GANs (Generative Adversarial Nets)

Ian J. Goodfellow

KESL
서동주

Contents

Alchemist
(Generator)

Jeweler
(Discriminator)

The coolest idea in ML in the last twenty years - Yann Lecun

Alchemist

(Generator)

She have to make gold bars that others can not distinguish.

He must distinguish the Fake.

Jeweler

(Discriminator)

both ability going up

Alchemist
(Generator)

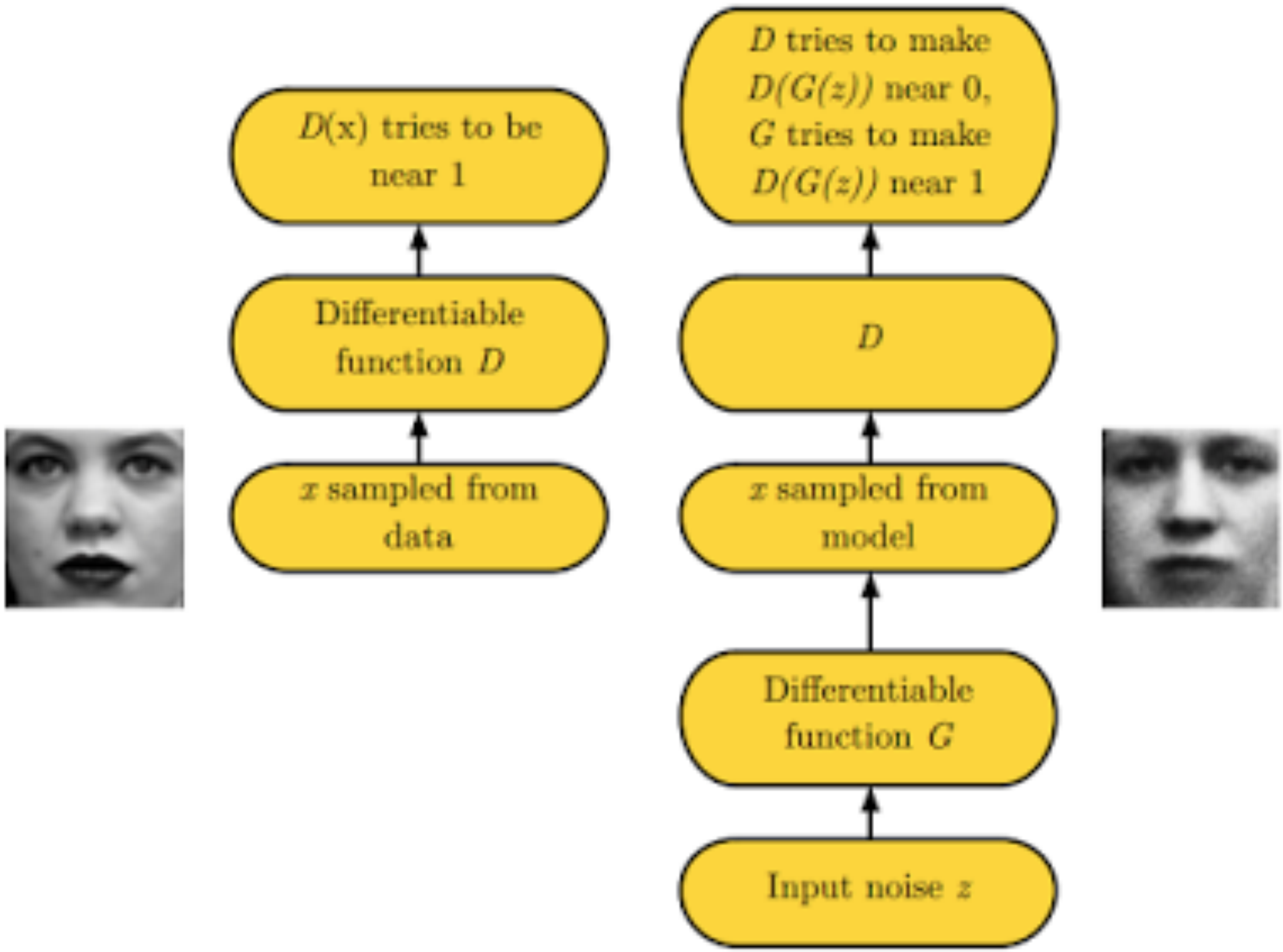
Vs

Jeweler
(Discriminator)

Goal : It can not really tell whether this is real gold or not

Alchemist

Jeweler



출처 : NIPS 2016 Tutorial: Generative Adversarial Networks

It could use another method,
but this paper used multilayer perceptron model for D and G.

$$\min_G \max_D V(D, G) = \mathbb{E}_{\boldsymbol{x} \sim p_{\text{data}}(\boldsymbol{x})}[\log D(\boldsymbol{x})] + \mathbb{E}_{\boldsymbol{z} \sim p_{\boldsymbol{z}}(\boldsymbol{z})}[\log(1 - D(G(\boldsymbol{z})))].$$

Define multilayer perceptron G (Generator)

