Machine Learning

GAN

Contents

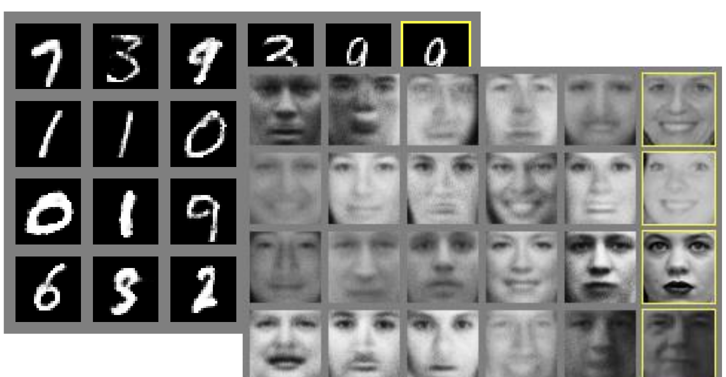
Introduction to GAN

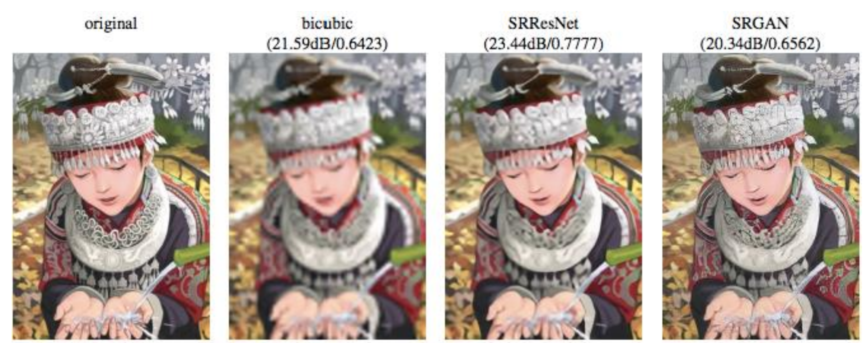
▪ Generative Adversarial Network

- Unsupervised learning
- Generative model (shows *state-of-the-art* performance)
- 데이터가 생성하는 분포 $p(x)$를 이용한 간접적인 학습방법
  - 모든 데이터는 확률분포를 가지고 있는 확률변수

    * 확률변수에 대한 확률분포 $p(x)$를 안다는 것

    → 데이터 전체를 이해할 수 있다





NIPS 2016 Tutorial: Generative Adversarial Network, Ian GoodFellow, 2016

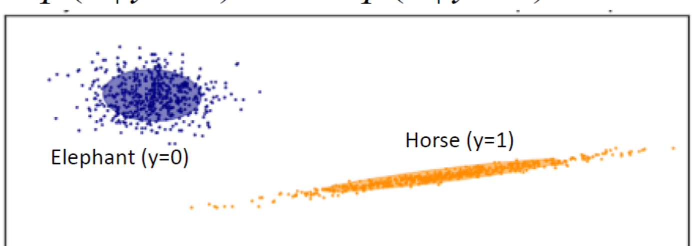
Generative Model

- Unsupervised learning Methods
  - 많은 비지도학습(판별모델)은 Maximum Likelihood를 사용
    - Image **X**가 주어지면, label **Y**를 예측 → $P(\boldsymbol{Y}|\boldsymbol{X})$를 추정
  - Discriminative model의 한계
    - $P(\boldsymbol{X})$를 만들어 낼 수 없음 (i.e. 특정 이미지의 확률)

      * 따라서, $P(\boldsymbol{X})$로부터 샘플을 추출할 수 없음

      → 새로운 이미지를 생성할 수 없다.
  - GAN에서는 likelihood를 직접 사용하지 않음
    - $P(\boldsymbol{X})$를 만들어낼 수 있음 → $P(\boldsymbol{X}|\boldsymbol{Y})$

