

# Solution tool

Chekalina V., Ratnikov F.



*British Hedgehog Preservation Society*

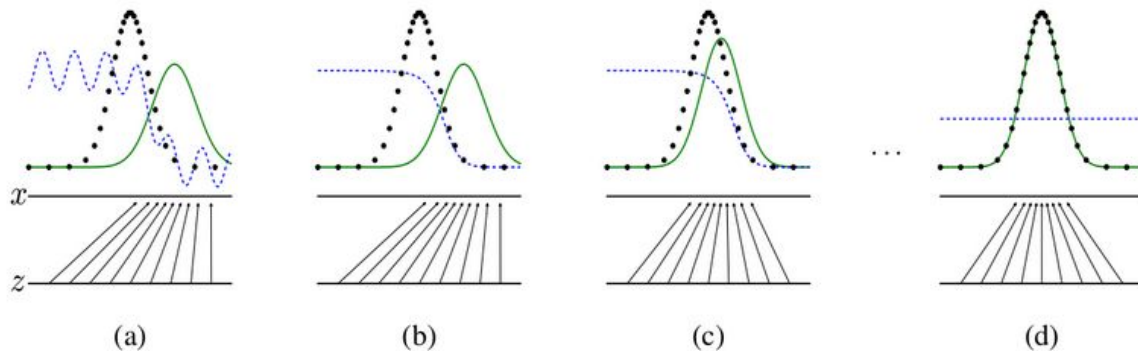
---

# Generative Adversial Nets

1.  $z$  - базовое распределение (*например, Гаусса*),  
 $G(z)$  - объект, полученный из генератора(*например, шум*)  
 $x$  - настоящее распределение (*например, картинка*)  
 $D(obj\_i)$  - вероятность того, что  $i$ -тый объект принадлежит настоящему распределению.
2. Генератор  $G$  создаёт объект, дискриминатор  $D$  оценивает вероятность созданного объекта происходить из настоящего распределения.
3. Задача  $D$  - отделить объекты из генератора от “настоящих”, задача  $G$  - сгенерировать максимально похожий объект.
4.  $Loss D \sim \max (\log D(x) + \log (1 - D(G(z)))$   
~ Уверенности дискриминатора, что объект из настоящего распределения является объектом из настоящего распределения, а объект из генератора не является объектом из настоящего распределения.  
 $Loss G \sim \min \log (1 - D(G(z)))$
5. Генерация заканчивается, когда дискриминатор выдает вероятность 1/2

## How to train it

1. Черная кривая - изначальное распределение
2. Зеленая - распределение генератора  $G(z)$
3. Синее - ответ дискриминатора



(a) Изначальные данные

и ответ генератора отличаются, но дискриминатор их отличает случайно  $\Rightarrow$  нужно  $k$  раз обучить дискриминатор.

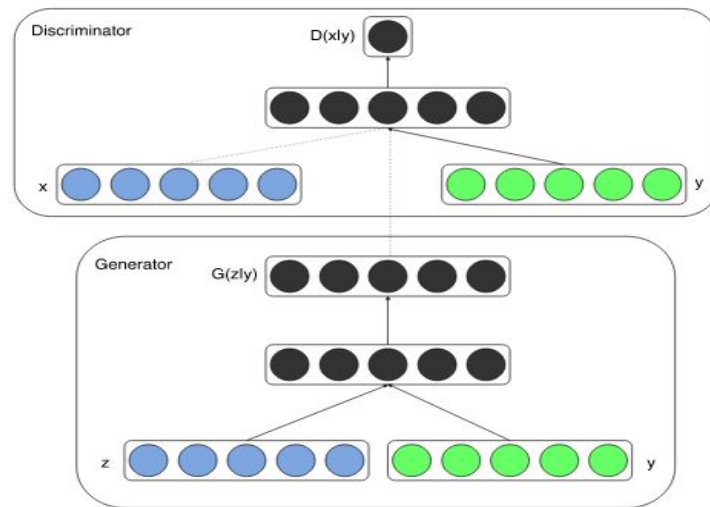
(b) После обучения дискриминатор отличает объекты увереннее, градиент даёт возможность (c) генератору “подвинуть” распределение ближе к изначальному.

(d) Оптимум - дискриминатор не способен отличить одно распределение от другого.

# Conditional GAN

Можно направлять процесс генерации, используя дополнительные фичи - метки классов, описания картинок.

В граф вычислений сети объект обычно попадает после предобработки. Дополнительная информация предобрабатывается вместе с объектом и попадает на вход сети как единое целое.



## Some practice notes

1. Для оптимизации генератора вместо минимизации ( $\log 1-D(G(z))$ ) максимизировать  $\log D(G(z))$   
- градиент функции потерь на начальных шагах не занулиться
2. Не смешивать в одном мини-батче объекты из настоящей выборки и пришедшие из генератора.
3. Для нахождения оптимального количества шагов для обучения каждой части можно попробовать:

*while lossD > A:*

*train D*

*while lossG > B:*

*train G*