Generator distribution : P_g

generator input : noise variables $P_z(z)$

Data space : $G(z; \theta_g)$

Define multilayer perceptron D(Discriminator)

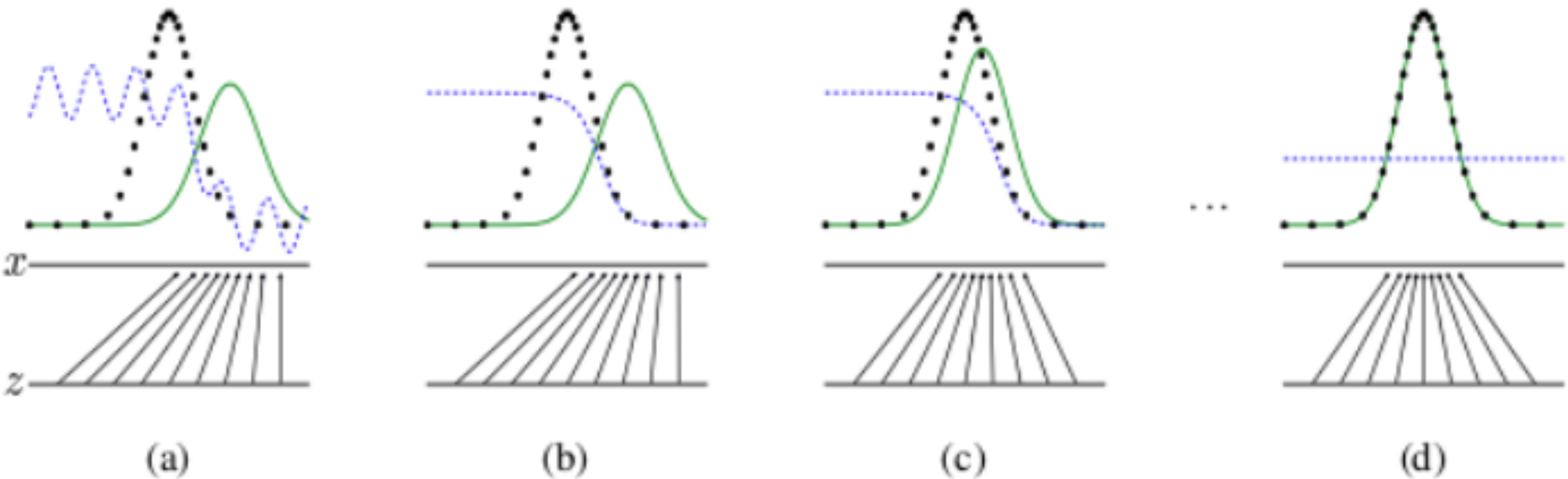
Data space : $D(x; \theta_d)$

Output : Probability (Fake or not)

$$\min_G \max_D V(D,G) = \mathbb{E}_{\boldsymbol{x}\sim p_{\text{data}}(\boldsymbol{x})}[\log D(\boldsymbol{x})] + \mathbb{E}_{\boldsymbol{z}\sim p_{\boldsymbol{z}}(\boldsymbol{z})}[\log(1 - D(G(\boldsymbol{z})))].$$

=

minimax problem for Value function V(D,G)



······· : data generating distribution

x , z : domain

: generative distribution

: x = G(z) mapping

······· : discriminator distribution

Two result

1. Introduced method will get global optimum when $P_g = P_{data}$
2. They prove that the algorithm presented in the paper finds the global optimum

Proposition

For G fixed, the optimal discriminator D is

$$D_G^*(x) = \frac{p_{data}(x)}{p_{data}(x) + p_g(x)}.$$

Proof. The training criterion for the discriminator D, given any generator G, is to maximize the quantity $V(G, D)$

$$\begin{aligned} V(G, D) &= \int_x p_{data}(x) log(D(x))dx + \int_z p_z(z) log(1 - D(G(z)))dz \\ &= \int_x p_{data}(x) log(D(x)) + p_g(x) log(1 - D(x))dx \end{aligned}$$

For any $(a, b) \in \mathbb{R}^2 \setminus \{0, 0\}$, the function $y \rightarrow a log(y) + \mathrm{b} log(1 - y)$ achieves its maximum in $[0, 1]$ at $\frac{a}{a+b}$. The discriminator does not need to be defined outside of $Supp(p_{data}) \cup Supp(p_g)$, concluding the proof. ■

Main Theorem

The global minimum of the virtual training criterion C(G) is
Achieved if and only if $P_g = P_{data}$. At that point, C(G) achieves the value -log4

$$C(G) = -\log(4) + KL\left(p_{\text{data}} \left\| \frac{p_{\text{data}} + p_g}{2}\right.\right) + KL\left(p_g \left\| \frac{p_{\text{data}} + p_g}{2}\right.\right)$$