$p(x \mid y=0)$ $p(x \mid y=1)$



Adversarial Training

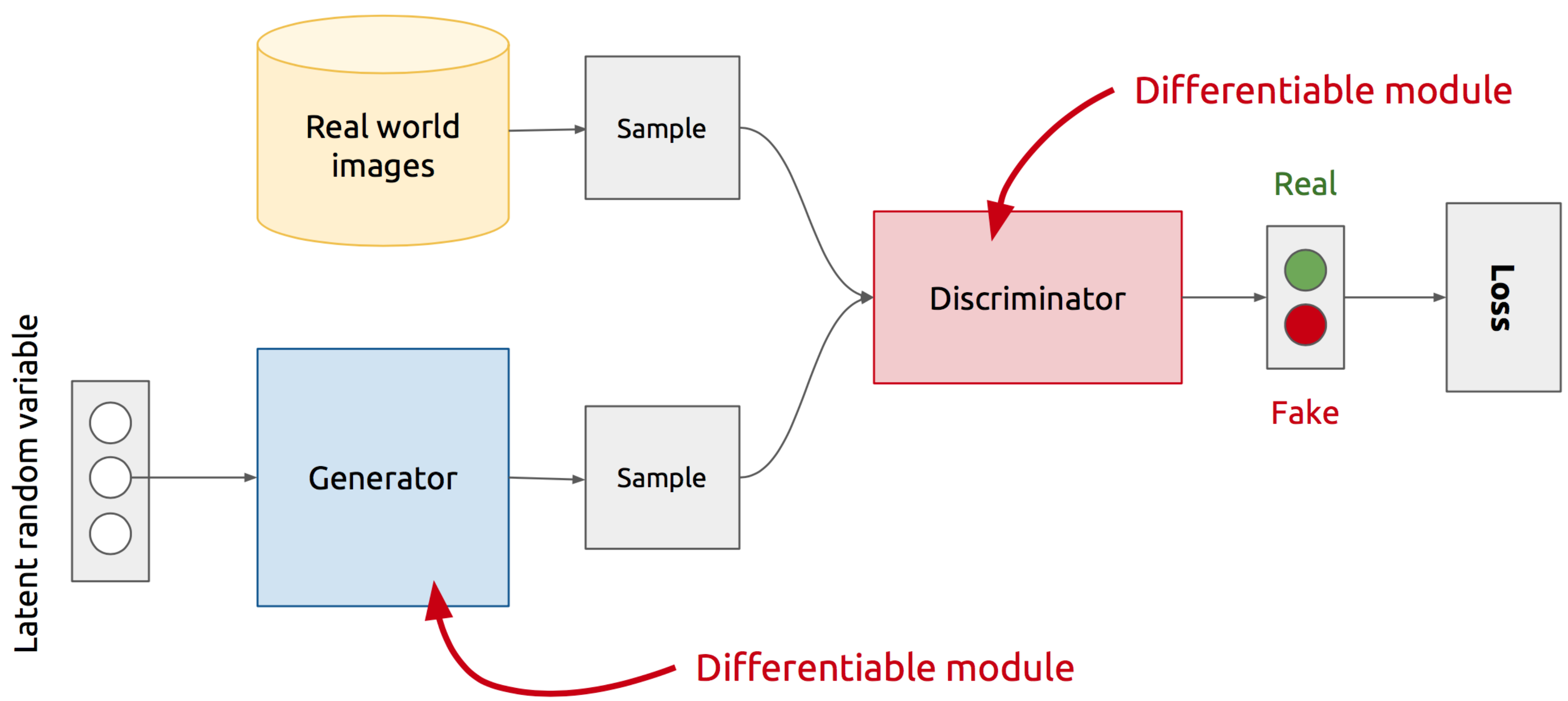
▪ Generator (G)

- 훈련 데이터와 비슷한 샘플을 생성
- z : 랜덤 노이즈 벡터 (Gaussian/Uniform)

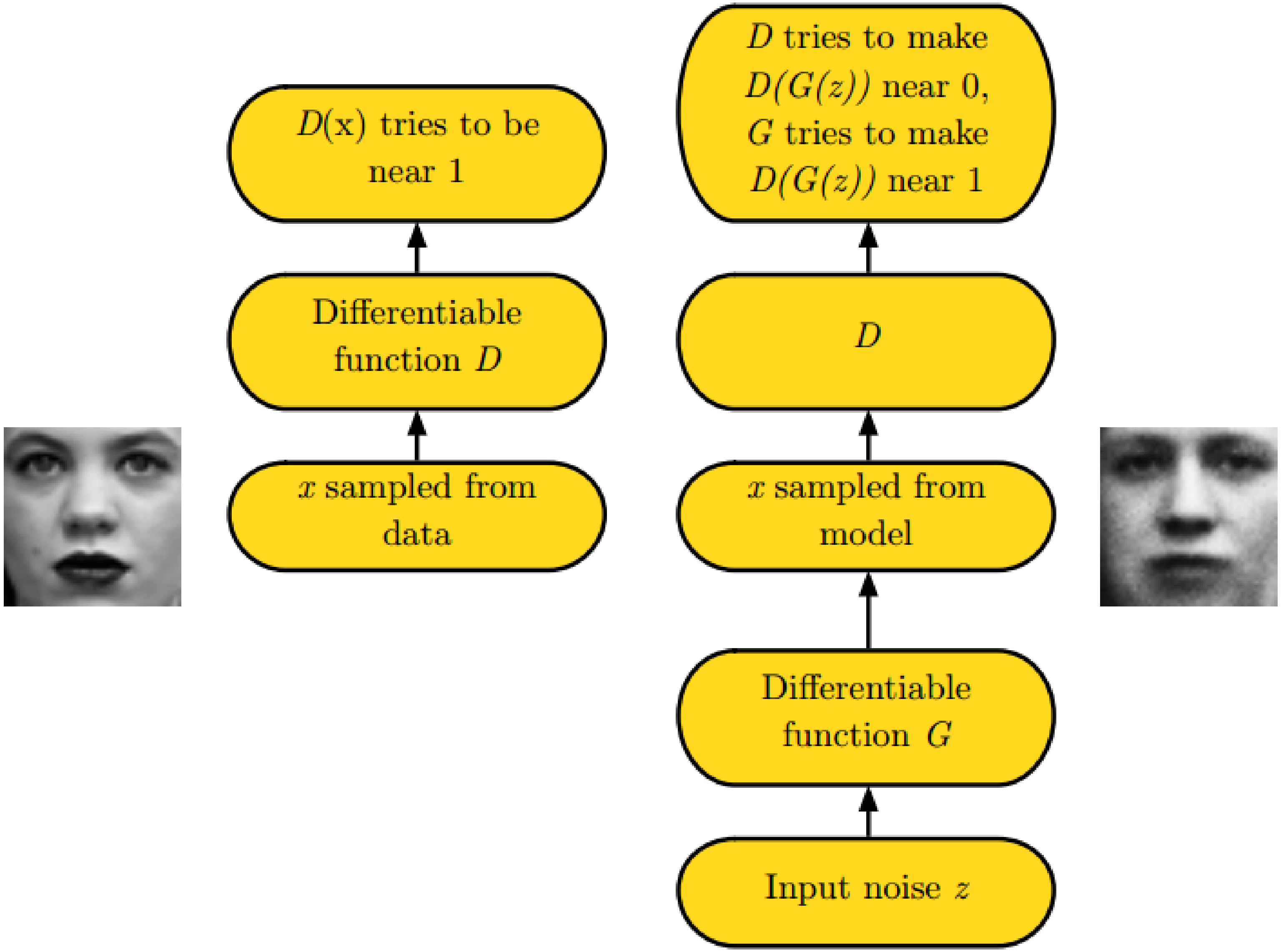
▪ Discriminator (D)

