times Z$  преобразуется в несколько (обозначим это число  $F$ ) карт признаков, каждая из карт признаков имеет размерность  $V \times V$ , где  $V$  - параметр сети. При выборе значения  $V$  нужно учитывать, что из этих карт признаков методом обратной конволюции, увеличивая количество пикселей в 2 раза, в конечном итоге получают изображение требуемого размера. В нашем случае (для генерации изображения размером  $32 \times 32$ )  $V=8$ .

Итого, входной шум преобразуем в тензор размерностью  $V \times V \times F$ .

2.1.2) Число преобразуем в вектор размерностью  $1 \times V$ , который затем преобразуется в матрицу размерностью  $V \times V$ .

2.1.3) Объединение. Тензор размерностью  $V \times V \times F$  и матрица размерностью  $V \times V$  объединяются в тензор размерностью  $V \times V \times (F+1)$ .

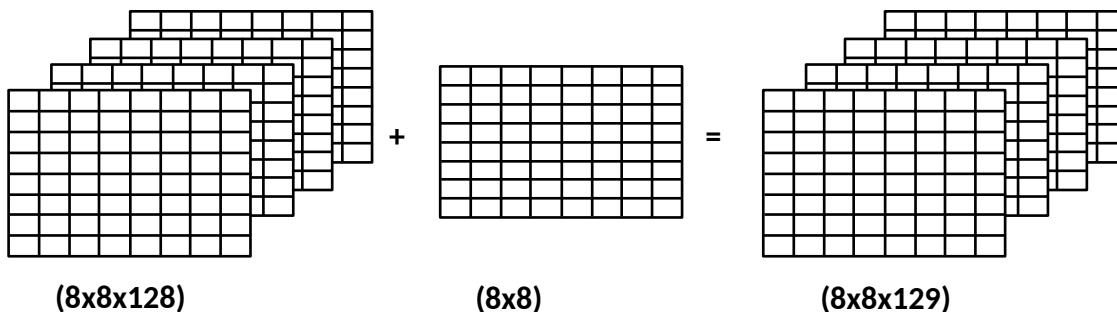
2.2. Реализация: У генератора два входа - шум и номер типа. Рассмотрим наш пример, когда  $V=8$ ,  $F=128$ .

2.2.1) Шум (вектор размерностью  $1 \times Z$ ) подаем на вход полносвязного слоя (Dense), содержащего  $V \times V \times F$ :  $8 \times 8 \times 128 = 8192$  нейрона. На выходе полносвязного слоя получаем вектор размерностью 1 на 8192. Затем выход полносвязного слоя преобразуется (так же, как и для дискриминатора в п.1.2.2 - простой перестановкой компонент) в тензор размерностью  $8 \times 8 \times 128$ .

2.2.2) Номер типа (одно число) передаем на слой Embedding. За слоем Embedding следует полносвязный слой (Dense), который в рассматриваемом случае должен содержать  $8 \times 8 = 64$  нейрона. - Обращаю внимание: это не тот же слой Dense, который мы использовали в п.2.2.1.

Итого, на выходе полносвязного слоя получаем вектор размерностью 1 на 64. Затем выход полносвязного слоя преобразуется в матрицу размерностью  $(8 \times 8)$ .

2.3) Объединение. Карта признаков размерностью  $8 \times 8 \times 128$  из п.2.2.1 и преобразованный выход слоя Dense из п.2.2.2 (размерностью  $8 \times 8$ ) объединяются методом Concatenate в тензор размерностью  $8 \times 8 \times 129$ .



Полученный объединенный тензор передается на слои обратной конволюции и обрабатываются дальше как в обычной GAN.