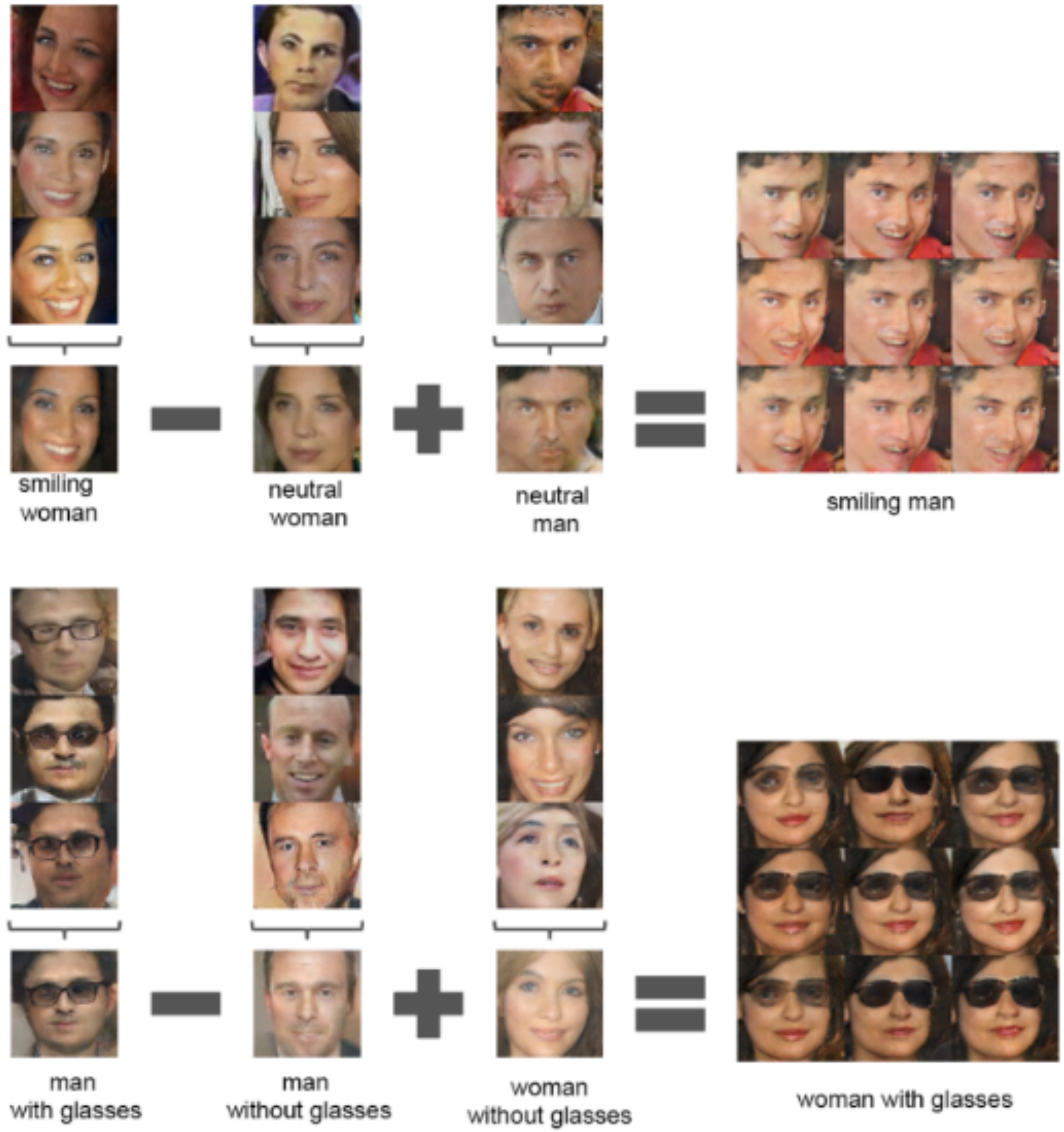
$$C(G) = -\log(4) + 2 \cdot JSD\left(p_{\text{data}} \| p_g\right)$$

Convergence of Algorithm

Proposition 2. *If G and D have enough capacity, and at each step of Algorithm 1, the discriminator is allowed to reach its optimum given G, and p_g is updated so as to improve the criterion*

$$\mathbb{E}_{\boldsymbol{x}\sim p_{data}}[\log D_G^*(\boldsymbol{x})] + \mathbb{E}_{\boldsymbol{x}\sim p_g}[\log(1 - D_G^*(\boldsymbol{x}))]$$

Proof. Consider $V(G, D) = U(p_g, D)$ as a function of p_g as done in the above criterion. Note that $U(p_g, D)$ is convex in p_g. The subderivatives of a supremum of convex functions include the derivative of the function at the point where the maximum is attained. In other words, if $f(x) = \sup_{\alpha\in\mathcal{A}} f_\alpha(x)$ and $f_\alpha(x)$ is convex in x for every α, then $\partial f_\beta(x) \in \partial f$ if $\beta = \arg\sup_{\alpha\in\mathcal{A}} f_\alpha(x)$. This is equivalent to computing a gradient descent update for p_g at the optimal D given the corresponding G. $\sup_D U(p_g, D)$ is convex in p_g with a unique global optima as proven in Thm 1, therefore with sufficiently small updates of p_g, p_g converges to p_x, concluding the proof. $\square$



DCGAN , Alec Radford et al.2016

BEGAN , David Berthelot et al.2017

QnA