- 가짜 샘플과 훈련 데이터를 구별

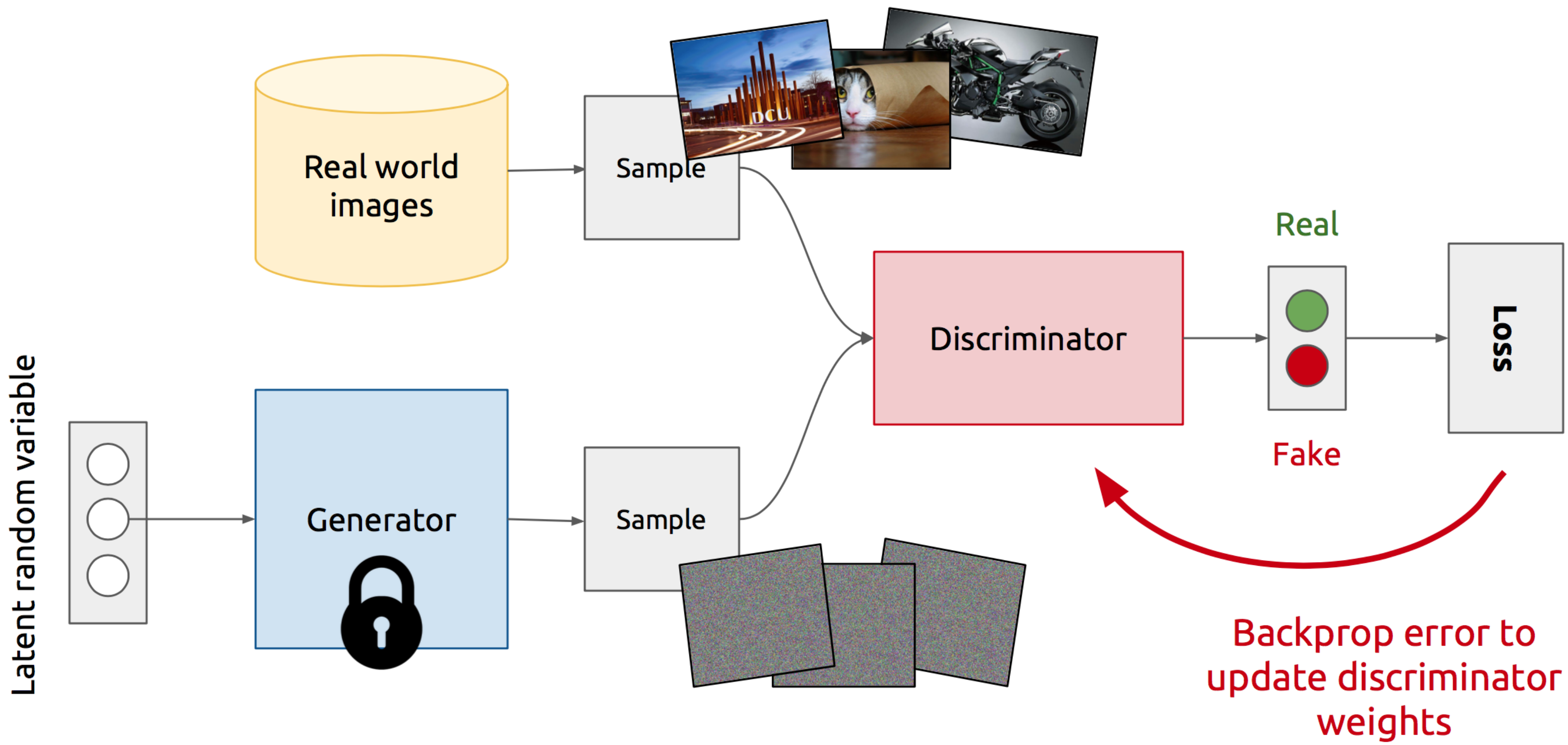
Architecture



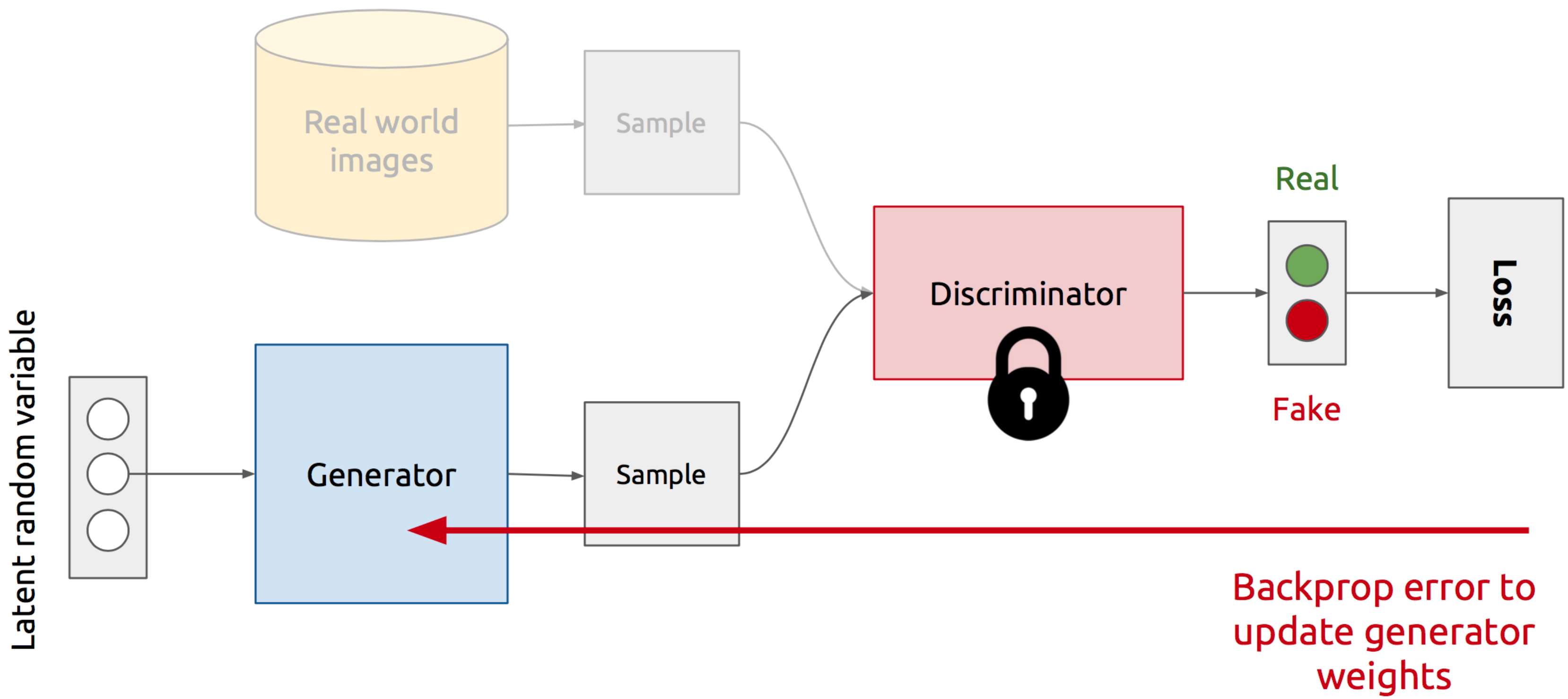
Architecture



Training Discriminator

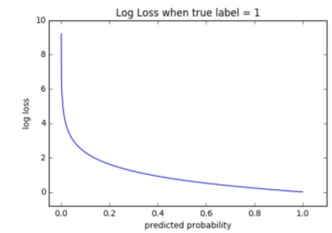


Training Generator



Objective

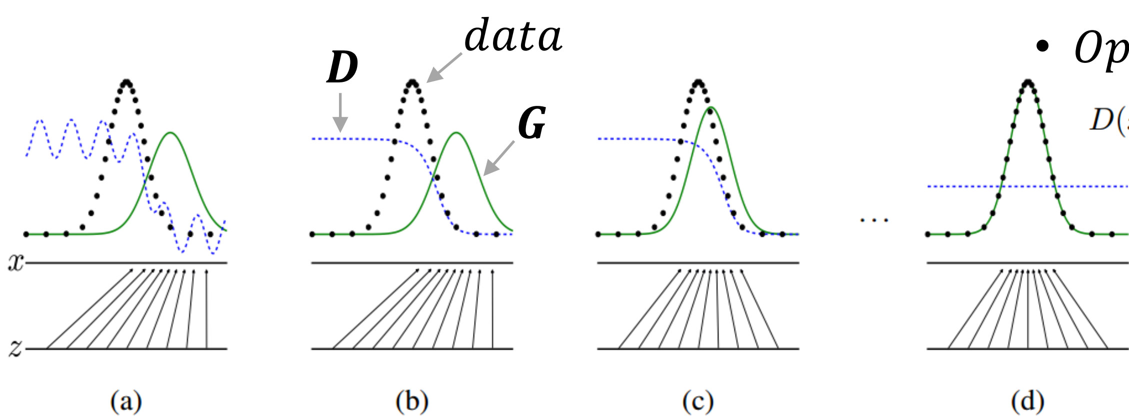
- Error function
  - Binary Cross Entropy (BCE) : $-(y\log(p) + (1-y)\log(1-p))$
    - 이항 분류 (0 또는 1)



- **Objective function**

$$\min_G \max_D V(D,G)$$

$$V(D,G) = \mathbb{E}_{x\sim p_{data}(x)}[logD(x)] + \mathbb{E}_{z\sim p_z(z)}[log(1-D(G(z)))]$$



- *Optimal **D***

$$D(x) = \frac{p_{\text{data}}(x)}{p_{\text{data}}(x) + p_{\text{model}}(x)}$$

Formulation

$$\min_G \max_D V(D, G)$$

$$V(D,G) = \mathbb{E}_{x \sim p_{data}(x)}[logD(x)] + \mathbb{E}_{z \sim p_z(z)}[log(1 - D(G(z)))]$$

- **Minimax game**
  - Discriminator는 보상 V(D,G)를 최대화하려고 함
  - Generator는 Discriminator의 보상을 줄이려고 함 (또는 Discriminator의 실수를 최대화)
- Nash equilibrium이 이루어지는 지점
  - $P_{data}(x) = P_{gen}(x) \ \forall x$
  - $D(x) = \frac{1}{2} \ \forall x$

Algorithm

for number of training iterations **do**

for k steps **do**

- Sample minibatch of m noise samples $\{\boldsymbol{z}^{(1)}, \dots, \boldsymbol{z}^{(m)}\}$ from noise prior $p_g(\boldsymbol{z})$.
- Sample minibatch of m examples $\{\boldsymbol{x}^{(1)}, \dots, \boldsymbol{x}^{(m)}\}$ from data generating distribution $p_{\text{data}}(\boldsymbol{x})$.
- Update the discriminator by ascending its stochastic gradient:

Update ***D***

$$\nabla_{\theta_d} \frac{1}{m} \sum_{i=1}^{m} \left[\log D\left(\boldsymbol{x}^{(i)}\right) + \log\left(1 - D\left(G\left(\boldsymbol{z}^{(i)}\right)\right)\right) \right].$$

end for

- Sample minibatch of m noise samples $\{\boldsymbol{z}^{(1)}, \dots, \boldsymbol{z}^{(m)}\}$ from noise prior $p_g(\boldsymbol{z})$.
- Update the generator by descending its stochastic gradient:

Update ***G***

$$\nabla_{\theta_g} \frac{1}{m} \sum_{i=1}^{m} \log\left(1 - D\left(G\left(\boldsymbol{z}^{(i)}\right)\right)\right).$$

end for

The gradient-based updates can use any standard gradient-based learning rule. We used momentum in our experiments.

Gradient Vanishing

$$\min_G \max_D V(D, G)$$

$$V(D, G) = \mathbb{E}_{x \sim p_{data}\,(x)}[logD(x)] + \boxed{\mathbb{E}_{z \sim p_x(z)}[log(1 - D(G(z)))]}$$

$$\nabla_{\theta_G} V(D, G) = \nabla_{\theta_G} \mathbb{E}_{z \sim q(z)} \left[\log\left(1 - D\left(G(z)\right)\right)\right]$$

- $\nabla_a \log\left(1 - \sigma(a)\right) = \frac{-\nabla_a \sigma(a)}{1 - \sigma(a)} = \frac{-\sigma(a)\left(1 - \sigma(a)\right)}{1 - \sigma(a)} = -\sigma(a) = -D\left(G(z)\right)$
- Gradient goes to 0 if D is confident, i.e. $D\left(G(z)\right) \to 0$
- Minimize $\boxed{-\mathbb{E}_{z \sim q(z)}\left[\log D\left(G(z)\right)\right]}$ for **Generator** instead (keep Discriminator as it is)

Global Optimality

- Is there a proof that minimax loss leads to $p_G^{fake}(x) = p^{real}(x)$? **YES**.
  1. For a fixed G, Note that $D_G^* = \arg\max_D V(D, G) = \frac{p^{real}}{p^{real} + p_G^{fake}}$.
  2. Then, $\min_G V(D_G^*, G)$ is achieved if and only if $p_G^{fake}(x) = p^{real}(x)$.

Global Optimality

- Proof of 1.

$$V(D,G) = \mathbb{E}_{x\sim p^{real}(x)}[\log D(x)] + \mathbb{E}_{x\sim p_G^{fake}(x)}[\log(1-D(x))]$$
$$= \int [p^{real}(x)\log D(x) + p_G^{fake}(x)\log(1-D(x))]dx$$

For a fixed G, the extrema is at:

$$\frac{\partial V(D,G)}{\partial D} = \int \frac{p^{real}(x)}{D(x)} - \frac{p_G^{fake}(x)}{1-D(x)} dx = 0$$

For this to be satisfied for any p^{real} and p_G^{fake}, the term inside the integral has to be zero.

$$\therefore D_G^*(x) = \frac{p^{real}(x)}{p^{real}(x) + p^{fake}(x)}$$

Global Optimality

- Proof of 2.

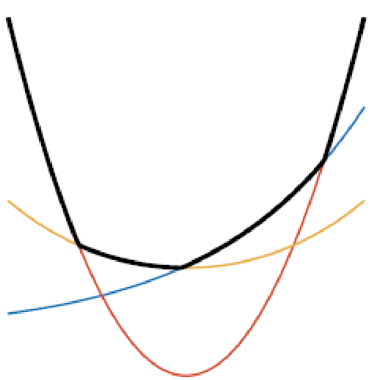
Substituting $D_G^*(x)$ to the loss,

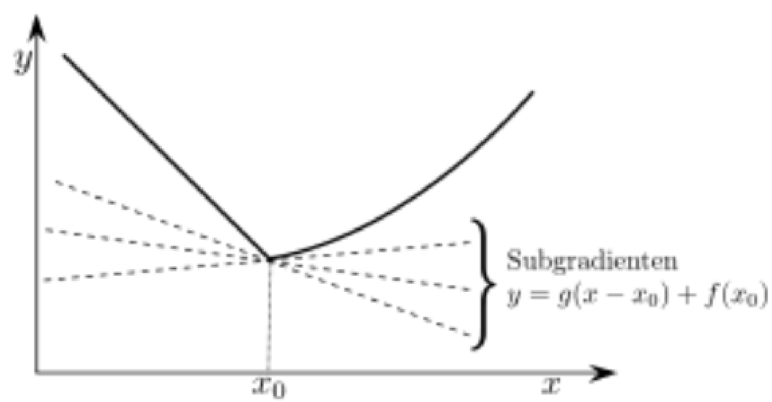
$$V(D_G^*, G) = \mathbb{E}_{x\sim p^{real}(x)}[\log(\frac{p^{real}(x)}{(p^{real}(x)+p_G^{fake}(x))/2}) + \log\frac{1}{2}]$$
$$+ \mathbb{E}_{x\sim p_G^{fake}(x)}[\log(\frac{p_G^{fake}(x)}{(p^{real}(x)+p_G^{fake}(x))/2}) + \log\frac{1}{2}]$$
$$= -\log 4 + KL(p^{real}||\frac{p^{real}(x)+p_G^{fake}(x)}{2})$$
$$+ KL(p_G^{fake}||\frac{p^{real}(x)+p_G^{fake}(x)}{2})$$
$$= -\log 4 + 2JS(p^{real}(x)||p_G^{fake}(x))$$
$$\geq -\log 4$$

becomes equality when $p^{real}(x) = p_G^{fake}(x)$.

Convergence of Algorithm

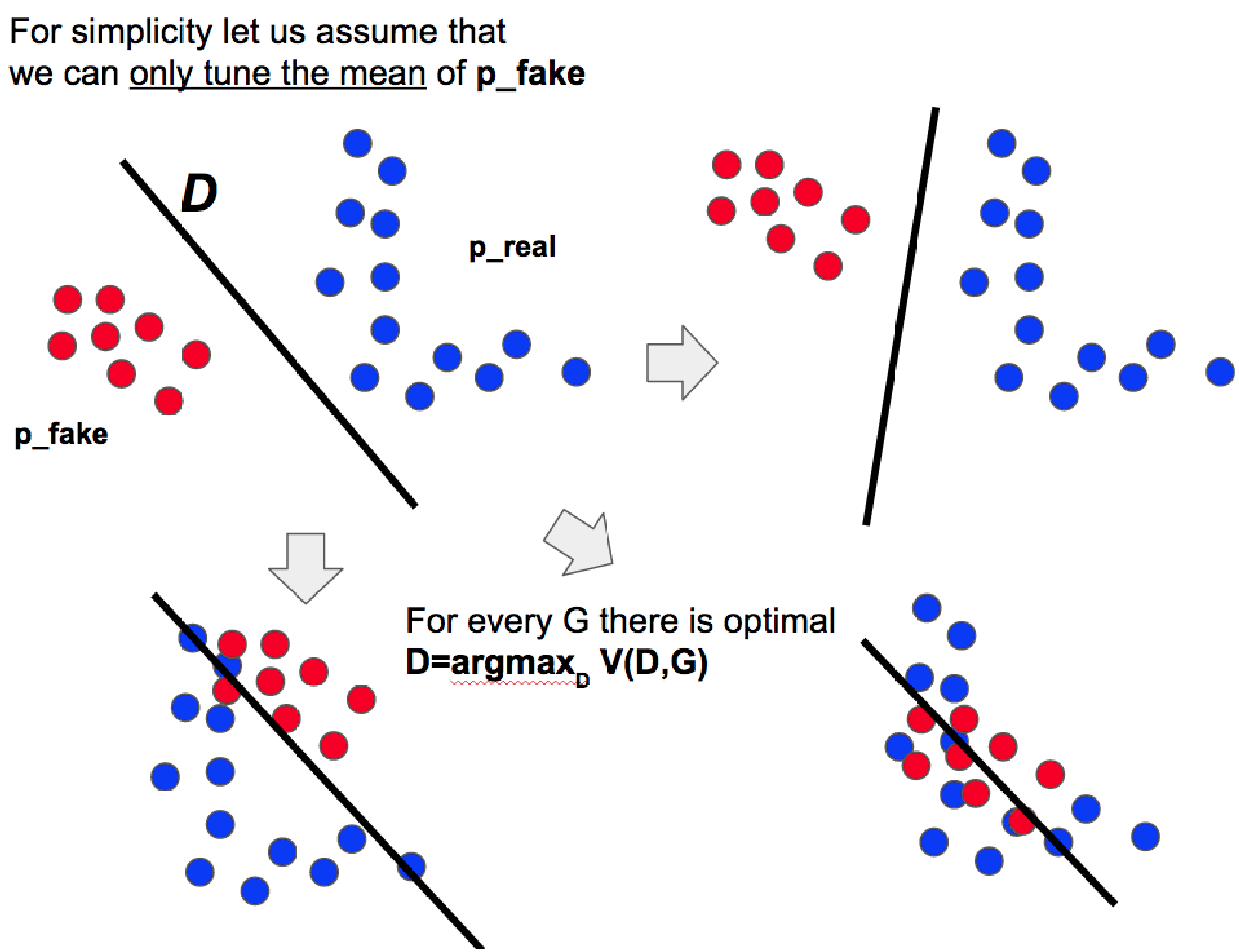
- How can we assure that this algorithm is convergent?
  1. For a fixed D, let us denote the loss as a function of p_G^{fake}, $U(p_G^{fake}) = V(G, D)$. Then, $U(p_G^{fake})$ is convex in p_G^{fake} (property of KL-div.).
  2. Our interest is to minimize not $V(G, D)$ but $\max_D V(G, D)$. However, note that if $V(G, D)$ is convex for all D, $\max_D V(G, D)$ is convex as well.
  3. Therefore, at $G = \bar{G}$, the gradient of $V(G, D^*_{\bar{G}})$ is always a subgradient of the original loss $\max_D V(G, D)$. Thus, it is okay to follow the gradient of $V(G, D^*_{\bar{G}})$, as long as the step size is sufficiently small.





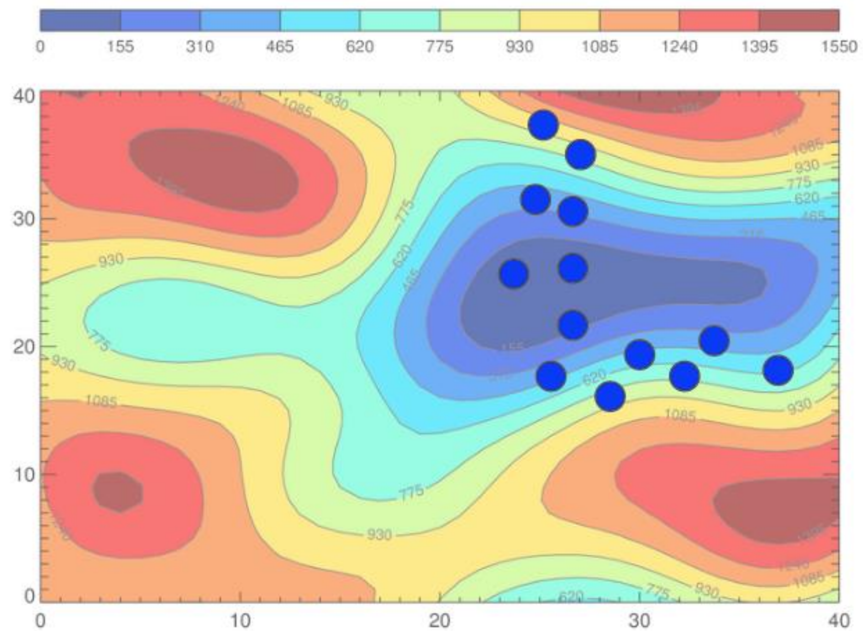
Convergence of Algorithm

- Graphical Explanation



Convergence of Algorithm

- Graphical Explanation



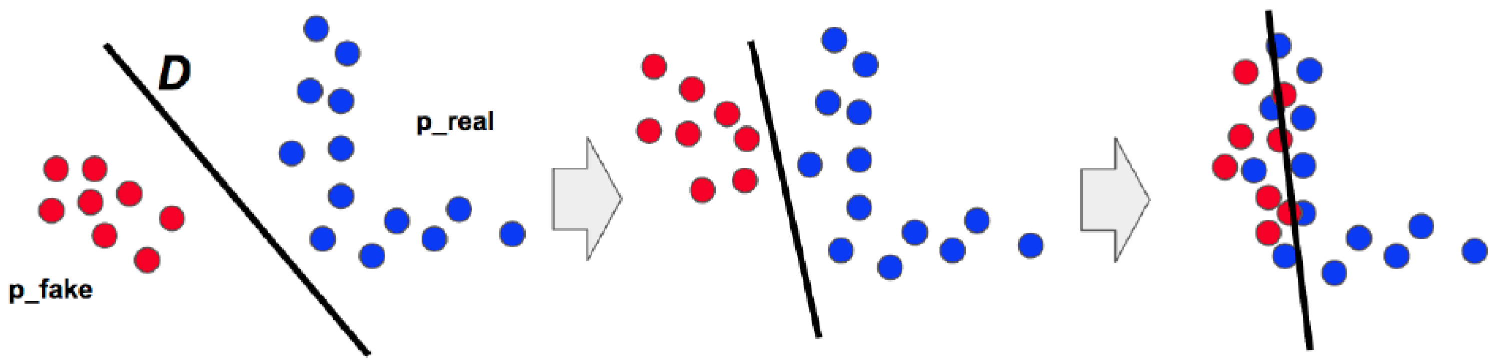
We can draw a contour plot of **max$_D$ V(D,G)** according to the mean value of p_fake.

Now we can optimize p_fake (G) by minimizing max$_D$ V(D,G) .

Convergence of Algorithm

- Graphical Explanation

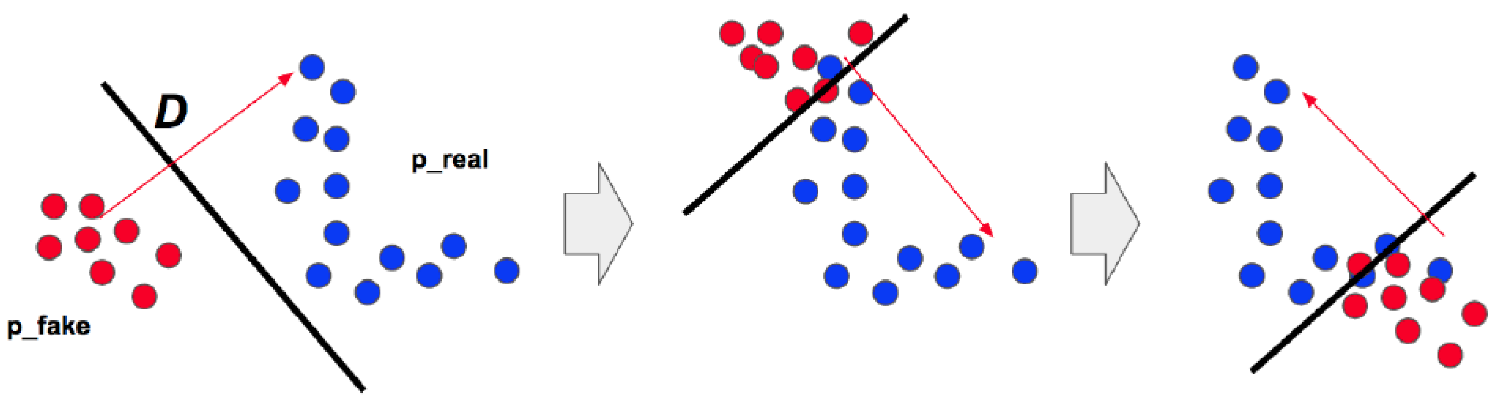
  - But, finding $D^*_G = \arg\max_D V(D, G)$ for every G is intractable.
  - Instead, find $D^*_{\bar{G}} = \arg\max_D V(D, \bar{G})$, then solve for $\min_G V(D^*_{\bar{G}}, G)$.
  - Above proposition guarantees that this is sufficient.



Problem

▪ Mode Collapse

- In practice, D is not optimal (maximal) after only k step training. Also, its learning capacity is limited.
- In this case, $p_G^{fake}(x)$ is guided by defective D. Then, the density of $p_G^{fake}(x)$ is concentrated at small (collapsed) area where $D(x)$ is the highest (~ 1), rather than distributed throughout wide are where $D(x)$ is roughly high (> 0.5).



Problem

- Gradient Vanishing
  - When p^{real} and p_G^{fake} are too different that their supports do not significantly overlap, $D(x)$ would perfectly classify them.
  - In this case, the gradient vanshes for $G(x)$ and GAN is not trained.
  - In other terms, the problem is due to the **ratio-based** divergence measure (f-divergence), which does not give meaningful value when the supports of the two distributions are (almost) disjoint.